Automatic Deployment Rules within SCCM

Using SCCM ADR to achieve automatic approval and distribution of Windows Updates

Document Owner: Gary Wells

Issue Date: 08 August 2017

Version: 0.1

Classification: INTERNAL

Contents

[1. Introduction 4](#_Toc490047496)

[2. Current Scenario 5](#_Toc490047497)

[3. Ways Forward 6](#_Toc490047498)

[4. Software Update Deployment Workflows 7](#_Toc490047499)

[4.1. Manual Deployment of Software Updates 7](#_Toc490047500)

[4.2. Automatic Deployment of Software Updates 7](#_Toc490047501)

[5. Software Update Deployment Process 9](#_Toc490047502)

[5.1. Required System Restart 9](#_Toc490047503)

[5.2. Deployment Re-evaluation Cycle 9](#_Toc490047504)

[6. Best Practice 10](#_Toc490047505)

[6.1. Installation Best Practices 10](#_Toc490047506)

[6.2. Operational Best Practices 12](#_Toc490047507)

[7. ADR Deployment 13](#_Toc490047508)

[7.1. Create and Deploy a Software Update Group (SUG) 13](#_Toc490047509)

[7.2. Add Software Updates to a deployed SUG 13](#_Toc490047510)

[7.3. Create an Automatic Deployment Rule (ADR) 13](#_Toc490047511)

[7.4. Add a new Deployment to an existing ADR 24](#_Toc490047512)

[8. Maintenance Windows 29](#_Toc490047513)

[8.1. Using multiple Maintenance Windows 29](#_Toc490047514)

[8.2. How to configure Maintenance Windows 29](#_Toc490047515)

[9. Using WMI and PowerShell to manage SCCM Collections 30](#_Toc490047516)

[9.1. Overview 30](#_Toc490047517)

[9.2. SMS Provider Role in SCCM 30](#_Toc490047518)

[9.3. Using native WMI to create Collection query rules 30](#_Toc490047519)

[9.4. Using .NET Type Accelerators 30](#_Toc490047520)

[9.5. Membership Rules 32](#_Toc490047521)

[9.6. Refresh Schedules 33](#_Toc490047522)

[10. Appendix 35](#_Toc490047523)

Document Information

Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Version** | **Status** | **Prepared by** | **Comments** |
| 19th July 2017 | 0.1 | Draft | Gary Wells | Initial Draft |
|  |  |  |  |  |
|  |  |  |  |  |

Document Control

|  |  |  |
| --- | --- | --- |
| **Role** | **Name** | **E-mail** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Approval

|  |  |  |  |
| --- | --- | --- | --- |
| **Role** | **Name** | **Signature** | **Sign-off Date** |
|  |  |  |  |
|  |  |  |  |

# Introduction

HSBC currently have a vast estate of Windows Server – over 30000 devices and increasing daily. With the imminent addition of Server 2016 images to the environment, this could result in many more patches to be distributed – and 2016 will have either a GUI version or a Core version of patches to evaluate. The challenge is to patch all servers in a timely manner without severely impacting the team that is responsible. Update management is performed using SCCM rather than WSUS as there are many advantages to this.

1. **One Client**:  
   The ConfigMgr client controls all installations on a computer, both software updates and application installations. No more “Another installation is already running” errors. When ConfigMgr installs a Software Update to a client, Software **distribution** is paused to avoid these.
2. **Reporting**:  
   There are many built-in reports for Software Update Compliance, troubleshooting and details. These reports combined with all the other information ConfigMgr holds about the clients in the environment means you can easily create customized reports to aid with identifying non-compliance.
3. **Unified Management / One console:**When using Configuration Manager 2012 for Software Updates as well as all other features in Configuration Manager 2012 you will have a single management console.
4. **Maintenance Windows:**Maintenance Windows can be used to control when changes are allowed to be made to specific systems. This means that you deploy the update once and then based on Maintenance Windows the updates are installed and the servers are rebooted according to the deployment. More information about Maintenance Windows can be found at: <http://technet.microsoft.com/en-us/library/hh508762>
5. **Scheduling**:  
   In Configuration Manager 2012 there are many more available options when it comes to scheduling an update and in combination with Maintenance Windows, is a powerful tool.
6. **One Infrastructure**:  
   The Software Update files are downloaded from the local SCCM DP and not from a dedicated WSUS/SUP server.
7. **Automatic Deployment Rules**:  
   It is now possible to automatically approve updates, download them and distribute them to the DP’s automatically, just as you could in WSUS. More Information on ADRs: <http://technet.microsoft.com/en-us/library/gg682168>
8. **System Center Updates Publisher**:  
   You can use System Center Updates Publisher to download vendors catalog’s with updates from sources such as Adobe, HP and Dell and deploy them as updates in Configuration Manager 2012. More Information: <http://technet.microsoft.com/en-us/library/hh134742.aspx>
9. **Targeting:**Query-based collections provide powerful options for targeting. It is possible to dynamically create a collection based on any value that exists in the database - for instance divide all clients based on the last number in the computer name, and deploy Software Updates to computers with odd computers on one day and all with even numbers the day after.
10. **Offline Servicing of Images:**  
    SCCM 2012 enables the use of Offline Servicing on OS Images. This means you can install OS related Software Updates in the image without rebuilding it from ISO.More information: <http://blogs.technet.com/b/inside_osd/archive/2011/04/18/configuration-manager-2012-offline-servicing-for-operating-system-images.aspx>

# Current Scenario

Currently, the Windows Updates are downloaded and manually tested outside of production. The required patches are then manually added to various collections within SCCM and distributed on a schedule. The process can take in excess of 1 month if there is a significant number of updates, which means there is another set of patches already released before the previous month’s has been fully deployed.

Although a Maintenance Window is required to be set during self-service Server Setup, these are not enforced. Currently ALL patching is performed as a managed piece of work, and as such is subject to Change Control. This means Service Owners are able to veto patching works which can leave systems vulnerable for some considerable time

In addition, there are still Operating Systems in the estate that have no ongoing patches being released – Server 2003 and 2003 R2 as well as Server 2000. This means that any existing vulnerabilities could now be potentially exploited if these are NOT fully up to date with ALL patches available

# Ways Forward

A way to mitigate this to a large extent is to utilise Automatic Approval and Automated Deployment Rules within SCCM

Software Updates in System Center 2012 Configuration Manager provides a set of tools and resources that can help manage the complex task of tracking and applying software updates to client computers in the enterprise. An effective software update management process is necessary to maintain operational efficiency, overcome security issues, and maintain the stability of the network infrastructure. However, because of the changing nature of technology and the continual appearance of new security threats, effective software update management requires consistent and continual attention.

# Software Update Deployment Workflows

There are two scenarios for deploying Software Updates with SCCM: Manual Deployment and Automatic Deployment. Typically, you deploy Software Updates manually to create a baseline for client computers, and then you manage ongoing Software Updates on clients by using Automatic Deployment. The following sections provide a summary for the workflow for Manual and Automatic deployment for Software Updates.

## Manual Deployment of Software Updates

Manual deployment of Software Updates is the process of selecting software updates in the Configuration Manager console and manually starting the deployment process. This process typically requires significant effort, **as each patch for each OS** needs to be evaluated, approved and added to a collection manually. You typically use this method to get the client computers up-to-date with required Software Updates before you create Automatic Deployment Rules that manage ongoing monthly Software Updates. The following steps show the general workflow for manual deployment of Software Updates:

1. Filter for Software Updates that use specific requirements. *(For example, you could provide criteria that retrieves all security or critical software updates that are required on all clients)*
2. Create a Software Update Group that contains the Software Updates.
3. Download the content for the Software Updates in the Software Update Group.
4. Manually deploy the Software Update Group.

## Automatic Deployment of Software Updates

Automatic Software Updates deployment is configured by using Automatic Deployment Rules - ADRs. You typically use this method of deployment for monthly Software Updates (generally known as Patch Tuesday) and for managing definition updates if System Center Endpoint Protection (SCEP) is in use.

When the rule runs, previous Software Updates are removed from the Software Update Group (if using an existing group). The Software Updates that meet a specified criteria (*e.g. all Security updates released in the last week)* are added to the Software Update Group. The content files are downloaded and copied to Distribution Points and the Software Updates are deployed to client computers in the target collection. The following list provides the general workflow for automatic deployment of Software Updates:

1. Create an automatic deployment rule that specifies deployment settings such as:
   * Target collection
   * Decide whether to enable the deployment or report on Software Updates compliance for the client computers in the target collection
   * Software Updates criteria
   * Evaluation and deployment schedules
   * User experience
   * Download properties
2. The Software Updates are added to a Software Update Group.
3. The Software Update Group is deployed to the client computers in the target collection

Deployment strategy can be very flexible: For example, you might create the ADR and target a small collection of dedicated test clients for the first deployment cycle. After you verify that the Software Updates are installed on the test group, and they have operated with no impact from the patching for a set number of days, you can change the ADR to a target collection that includes a larger set of clients, and then an even larger set until all clients have been updated.

The software update objects that are created by the ADRs are interactive.

* Software Updates that were deployed by using an ADR are automatically deployed to new clients added to the target collection.
* New Software Updates added to a Software Update Group are automatically deployed to the clients in the target collection.
* You can enable or disable deployments at any time for the ADR

# Software Update Deployment Process

After you deploy Software Updates or when an ADR runs and deploys Software Updates, a Deployment Assignment Policy is added to the machine policy for the SCCM site. The Software Updates are downloaded from the download location, the Internet, or network shared folder, to the package source. The Software Updates are copied from the package source to the Content Library on the Site Server, and then copied to the Content Library on the Distribution Point(s).

When a client computer in the target collection for the deployment receives the machine policy, the Software Update Client Agent starts an evaluation scan. The client agent downloads the content for required Software Updates from a Distribution Point to the local client cache soon after it receives the deployment, but waits until after the **Software available time** setting for the deployment before the Software Updates are available to install. *The Software Updates in optional deployments (deployments that do not have an installation deadline) are not downloaded until a user manually starts the installation.*

When the configured deadline passes, the Software Updates Client Agent performs a scan to verify that the Software Updates are still required. Then it checks the local cache on the client computer to verify that the Software Update source files are still available. Finally, the client installs the Software Updates. If the content was deleted from the client cache to make room for another deployment, the client re-downloads the Software Updates from the Distribution Point to the client cache. **Software Updates are always downloaded to the client cache regardless of the configured maximum client cache size.** When the installation is complete, the client agent verifies that the Software Updates are no longer required, and then sends a state message to the management point to indicate that the Software Updates are now installed on the client.

## Required System Restart

***By default***, when software updates from a required deployment are installed on a client computer and a system restart is required for the installation to finish, the system restart is performed automatically.

For software updates that were installed before the deadline, the automatic system restart is postponed until the deadline is reached, unless the computer is restarted before that for some other reason.

The system restart can be suppressed for servers and workstations. These settings are configured in the **User Experience** page of the Deploy Software Updates Wizard or Create Automatic Updates Rule Wizard.

## Deployment Re-evaluation Cycle

***By default***, client computers start a deployment reevaluation cycle every 7 days. During this evaluation cycle, the client computer scans for software updates that were previously deployed and installed. If any software updates are missing, the software updates are reinstalled from the local cache. If a software update is no longer available in the local cache, it is downloaded from a Distribution Point and then installed. You can configure the reevaluation schedule on the **Software Updates** page in client settings for the site.

# Best Practice

## Installation Best Practices

Use the following best practices when you install software updates in Configuration Manager:

* **Use a Shared WSUS Database for Software Update Points (SCCM 2012 SP1 onwards)**

When you install more than one Software Update Point at a primary site, use the same WSUS database for each software update point in the same Active Directory forest. By sharing the same database you can significantly mitigate the client and network performance impact that can occur when clients switch to a new software update point.

When a client switches to a new software update point that shares a database with the old software update point, a delta scan still occurs, but this scan is much smaller than it would be if the WSUS server had its own database.

Important

You must also share the local WSUS content folders when you use a shared WSUS database for Software Update Points

For more information:

See step 3 of [Prepare the Network Environment for NLB Software Update Point Site Systems](https://technet.microsoft.com/en-us/library/hh237369.aspx#BKMK_PrepareNetworkForNLB) & step 7 of [Install WSUS 3.0 (on each server that will host the software update point site system role)](https://technet.microsoft.com/en-us/library/hh237369.aspx#BKMK_InstallWSUS)

For more information about Software Update Point switching, see the [Software Update Point Switching](https://technet.microsoft.com/en-us/library/gg712696.aspx#BKMK_SUPSwitching) section in the [Planning for Software Updates in Configuration Manager](https://technet.microsoft.com/en-us/library/gg712696.aspx) article

* **When Configuration Manager and WSUS use the same SQL Server, configure one of these to use a named instance and the other to use the default instance**

*When the Configuration Manager and WSUS databases use the same SQL Server and share the same instance of SQL Server, you cannot easily determine the resource usage between the two applications. When you use a different SQL Server instance for Configuration Manager and WSUS, it is easier to troubleshoot and diagnose resource usage issues that might occur for each application.*

* **Use a custom website for the WSUS installation**

*When you install WSUS 3.0, you can specify whether to use the default Internet Information Services (IIS) website or create a new custom WSUS 3.0 website. As a best practice, select* ***Create a Windows Server Update Services 3.0 Web site*** *so that IIS hosts the WSUS 3.0 services in a dedicated website instead of sharing the same website with other Configuration Manager site systems or other software applications. When you use a custom website for WSUS 3.0, WSUS configures port 8530 for HTTP and port 8531 for HTTPS. You must specify these port settings when you create the software update point for the site.*

* **Specify the "Store updates locally" setting for the WSUS installation**

*When you install WSUS 3.0, select the* ***Store updates locally*** *setting. When this setting is selected, the license terms that are associated with software updates are downloaded during the synchronization process and stored on the local hard drive for the WSUS server. When this setting is not selected, client computers might fail to scan for software updates compliance for software updates that have license terms. When you install the software update point, WSUS Synchronization Manager verifies that this setting is enabled every 60 minutes, by default.*

## Operational Best Practices

Use the following best practices when you use software updates:

* Limit software updates to 1000 in a single software update deployment

You must limit the number of software updates to 1000 for each software update deployment. When you create an automatic deployment rule, verify that the criteria that you specify does not result in more than 1000 software updates. When you manually deploy software updates, do not select more than 1000 updates to deploy.

* Create a new software update group each time an automatic deployment rule runs for “Patch Tuesday” and for general deployment

There is a limit of 1000 software updates for a software update deployment. When you create an Automatic Deployment Rule, you specify whether to use an existing Update Group or create a new Update Group each time the rule runs. When you specify criteria in an Automatic Deployment Rule that results in multiple Software Updates and the rule runs on a recurring schedule, specify to create a new Software Update Group each time the rule runs. This will prevent the deployment from surpassing the limit of 1000 Software Updates per deployment.

* Use an existing Software Update Group for Automatic Deployment Rules for Endpoint Protection definition updates

Always use an existing Software Update Group when you use an Automatic Deployment Rule to deploy Endpoint Protection definition updates on a frequent basis. Otherwise, potentially hundreds of Software Update Groups will be created over time. Typically, definition update publishers will set definition updates to expire when they are superseded by four newer updates. Therefore, the Software Update Group that is created by the Automatic Deployment Rule will never contain more than four definition updates for the publisher: one active and three superseded.

# ADR Deployment

You can automatically deploy software updates by adding new software updates to an existing software update group associated with an active deployment or you can use an automatic deployment rule (ADR).

## Create and Deploy a Software Update Group (SUG)

## Add Software Updates to a deployed SUG

After you create and deploy a software update group, you can add software updates to the update group and they will be automatically deployed.

Note

When you add software updates to an existing Software Update Group that has already been deployed, it might take several minutes before the additional Software Updates are added to the deployment.

1. In the Configuration Manager console, navigate to **Software Library** > **Overview** > **Software Updates**.
2. Select the software updates that are to be added to the new software update group.
3. On the **Home** tab, in the **Update** group, click **Edit Membership**.
4. Select the software update group to which you want to add the software updates as members.

Click the **Software Update Groups** node to display the software update group.

1. Click the software update group, and in the **Home** tab, in the **Update** group, click **Show Members** to display a list of the software updates in the group.

## Create an Automatic Deployment Rule (ADR)

You can automatically approve and deploy software updates by using an ADR. You can have the rule add software updates to a new software update group each time the rule runs or add software updates to an existing group.

When a rule runs and adds software updates to an existing group, the rule removes all existing software updates from the group and then adds the software updates that meet the criteria that you define to the group.

To run an ADR to find newly released software updates each day and deploy them to clients, for example, you must choose the option to create a new software update group instead of adding the software updates to an existing group.

Warning

Before you create an ADR for the first time, verify that software updates synchronization has completed at the site. This is particularly important when you run Configuration Manager with a non-English language because software update classifications are displayed in English before the first synchronization, and then displayed in the localized language **after** Software Update synchronization completes. Rules that you create before you synchronize software updates might not work properly after synchronization because the text string might not match.

1. In the **Configuration Manager console**, navigate to **Software Library** **Overview** > **Software Updates** > **Automatic Deployment Rules**.
2. On the **Home** tab, in the **Create** group, click **Create Automatic Deployment Rule**. The Create Automatic Deployment Rule Wizard opens.
3. On the **General** page, configure the following settings:
   * **Name**: *Specify the name for the ADR. The name must be unique, help to describe the objective of the rule, and identify it from others in the Configuration Manager site.*
   * **Description**: *Specify a description for the ADR. The description should provide an overview of the deployment rule and any other relevant information that helps to identify and differentiate the rule among others in the Configuration Manager site. The description field is optional, has a limit of 256 characters, and has a blank value by default.*
   * **Select Deployment Template**: *Specify whether to apply a previously saved deployment template. You can configure a deployment template to contain multiple common software update deployment properties that can then be used when creating ADRs. These templates help to ensure consistency across similar deployments and to save time.*

*You can select from the built-in software update deployment templates from the Automatic Deployment Rule Wizard. The* ***Definition Updates*** *template provides common settings to use when you deploy definition software updates. The* ***Patch Tuesday*** *template provides common settings to use when you deploy software updates on a monthly cycle.*

* **Collection**: *Specifies the target collection to be used for the deployment. Members of the collection receive the software updates that are defined in the deployment.* 
  + *Decide whether to add software updates to a new or existing software update group. In most cases, you will probably choose to create a new software update group when the ADR is run. However, you might choose to use an existing group if the rule runs on a more aggressive schedule.*

*For example, if you will run the rule daily for definition updates, then you could add the software updates to an existing software update group.*

* + ***Enable the deployment after this rule is run****: Specify whether to enable the software update deployment after the ADR runs.*

**Regarding this specification, consider the following:**

***When you enable the deployment***

* + *The Software Updates that meet the criteria defined in the rule are added to a Software Update Group*
  + *The Software Update content is downloaded as necessary*
  + *The content is copied to the specified Distribution Point(s)*
  + *The Software Updates are deployed to the clients in the target collection.*

***When you do not enable the deployment***

* + *The software updates that meet the criteria defined in the rule are added to a Software Update Group and the Software Updates deployment policy is configured but the Software Updates are* ***not*** *downloaded, nor deployed to clients. This situation provides you time as needed to prepare to deploy the software updates, verify that the software updates that meet the criteria are adequate, and then enable the deployment at a later time.*

1. On the **Deployment Settings** page, configure the following settings:
   * **Use Wake-on-LAN to wake up clients for required deployments**: *Specifies whether to enable Wake On LAN at the deadline to send wake-up packets to computers that require one or more software updates in the deployment. Any computers that are in sleep mode at the installation deadline time will be awakened so the software update installation can initiate. Clients that are in sleep mode that do not require any software updates in the deployment are not started. By default, this setting is not enabled.*

Note

Before you can use this option, you must configure computers and networks for “Wake-On-LAN”

* + **Detail level**: Specify the level of detail for the state messages that are reported by client computers.

Note

When you deploy definition updates, set the detail level to **Error only** to have the client report a state message only when a definition update fails to be delivered to the client. Otherwise, the client will report a large number of state messages that might impact performance on the site server.

* + **License terms setting**: Specify whether to automatically deploy software updates with associated license terms. Some software updates include license terms, such as a service pack. When you automatically deploy software updates, the license terms are not displayed and there is not an option to accept the license terms. You can choose to automatically deploy all software updates regardless of an associated license terms or only deploy software updates that do not have associated license terms.

Note

To review the license terms for a software update, you can select the software update in the **All Software Updates** node of the **Software Library** workspace, and then on the **Home** tab, in the **Update** group, click **Review License**.

To find software updates with associated license terms, you can add the **License Terms** column to the results pane in the **All Software Updates** node, and then click the heading for the column to sort by the software updates with license terms.

1. On the **Software Updates** page, configure the criteria for the software updates that the ADR retrieves and adds to the software update group.

Note

The limit for software updates in the ADR is 1000 software updates. To ensure that the criteria that you specify on this page retrieves less than 1000 software updates, consider setting the same criteria on the **All Software Updates** node in the **Software Library** workspace.

Starting in Configuration Manager version 1610, you can filter on the content size for software updates in automatic deployment rules. For example, you can set the **Content Size (KB)** filter to **< 2048** to only download software updates that are smaller than 2MB. Using this filter prevents large software updates from automatically downloading to better support simplified Windows down-level servicing when network bandwidth is limited. For details, see [Configuration Manager and Simplified Windows Servicing on Down Level Operating Systems](https://blogs.technet.microsoft.com/enterprisemobility/2016/10/07/configuration-manager-and-simplified-windows-servicing-on-down-level-operating-systems/).

1. On the **Evaluation Schedule** page, specify whether to enable the ADR to run on a schedule. When enabled, click **Customize** to set the recurring schedule.

*The Software Update Point synchronization schedule is displayed to help you determine the frequency of the evaluation schedule. You should never set the evaluation schedule with a frequency that exceeds the software updates synchronization schedule. The start time configuration for the schedule is based on the local time of the computer that runs the Configuration Manager console.*

Note

To manually run the ADR, select the rule, and then click **Run Now** on the **Home** tab in the **Automatic Deployment Rule** group. Before you manually run the ADR, verify that software updates synchronization has been run since the last time you ran the rule.

The ADR evaluation can run as often as three times per day.

1. On the **Deployment Schedule** page, configure the following settings:
   * **Schedule evaluation**: Specify whether Configuration Manager evaluates the available time and installation deadline times by using UTC or the local time of the computer that runs the Configuration Manager console.

Note

When you select local time, and then select **As soon as possible** for the **Software available time** or **Installation deadline**, the current time on the computer running the Configuration Manager Console is used to evaluate when updates are available or when they are installed on a client. If the client is in a different time zone, these actions will occur when the client's time reaches the evaluation time.

* + **Software available time**: Select one of the following settings to specify when the software updates are available to clients:
    - **As soon as possible**: *Select this setting to make the software updates that are included in the deployment available to the client computers as soon as possible. When you create the deployment with this setting selected, Configuration Manager updates the client policy. Then, at the next client policy polling cycle, clients become aware of the deployment and can obtain the updates that are available for installation.*
    - **Specific time**: *Select this setting to make the software updates that are included in the deployment available to the client computers at a specific date and time. When you create the deployment with this setting enabled, Configuration Manager updates the client policy. Then, at the next client policy polling cycle, clients become aware of the deployment. However, the software updates in the deployment are not available for installation until after the configured date and time.*
  + **Installation deadline**: Select one of the following settings to specify the installation deadline for the software updates in the deployment:
    - **As soon as possible**: *Select this setting to automatically install the software updates in the deployment as soon as possible.*
    - **Specific time**: *Select this setting to automatically install the software updates in the deployment at a specific date and time. Configuration Manager determines the deadline to install software updates by adding the configured* ***Specific time*** *interval to the* ***Software available time****.*

Note

The actual installation deadline time is the displayed deadline time plus a random amount of time up to 2 hours. This reduces the potential impact of all client computers in the destination collection installing the software updates in the deployment at the same time.

You can configure the **Computer Agent** client setting **Disable deadline randomization** to disable the installation randomization delay for required software updates. For more information, see [Computer Agent](https://docs.microsoft.com/en-us/sccm/core/clients/deploy/about-client-settings#computer-agent).

1. On the **User Experience** page, configure the following settings:
   * **User notifications**: *Specify whether to display notification of the software updates in Software Center on the client computer at the configured* ***Software available time*** *and whether to display user notifications on the client computers.*
   * **Deadline behavior**: *Specify the behavior that is to occur when the deadline is reached for the software update deployment. Specify whether to install the software updates in the deployment. Also specify whether to perform a system restart after software update installation regardless of a configured maintenance window. For more information about maintenance windows, see* [*How to use maintenance windows*](https://docs.microsoft.com/en-us/sccm/core/clients/manage/collections/use-maintenance-windows)*.*
   * **Device restart behavior**: *Specify whether to suppress a system restart on servers and workstations after software updates are installed and a system restart is required to complete the installation.*

Important

Suppressing system restarts can be useful in server environments or in cases in which you do not want the computers that are installing the software updates to restart by default. However, doing so can leave computers in an insecure state, whereas allowing a forced restart helps to ensure immediate completion of the software update installation.

* + **Write filter handling for Windows Embedded devices**: When you deploy software updates to Windows Embedded devices that are write filter enabled, you can specify to install the software update on the temporary overlay and either commit changes later or commit the changes at the installation deadline or during a maintenance window. When you commit changes at the installation deadline or during a maintenance window, a restart is required and the changes persist on the device.

Note

When you deploy a Software Update to a Windows Embedded device, make sure that the device is a member of a collection that has a configured maintenance window.

* + **Software updates deployment re-evaluation behavior upon restart**: Starting in Configuration Manager version 1606, select this setting to configure software updates deployments to have clients run a software updates compliance scan immediately after a client installs software updates and restarts. This enables the client to check for additional software updates that become applicable after the client restarts, and to then install them (and become compliant) during the same maintenance window.

1. On the **Alerts** page, configure how Configuration Manager and System Center Operations Manager will generate alerts for this deployment. You can review recent software updates alerts from the **Software Updates** node in the **Software Library** workspace.
2. On the **Download Settings** page, configure the following settings:
   * Specify whether the client will download and install the software updates when a client is connected to a slow network or is using a fallback content location.
   * Specify whether to have the client download and install the software updates from a fallback distribution point when the content for the software updates is not available on a preferred distribution point.
   * **Allow clients to share content with other clients on the same subnet**: Specify whether to enable the use of BranchCache for content downloads. For more information about BranchCache, see [Concepts for content management](https://docs.microsoft.com/en-us/sccm/core/plan-design/hierarchy/fundamental-concepts-for-content-management#branchcache).
   * **If software updates are not available on distribution point in current, neighbor or site groups, download content from Microsoft Updates**: Select this setting to have clients that are connected to the intranet download software updates from Microsoft Update if software updates are not available on distribution points. Internet-based clients can always go to Microsoft Update for software updates content.
   * Specify whether to allow clients to download after an installation deadline when they use metered Internet connections. Internet providers sometimes charge by the amount of data that you send and receive when you are on a metered Internet connection.

Note

Clients request the content location from a Management Point for the Software Updates in a deployment. The download behaviour depends upon how you have configured the Distribution Point, deployment package, and the settings on this page. For more information, see [Content source location scenarios](https://docs.microsoft.com/en-us/sccm/core/plan-design/hierarchy/content-source-location-scenarios).

1. On the **Deployment Package** page, select an existing deployment package or configure the following settings to create a new deployment package:
   * **Name**: *Specify the name of the deployment package. This must be a unique name that describes the package content. It is limited to 50 characters.*
   * **Description**: *Specify a description that provides information about the deployment package. The description is limited to 127 characters*.
   * **Package source**: *Specifies the location of the software update source files. Type a network path for the source location, for example,* ***\\server\sharename\path****, or click* ***Browse*** *to find the network location. You must create the shared folder for the deployment package source files before you proceed to the next page.*

Important

The deployment package source location that you specify cannot be used by another software deployment package.

The SMS Provider 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. You should carefully restrict access to the download location in order to reduce the risk of attackers tampering with the software update source files.

You can change the package source location in the Deployment Package properties after Configuration Manager creates the deployment package. But if you do so, you must first copy the content from the original package source to the new package source location.

* + **Sending priority**: *Specify the sending priority for the deployment package. Configuration Manager uses the sending priority for the deployment package when it sends the package to distribution points. Deployment packages are sent in priority order: High, Medium, or Low. Packages with identical priorities are sent in the order in which they were created. If there is no backlog, the package will process immediately regardless of its priority.*

1. On the **Distribution Points** page, specify the distribution points or distribution point groups that will host the software update files. For more information about distribution points, see [Distribution point configurations](https://docs.microsoft.com/en-us/sccm/core/servers/deploy/configure/install-and-configure-distribution-points#bkmk_configs).

Note

This page is available only when you create a new software update deployment package.

1. On the **Download Location** page, specify whether to download the software update files from the Internet or from your local network. Configure the following settings:
   * **Download software updates from the Internet**: *Select this setting to download the software updates from a specified location on the Internet. This setting is enabled by default.*
   * **Download software updates from a location on the local network**: *Select this setting to download the software updates from a local directory or shared folder. This setting is useful when the computer that runs the wizard does not have Internet access. Any computer with Internet access can preliminarily download the software updates and store them in a location on the local network that is accessible from the computer that runs the wizard.*
2. On the Language Selection page, select the languages for which the selected software updates are downloaded. The software updates are downloaded only if they are available in the selected languages. Software updates that are not language specific are always downloaded. By default, the wizard selects the languages that you have configured in the software update point properties. At least one language must be selected before proceeding to the next page. When you select only languages that are not supported by a software update, the download will fail for the software update.
3. On the Summary page, review the settings. To save the settings to a deployment template, click **Save As Template**, enter a name and select the settings that you want to include in the template, and then click **Save**. To change a configured setting, click the associated wizard page and change the setting.

Note

The template name can consist of alphanumeric ASCII characters as well as **\** (backslash) or **‘** (single quotation mark).

1. Click **Next** to create the ADR.

After you have completed the wizard, the ADR will run. It will

1. Add the software updates that meet the specified criteria to a software update group
2. Download the software updates to the content library on the site server
3. Distribute the software updates to the configured distribution point(s)
4. Deploy the software update group to clients in the target collection

## Add a new Deployment to an existing ADR

After you create an ADR, you can add additional deployments to the rule. This can help you manage the complexity of deploying different updates to different collections. Each new deployment has the full range of functionality and deployment monitoring experience.

1. In the Configuration Manager console, navigate to **Software Library** > **Overview** > **Software Updates** > **Automatic Deployment Rules**, and then select the desired rule.
2. On the **Home** tab, in the **Automatic Deployment Rule** group, click **Add Deployment**. The Add Deployment Wizard opens.
3. On the **Collection** page, configure the following settings:
   * **Collection**: Specifies the target collection to be used for the deployment. Members of the collection receive the software updates that are defined in the deployment.
   * **Enable the deployment after this rule is run**: Specify whether to enable the software update deployment after the ADR runs. Regarding this specification, consider the following:
     + When you enable the deployment, the software updates that meet the criteria defined in the rule are added to a software update group, the software update content is downloaded as necessary, the content is copied to the specified distribution points, and the software updates are deployed to the clients in the target collection.
     + When you do not enable the deployment, the software updates that meet the criteria defined in the rule are added to a software update group and the software updates deployment policy is configured but the software updates are not downloaded or deployed to clients. This situation provides you time as needed to prepare to deploy the software updates, verify that the software updates that meet the criteria are adequate, and then enable the deployment at a later time.
4. On the Deployment Settings page, configure the following settings:
   * **Use Wake-on-LAN to wake up clients for required deployments**: Specifies whether to enable Wake On LAN at the deadline to send wake-up packets to computers that require one or more software updates in the deployment. Any computers that are in sleep mode at the installation deadline time will be awakened so the software update installation can initiate. Clients that are in sleep mode that do not require any software updates in the deployment are not started. By default, this setting is not enabled.

Note

Before you can use this option, you must configure computers and networks for “Wake-On-LAN”

* + **Detail level**: Specify the level of detail for the state messages that are reported by client computers.

Note

When you deploy definition updates, set the detail level to **Error only** to have the client report a state message only when a definition update fails to be delivered to the client. Otherwise, the client will report a large number of state messages that might impact performance on the site server.

1. On the Deployment Schedule page, configure the following settings:
   * **Schedule evaluation**: Specify whether Configuration Manager evaluates the available time and installation deadline times by using UTC or the local time of the computer that runs the Configuration Manager console.

Note

When you select local time, and then select **As soon as possible** for the **Software available time** or **Installation deadline**, the current time on the computer running the Configuration Manager Console is used to evaluate when updates are available or when they are installed on a client. If the client is in a different time zone, these actions will occur when the client's time reaches the evaluation time.

* + **Software available time**: Select one of the following settings to specify when the software updates are available to clients:
    - **As soon as possible**: Select this setting to make the software updates that are included in the deployment available to the client computers as soon as possible. When you create the deployment with this setting selected, Configuration Manager updates the client policy. Then, at the next client policy polling cycle, clients become aware of the deployment and can obtain the updates that are available for installation.
    - **Specific time**: Select this setting to make the software updates that are included in the deployment available to the client computers at a specific date and time. When you create the deployment with this setting enabled, Configuration Manager updates the client policy. Then, at the next client policy polling cycle, clients become aware of the deployment. However, the software updates in the deployment are not available for installation until after the configured date and time.
  + **Installation deadline**: Select one of the following settings to specify the installation deadline for the software updates in the deployment:
    - **As soon as possible**: Select this setting to automatically install the software updates in the deployment as soon as possible.
    - **Specific time**: Select this setting to automatically install the software updates in the deployment at a specific date and time. Configuration Manager determines the deadline to install software updates by adding the configured **Specific time** interval to the **Software available time**.

Note

The actual installation deadline time is the displayed deadline time plus a random amount of time up to 2 hours. This reduces the potential impact of all client computers in the destination collection installing the software updates in the deployment at the same time.

You can configure the **Computer Agent** client setting **Disable deadline randomization** to disable the installation randomization delay for required software updates. For more information, see [Computer Agent](https://docs.microsoft.com/en-us/sccm/core/clients/deploy/about-client-settings#computer-agent).

1. On the User Experience page, configure the following settings:
   * **User notifications**: Specify whether to display notification of the software updates in Software Center on the client computer at the configured **Software available time** and whether to display user notifications on the client computers.
   * **Deadline behavior**: Specify the behavior that is to occur when the deadline is reached for the software update deployment. Specify whether to install the software updates in the deployment. Also specify whether to perform a system restart after software update installation regardless of a configured maintenance window. For more information about maintenance windows, see [How to use maintenance windows](https://docs.microsoft.com/en-us/sccm/core/clients/manage/collections/use-maintenance-windows).
   * **Device restart behavior**: Specify whether to suppress a system restart on servers and workstations after software updates are installed and a system restart is required to complete the installation.

Important

Suppressing system restarts can be useful in server environments or in cases in which you do not want the computers that are installing the software updates to restart by default. However, doing so can leave computers in an insecure state, whereas allowing a forced restart helps to ensure immediate completion of the software update installation.

* + **Write filter handling for Windows Embedded devices**: When you deploy software updates to Windows Embedded devices that are write filter enabled, you can specify to install the software update on the temporary overlay and either commit changes later or commit the changes at the installation deadline or during a maintenance window. When you commit changes at the installation deadline or during a maintenance window, a restart is required and the changes persist on the device.

Note

When you deploy a Software Update to a Windows Embedded device, make sure that the device is a member of a collection that has a configured maintenance window.

1. On the Alerts page, configure how Configuration Manager and System Center Operations Manager will generate alerts for this deployment.

You can review recent software updates alerts from the **Software Updates** node in the **Software Library** workspace.

1. On the Download Settings page, configure the following settings:
   * Specify whether the client will download and install the software updates when a client is connected to a slow network or is using a fallback content location.
   * Specify whether to have the client download and install the software updates from a fallback distribution point when the content for the software updates is not available on a preferred distribution point.
   * **Allow clients to share content with other clients on the same subnet**: Specify whether to enable the use of BranchCache for content downloads. For more information about BranchCache, see [Concepts for content management](https://docs.microsoft.com/en-us/sccm/core/plan-design/hierarchy/fundamental-concepts-for-content-management#branchcache).
   * **If software updates are not available on distribution point in current, neighbor or site groups, download content from Microsoft Updates**: Select this setting to have clients that are connected to the intranet download software updates from Microsoft Update if software updates are not available on distribution points. Internet-based clients can always go to Microsoft Update for software updates content.
   * Specify whether to allow clients to download after an installation deadline when they use metered Internet connections. Internet providers sometimes charge by the amount of data that you send and receive when you are on a metered Internet connection.

Note

Clients request the content location from a Management Point for the Software Updates in a deployment. The download behaviour depends upon how you have configured the Distribution Point, deployment package, and the settings on this page. For more information, see [Content source location scenarios](https://docs.microsoft.com/en-us/sccm/core/plan-design/hierarchy/content-source-location-scenarios).

# Maintenance Windows

Maintenance windows enable you to define a time when Configuration Manager Operations can be carried out on a device collection. You can use Maintenance Windows to help ensure that client configuration changes occur during periods that do not affect productivity.

The following operations support Maintenance Windows:

* Software deployments
* Software Update deployments
* Compliance settings deployment and evaluation
* Operating System Deployments
* Task Sequence deployments

Configure maintenance windows with a start date, a start and finish time, and a recurrence pattern. The maximum duration of a window has to be less than 24 hours.

**By default,** computer restarts caused by a deployment are not allowed outside of a maintenance window, but you can override the default. Maintenance windows affect only the time when the deployment program runs

When a client computer is a member of a device collection that has a Maintenance Window, a Deployment program runs only if the maximum allowed run time does not exceed the duration configured for the window. If the program fails to run, an alert is generated and the deployment is rerun during the next scheduled maintenance window that has available time

## Using multiple Maintenance Windows

When a client computer is a member of multiple device collections that have Maintenance Windows, these rules apply:

* If the Maintenance Windows do not overlap, they are treated as two independent maintenance windows.
* If the Maintenance Windows overlap, they are treated as a single maintenance window encompassing the time period covered by both maintenance windows. For example, if two windows, each an hour in duration overlap by 30 minutes, the effective duration of the Maintenance Window would be 90 minutes.
* When a user initiates an Application installation from Software Center, the application is installed immediately, regardless of any maintenance windows.
* If an Application deployment with a purpose of **Required** reaches its installation deadline during the nonbusiness hours configured by a user in Software Center, the application will be installed.

## How to configure Maintenance Windows

In the **Configuration Manager console**, choose **Assets and Compliance**> **Device Collections**.

* In the **Device Collections** list, select a collection. You cannot create maintenance windows for the **All Systems** collection.
* On the **Home** tab, in the **Properties** group, choose **Properties**.
* In the **Maintenance Windows** tab of the **<collection name> Properties** dialog box, choose the **New** icon.
* Complete the **<new> Schedule** dialog.
* Make a selection from the **Apply this schedule to** drop-down list.
* Choose **OK** and then close the **<collection name> Properties** dialog box.

# Using WMI and PowerShell to manage SCCM Collections

## Overview

SC Configuration Manager is heavily reliant on Windows Management Instrumentation (WMI) to allow interaction with Configuration Manager data. On the client side, almost all Configuration Manager Information (such as policies and content locations) are stored in WMI.

On the SCCM Server we make extensive use of WMI via a special role called the SMS Provider. The purpose of the SMS Provider is to interact with the database and translate between WMI Query Language (WQL) and SQL.

All of our security is enforced by the SMS Provider, and it’s where all of our consoles connect to perform Administrative functions. Starting in Configuration Manager 2012, we can have multiple SMS Providers per site, which is useful if we are making heavy use of the SMS Provider through automation.

## SMS Provider Role in SCCM

Functionally, the SMS Provider organizes objects into WMI classes, with each class representing a different type of information for Configuration Manager. In Configuration Manager, there are classes for Collections, Software Update Groups, Clients, Task Sequence Steps and so forth. If it exists within SCCM, there is a class for it.

The SMS Provider is comprised of a single Primary Namespace that we want to connect to: ***root\sms\site\_<sitecode>.***

## Using native WMI to create Collection query rules

We can use WMI to perform SCCM tasks very easily

**New-CMDeviceCollection** - Creates a new Configuration Manager Device Collection

**Add-CMDeviceCollectionQueryMembershipRule** – Adds a rule that must be satisfied in order for a device to be a member of the new Collection

**New-CMSchedule** – Sets the time the Collection can be acted upon

## Using .NET Type Accelerators

The same result can also be obtained by using .NET Type Accelerators

Using this method allows us to get access directly to .NET WMI methods via PowerShell:

$CMCollection = ([WMIClass]”\\SMSPROVIDER\root\sms\SITE\_ID:SMS\_Collection”).CreateInstance()

First, we’re creating a new object of the WMIClass type: **NewCMCollection**.

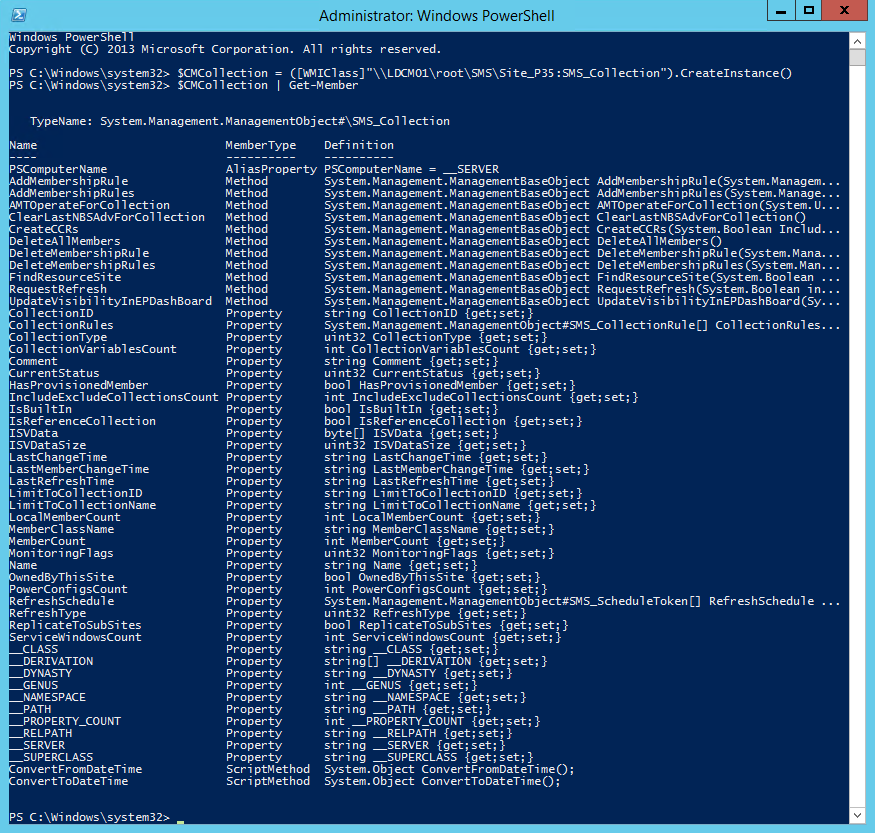
We are then assigning a new instance of the **SMS\_Collection** class.

Note

This method also exposes more properties and methods than using the **New-CMDeviceCollection** command with native WMI

We can see the properties by using the command Get-Member against the new Collection

$CMCollection | Get-Member

[](https://msdnshared.blob.core.windows.net/media/TNBlogsFS/prod.evol.blogs.technet.com/CommunityServer.Blogs.Components.WeblogFiles/00/00/00/76/18/hsg-2-28-14-1.png)

We see a number of properties are available for us to configure. However, you’ll see that we have additional methods available. These will change from WMI class to WMI class, but this shows how Configuration Manager exposes common functionality.

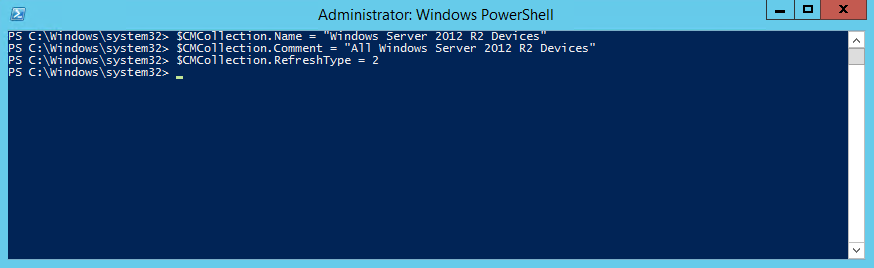
We can now set properties on the new Collection – again with PowerShell

$CMCollection.Name = “All Windows 2012 R2 Devices”

$CMCollection.Comment = “This collection contains all Windows Server 2012 R2 Devices”

$CMCollection.LimitToCollectionID = “SMS00001”

$CMCollection.RefreshType = 2

[](https://msdnshared.blob.core.windows.net/media/TNBlogsFS/prod.evol.blogs.technet.com/CommunityServer.Blogs.Components.WeblogFiles/00/00/00/76/18/hsg-2-28-14-2.png)

Commit the changes using the command $CMCollection.Put()

## Membership Rules

A rule needs to be created to pass to the Collection, so it is aware of what needs to be satisfied by a potential member in order to join

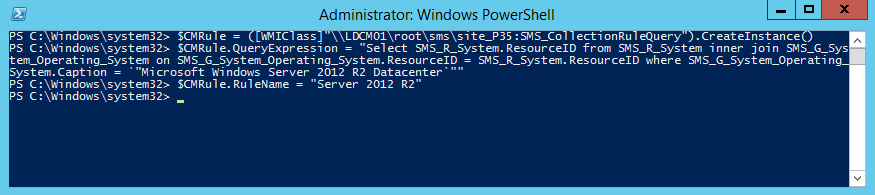
In this case, it requires an object of the **SMS\_CollectionRule** class

$CMRule = ([WMIClass]”\SMSPROVIDER\root\sms\SITE\_ID:SMS\_CollectionRuleQuery”).CreateInstance()

$CMRule.QueryExpression=”select SMS\_R\_SYSTEM.ResourceID,SMS\_R\_SYSTEM.ResourceType,SMS\_R\_SYSTEM.Name,SMS\_R\_SYSTEM.SMSUniqueIdentifier,SMS\_R\_SYSTEM.ResourceDomainORWorkgroup,SMS\_R\_SYSTEM.Client from SMS\_R\_System inner join SMS\_G\_System\_OPERATING\_SYSTEM on SMS\_G\_System\_OPERATING\_SYSTEM.ResourceId = SMS\_R\_System.ResourceId where SMS\_G\_System\_OPERATING\_SYSTEM.Caption = `"Microsoft Windows Server 2012 R2 Datacenter`"”

$CMRule.RuleName = “Windows Server 2012”

Now there is a **CollectionRuleQuery** object that represents a query-based collection rule. With this, we can set the **QueryExpression**, **QueryID**, and **RuleName** properties:

[](https://msdnshared.blob.core.windows.net/media/TNBlogsFS/prod.evol.blogs.technet.com/CommunityServer.Blogs.Components.WeblogFiles/00/00/00/76/18/hsg-2-28-14-3.png)

The new Membership Rule can now be added to the new Collection:

$CMCollection.AddMembershipRule($CMRule)

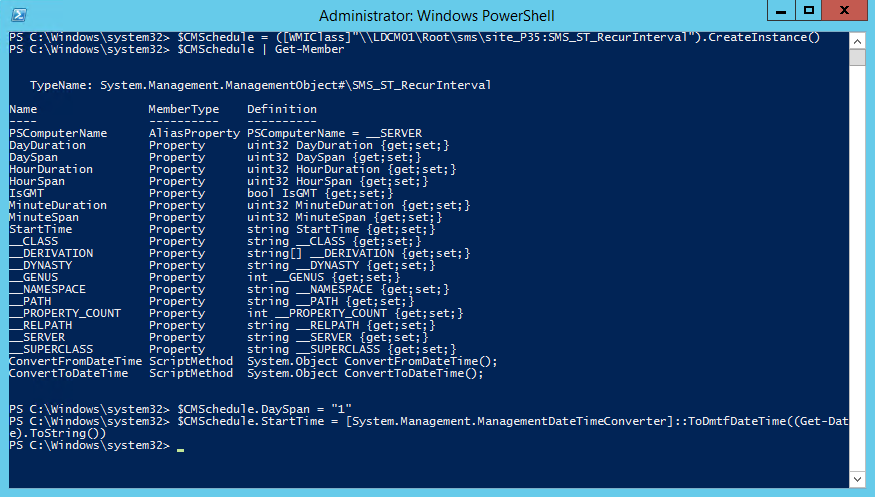
## Refresh Schedules

We can now set a refresh schedule so the query updates on a recurring schedule. Schedule tokens are used in several places in Configuration Manager, such as Collections and Maintenance Windows. To set a refresh schedule, we need to create a schedule token:

$CMSchedule = ([WMIClass]"\\ SMSPROVIDER\root\sms\SITE\_ID:SMS\_ST\_RecurInterval").CreateInstance()

$CMSchedule.DaySpan = “1”

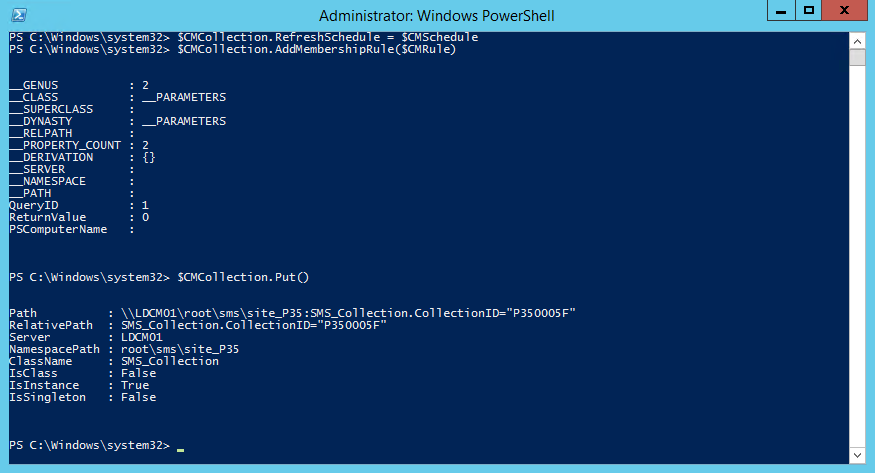
$CMSchedule.StartTime = [System.Management.ManagementDateTimeConverter]::ToDmtfDateTime((Get-Date).ToString())

[](https://msdnshared.blob.core.windows.net/media/TNBlogsFS/prod.evol.blogs.technet.com/CommunityServer.Blogs.Components.WeblogFiles/00/00/00/76/18/hsg-2-28-14-4.png)

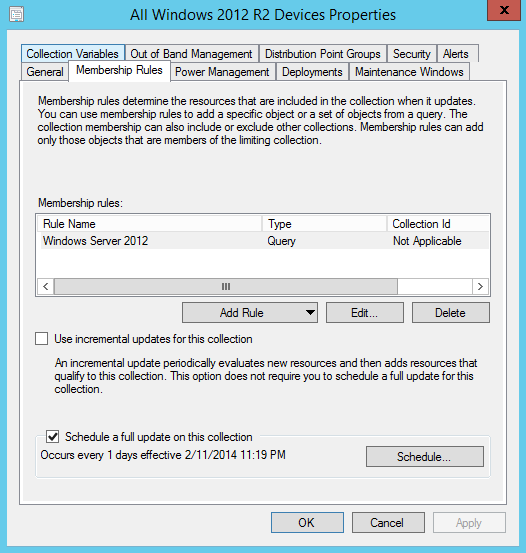
Now that there is a Schedule Token, it can be used it to set the **RefreshSchedule** property:

$CMCollection.RefreshSchedule=$CMSchedule

Commit the changes using the command $CMCollection.Put()

[](https://msdnshared.blob.core.windows.net/media/TNBlogsFS/prod.evol.blogs.technet.com/CommunityServer.Blogs.Components.WeblogFiles/00/00/00/76/18/hsg-2-28=14=5.png)

**As you can see, we now have a collection with a query-based rule and a refresh schedule:**

[](https://msdnshared.blob.core.windows.net/media/TNBlogsFS/prod.evol.blogs.technet.com/CommunityServer.Blogs.Components.WeblogFiles/00/00/00/76/18/hsg-2-28-14-6.png)

# Appendix

[System Center 2012 R2 Configuration Manager SDK](http://msdn.microsoft.com/en-us/library/hh948960.aspx).

Publically provided code examples make use of the documented WMI calls that are available via the SDK

[SMS\_Collection Server WMI Class](http://msdn.microsoft.com/en-us/library/hh948939.aspx)

The same data documented and ready for consumption.

SMSProv.log. This log will gives a good idea about the WMI query being executed and what classes and methods are used for various operations within the console

# Embedded classes and lazy properties in WMI

Unfortunately Microsoft doesn’t give us a cmdlet to just create a new Software Update group, so we have to do it ourself.

I’m still learning a lot of Powershell and those embedded properties in WMI were quite a challenge for me.

In order to create a new Software Update Group one has to use the WMI class SMS\_

For example the Software Update Group’s name is a lazy property and if you want to set it you have to do it via another embedded class ([MSDN: SMS\_CI\_LocalizedProperties](http://msdn.microsoft.com/en-us/library/cc145662.aspx)).

If you want to know what a lazy property is, have a look at Trevor Sullivan’s blog article about those weirdos: [http://trevorsullivan.net/2010/09/28/powershell-configmgr-wmi-provider-feat-lazy-properties/](https://david-obrien.net/2012/12/create-a-new-software-update-group-in-configmgr/%22http:/trevorsullivan.net/2010/09/28/powershell-configmgr-wmi-provider-feat-lazy-properties/%22%20http:/trevorsullivan.net/2010/09/28/powershell-configmgr-wmi-provider-feat-lazy-properties/)

For the stuff we want to do we need the following WMI classes:

* SMS\_SoftwareUpdate
* SMS\_CI\_LocalizedProperties
* SMS\_

# Find the updates and link them

What we’re going to do is the following.

First we need to find the future members of our new Update Group. That was a bit tricky at first, because I had to find a property that’s universal, meaning that can be used on every system. All the definitions and names are all localized, so they can’t be used. I chose to Knowledge Base ID.

## How do I find the Knowledge Base ID?

Every update by Microsoft is published on their site, for example a Dot Net 3.5 SP1 hotfix for Windows 7 can be found here: [http://support.microsoft.com/kb/976462](https://david-obrien.net/2012/12/create-a-new-software-update-group-in-configmgr/%22http:/support.microsoft.com/kb/976462%22%20http:/support.microsoft.com/kb/976462)

For our script to run you’ll need the number at the end of the URL, 976462.

With this ID the script can now parse the SMS\_SoftwareUpdate class and find the CI\_ID we need to link this update to our new Update Group we’re about to create.

(gwmi -ns root\sms\site\_$($SiteCode) -class SMS\_SoftwareUpdate | where {$\_.ArticleID -eq $KBID }).CI\_ID

# Name the Update Group

Looking at the SMS\_

* **\*\*LocalizedDescription**\*\*
* **\*\*LocalizedDisplayName**\*\*
* **\*\*LocalizedInformativeURL**\*\*
* **\*\*LocalizedPropertyLocaleID**\*\*

Cool, DisplayName, absolutely what we want! Unfortunately, this property is read-only. Damn!!!

What now?

The property **LocalizedInformation** is read/write and seems to be able of what we want to achieve, naming the group. This property consists of an embedded WMI class named SMS\_CI\_LocalizedProperties.

Filling in the properties isn’t that difficult, you’ll see.

# Create the Update Group

This one is easy again. Create an instance of the SMS\_

This is the whole script:

<#

Functionality: This script creates a new Software Update Group in Microsoft System Center 2012 Configuration Manager

How does it work: create-SoftwareUpdateGroup.ps1 -UpdateGroupName $Name -KnowledgeBaseIDs $KBID -SiteCode

KnowledgeBaseID can contain comma separated KnowledgeBase IDs like 981852,16795779

Date: 02.12.2012

#>

param (

[string]$UpdateGroupName,

[array]$KnowledgeBaseIDs,

[string]$SiteCode

)

Function create-Group {

[array]$CIIDs = @()

foreach ($KBID in $KnowledgeBaseIDs)

{

$KBID

$CIID = (gwmi -ns root\sms\site\_$($SiteCode) -class SMS\_SoftwareUpdate | where {$\_.ArticleID -eq $KBID }).CI\_ID

$CIIDs += $CIID

}

$SMS\_CI\_LocalizedProperties = "SMS\_CI\_LocalizedProperties"

$class\_Localization = [wmiclass]""

$class\_Localization.psbase.Path ="ROOT\SMS\Site\_$($SiteCode):$($SMS\_CI\_LocalizedProperties)"

$Localization = $class\_Localization.CreateInstance()

$Localization.DisplayName = $UpdateGroupName

$Localization.LocaleID = 1033

$Description += $Localization

$SMS

$class\_AuthList = [wmiclass]""

$class\_AuthList.psbase.Path ="ROOT\SMS\Site\_$($SiteCode):$($SMS

$AuthList = $class\_AuthList.CreateInstance()

$AuthList.Updates = $CIIDs

$AuthList.LocalizedInformation = $Description

$AuthList.Put() | Out-Null

}

create-Group

# How do you use it?

Save the script under what ever name you like and execute it like this:

create-SoftwareUpdateGroup.ps1 -UpdateGroupName $Name -KnowledgeBaseIDs $KBID –SiteCode $YourSiteCode

$Name is the name of the Update Group you’re about to create.

$KBID is an array of the Knowledge Base IDs, e.g. 981852,16795779

$SiteCode is your Configuration Manager Site SiteCode

# What’s next?

This is version 0.1.

Following the completion of this version I had some more ideas that I like to put into the script, like using a text file/csv file with the KB IDs, rather than typing them on the command line.

The script needs error handling! Right now it won’t tell you what it did and if there was an error at some point.

Interaction: One KB can contain more than one hotfixes (e.g. KB890830). If the script finds more than one hotfix it should ask which one to add. Right now it adds all of them.