Microsoft® System Center

Software Update Management Field Experience

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

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.

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 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

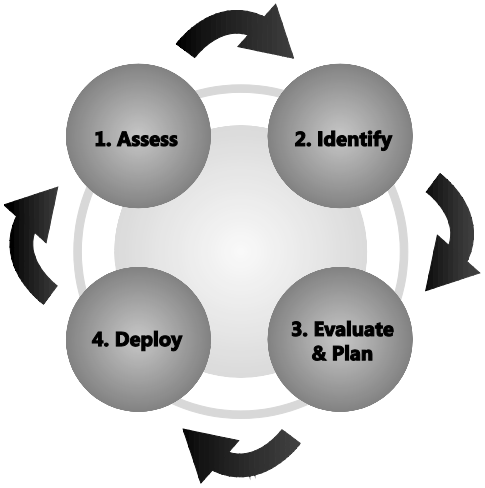


Figure 3-1 The four phases of the software update management process

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.

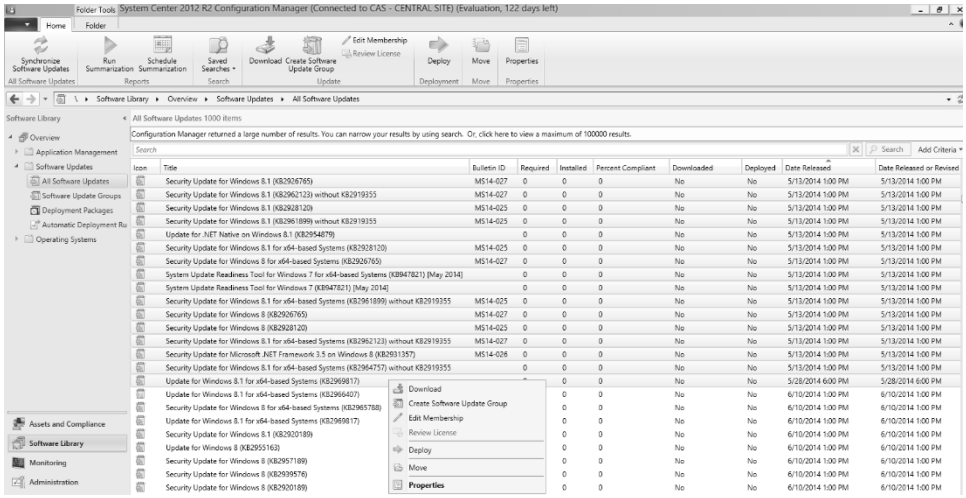
Quarterly and yearly groups

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.

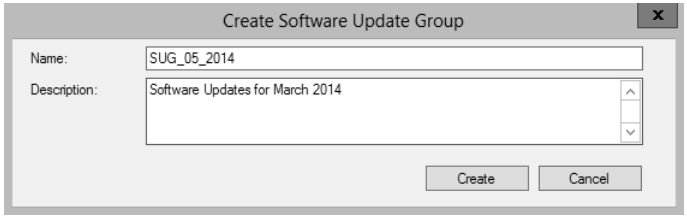
Using software update groups

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

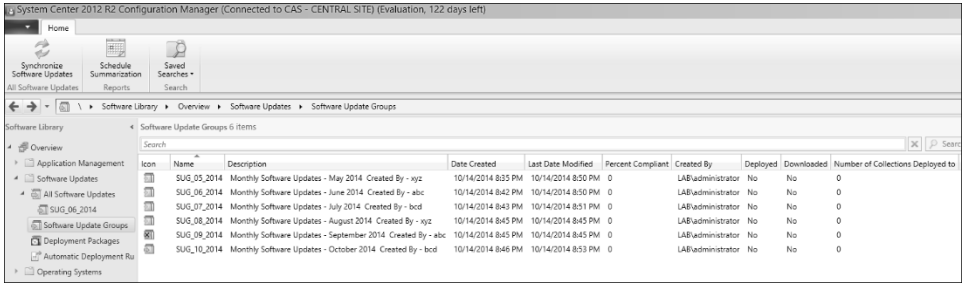
1. Open the Configuration Manager console, select Software Library, then Overview, then Software Updates, and then All Software Updates. Select multiple software updates. You can use search criteria to find a specific bulletin ID, article ID, or a string that would appear in the title for software updates.
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.



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:



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:



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.

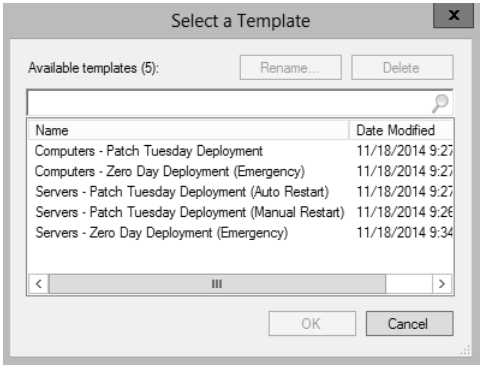


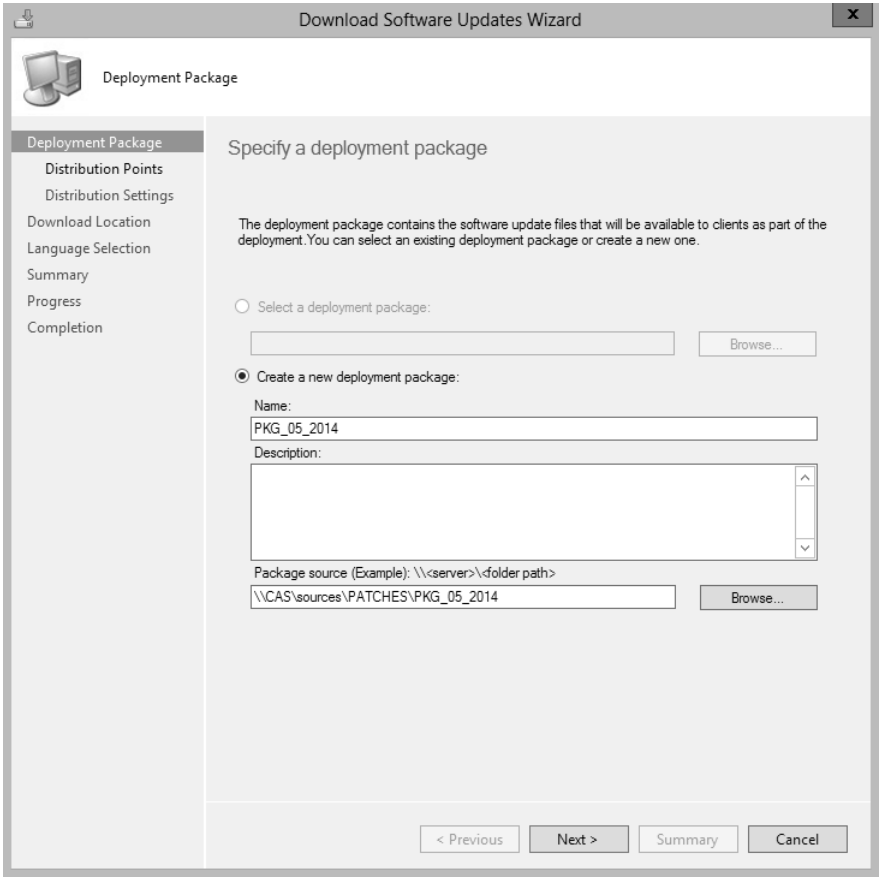
FIGURE 3-2 Examples of patch templates

Using deployment packages

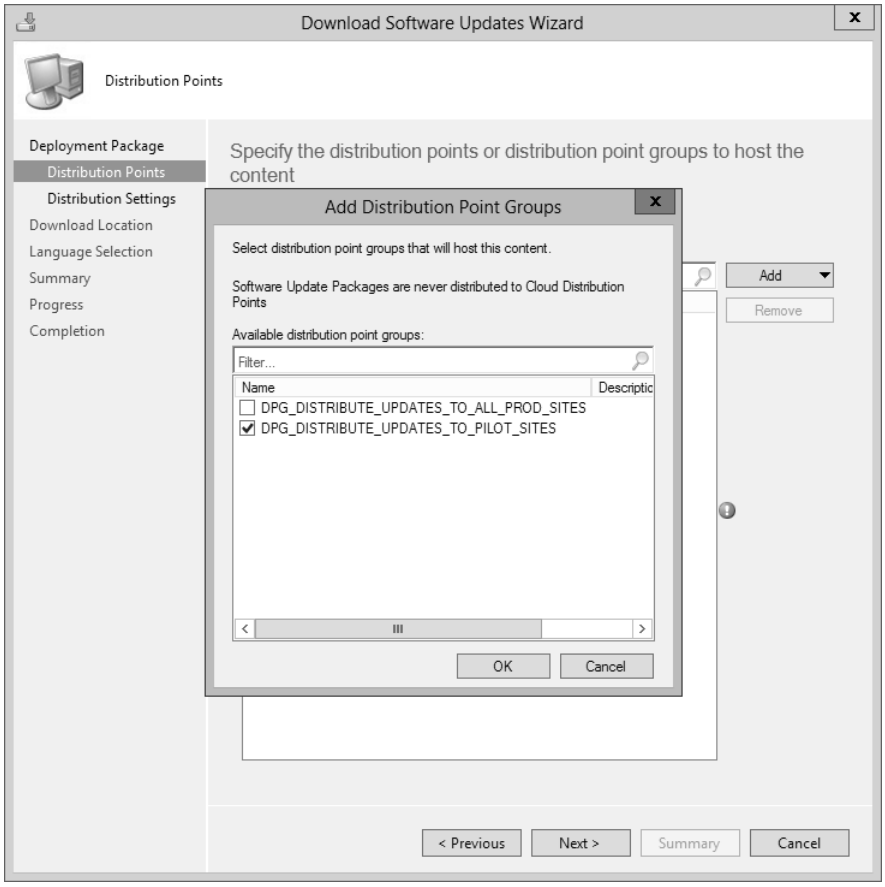
A software update deployment package is used to download software updates from Microsoft to a network shared folder and copy the software update source files to the content library on the site servers and on distribution points that are defined in the deployment.

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:

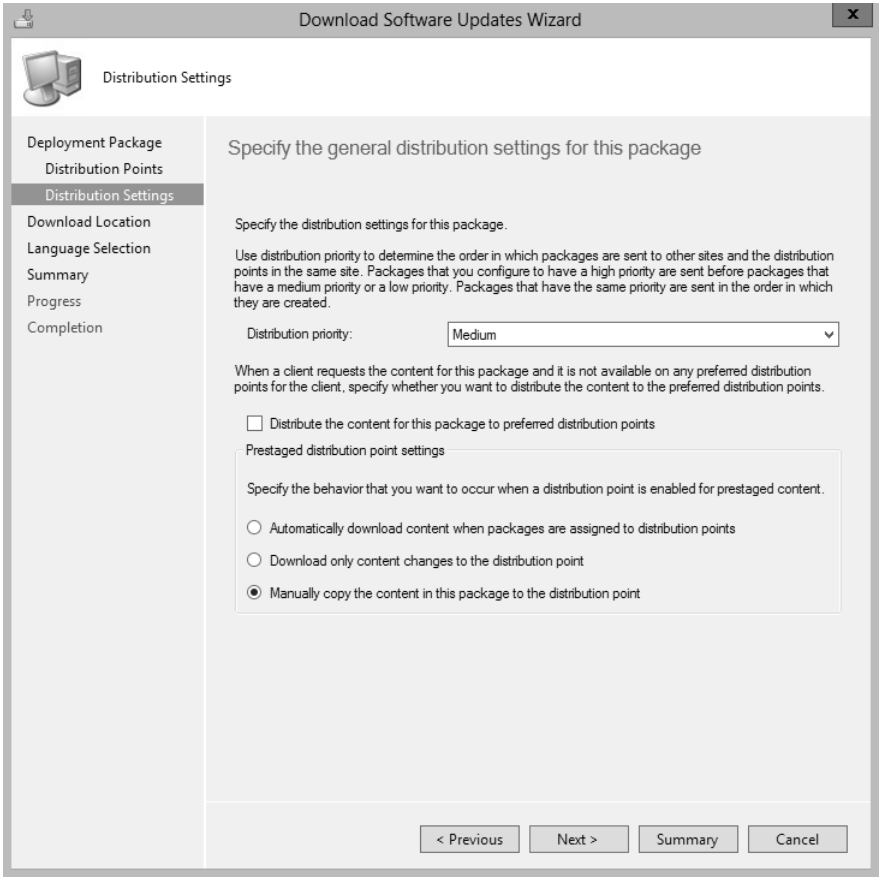


3. On the Distribution Points page, click Add to open the Add Distribution Point Groups dialog box. In this dialog box you can select distribution points that will host the content from the list of available distribution points displayed. In the following example, the distribution point where pilot computers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



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:



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.

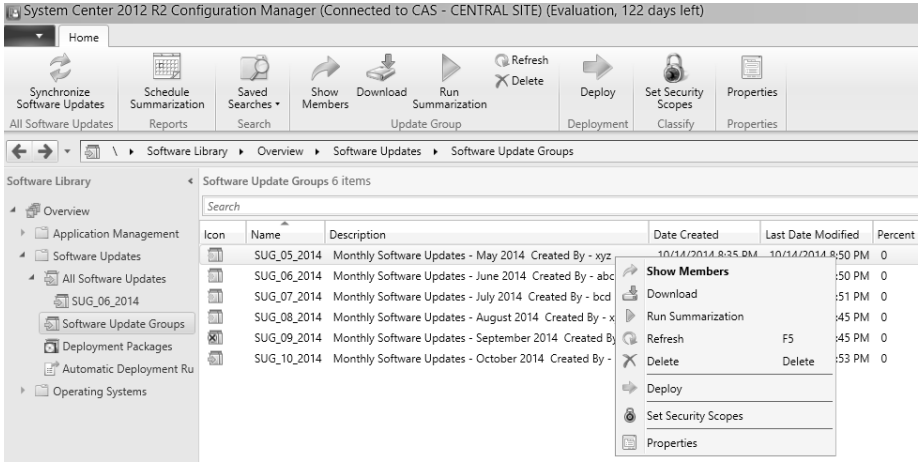
Complete the following steps to automatically deploy the software updates:

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.

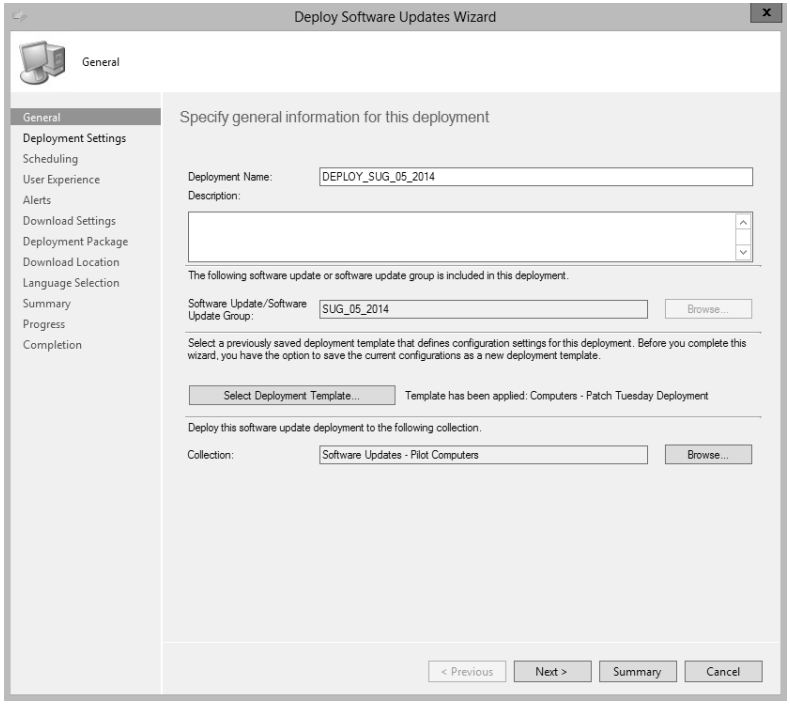
Manual deployment of software updates

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.



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:



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.

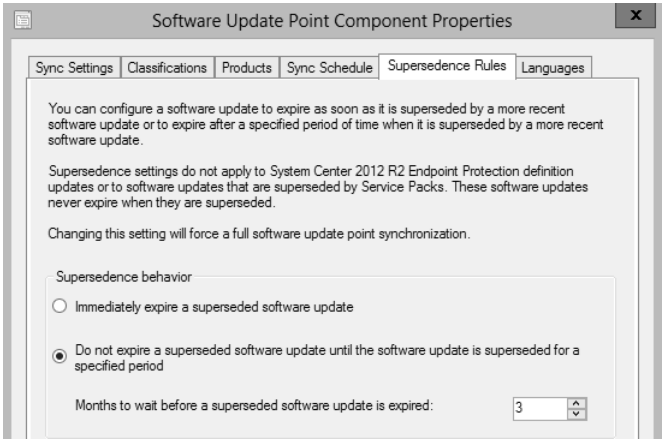


FIGURE 3-3 The Supersedence Rules tab of the Software Update Point Component Properties dialog box

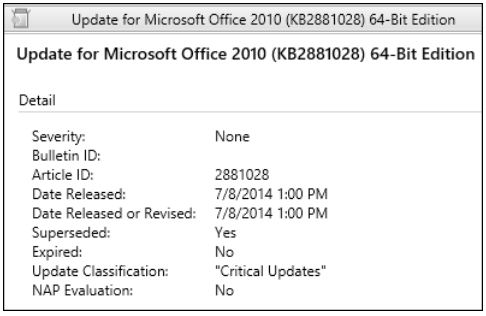


FIGURE 3-4 An example of a superseded but not expired update

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.

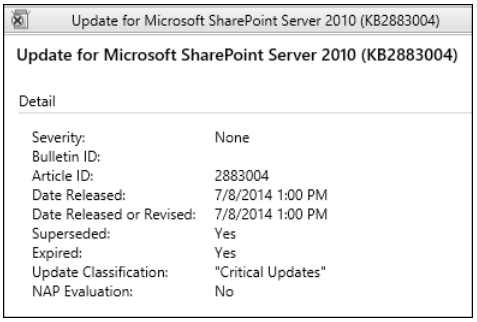


FIGURE 3-5 Example of a superseded and expired software update

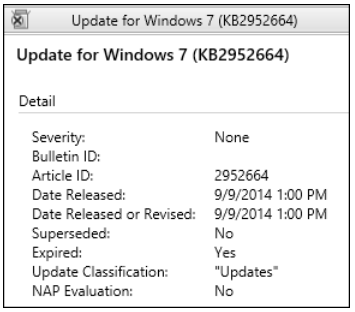


FIGURE 3-6 Example of an expired software update that is not superseded

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.

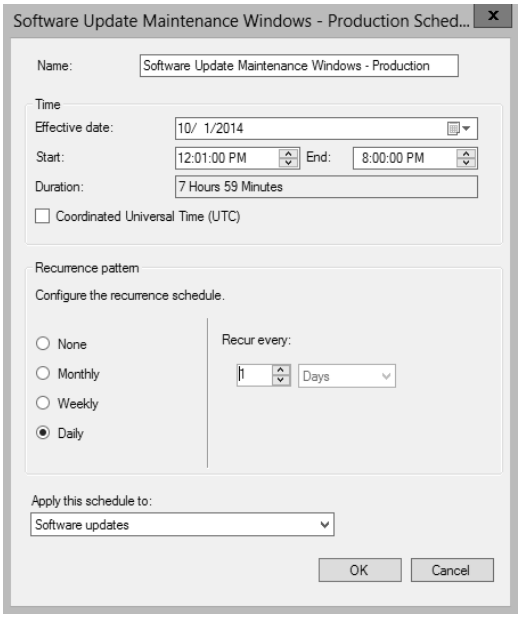


FIGURE 3-8 Software Update Maintenance Windows settings

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.

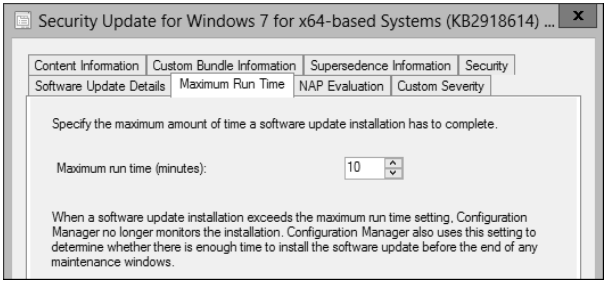


FIGURE 3-9 Software update maximum run time

This page intentionally left blank