2/21/2024

Control My Update

v2.3 documentation

Grischa Ernst, Camille Debay
VMWARE GLOBAL INC

Reviewed by Katie Held

1. Table of Contents

1. General feature overview

Control My Update (CMU) is comprised of two different tools.

- Custom Profile Generator
  This graphical PowerShell tool will provide an easy-to-use GUI for configuring Windows Update profiles with the latest supported configuration settings.
  a. Create Windows Update profiles that are supported from 20H2 and later
  b. Create Delivery Optimization profiles that are supported from 20H2 and later
  c. Create the configuration for the Control my update tool itself
- Control My Update
  This tool will take control over Windows Update and will provide an end-to-end solution for patching and reporting.
  a. Select the source of updates (WSUS or Microsoft Update)
  b. Define Maintenance Windows to install updates only during this time
  c. Download all required updates before the installation starts – this will reduce the installation time
  d. Install emergency patches
  e. Create custom toast notifications
  f. Block specific KB's from installation

The complete solution is free to use and open source. Feel free to raise feature requests or other feedback.

2. Custom Profile Generator

Microsoft Windows configuration settings changed quite a lot over the past several years. The Custom Profile Generator (CPG) can be used to generate fully supported profiles for Windows 10 20H2 and later.
There are a lot of added, replaced or removed settings which will cause the need for cleanup of profiles and require using custom profiles to respect the best practices of Microsoft.

With the CPG, you can create three different profiles:

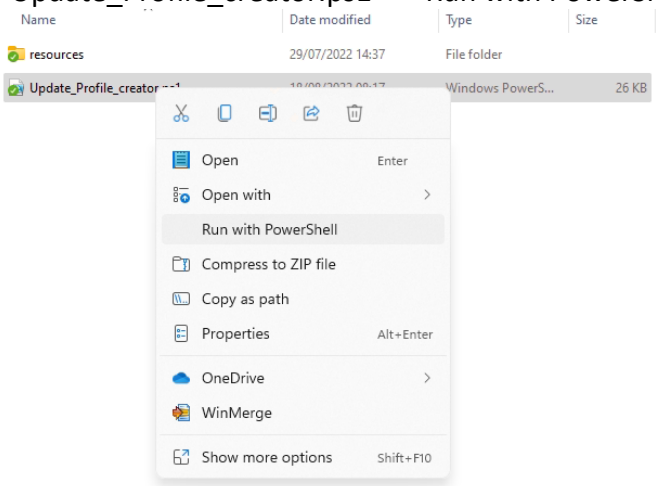
1. Windows Update
2. Delivery Optimization
3. Control My Update

The purpose of this guide is to provide full documentation (and clarification) of the settings, including which of the settings are set in the background.

WARNING: The Windows Update and Delivery Optimization profiles require Windows 10 20H2 or later!

2.1. How to start the GUI

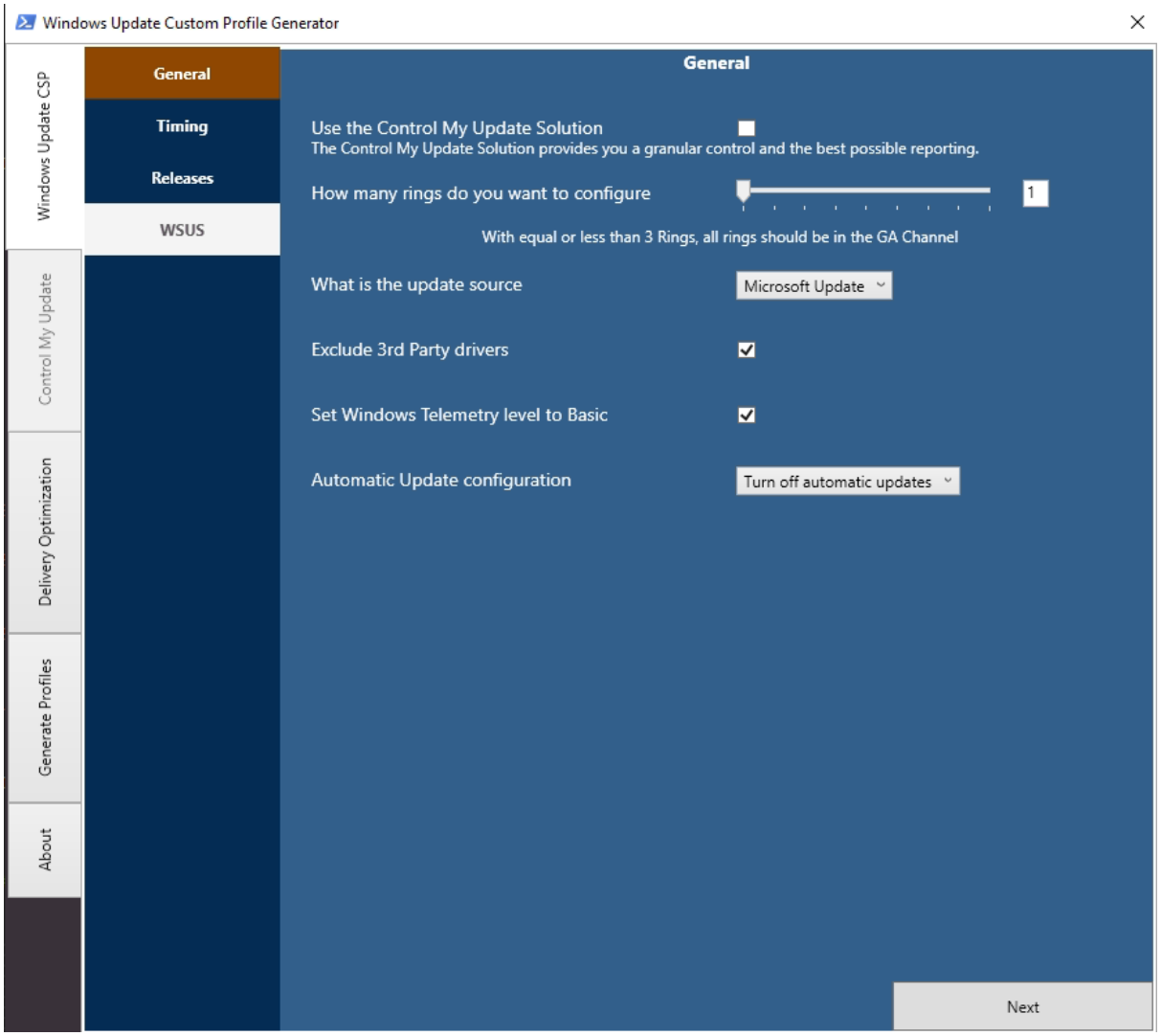
For opening the GUI, navigate to the folder “Profile Generator” and right click on “Update_Profile_creator.ps1” -> Run with PowerShell.



For this the PowerShell execution policy needs to be set on unrestricted (read https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_execution_policies?view=powershell-7.2 for more information).

Please be aware that only PowerShell 5.1 or higher on Windows are supported!

2.2. Windows Update



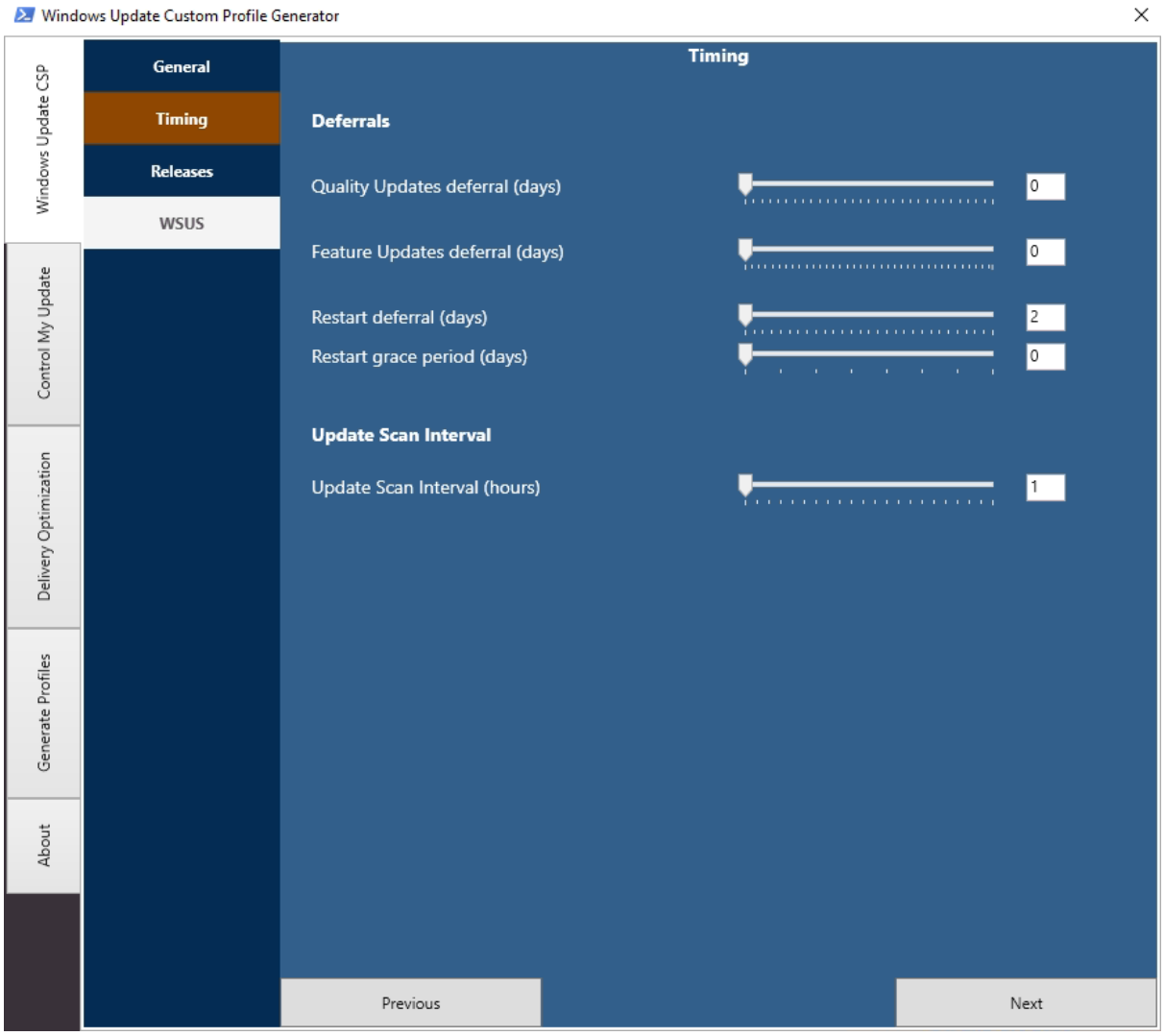
- Use the Control My Update Solution
  Select this option if you want to use the Control My Update Solution. If you just want to configure a Windows Update profile, please unselect this option.

- How many rings do you want to configure
  This will automatically multiply the deferral days. For example, you select 4 Rings and have a Quality Update deferral set to 7, the first ring has 7 days deferral, the second 14 days and so on.
  It will create several files for the different ring profiles.

- What is the update source
  Based on this selection the supported settings will be dis- or enabled.

- Exclude 3rd Party drivers
  If you don't want to install 3rd party drivers, you can select this option.

- Set Windows Telemetry level to Basic
  This setting is OPTIONAL – you don't need Windows telemetry anymore for Windows Update for business.
- Automatic Update configuration
  If you use the Control My Update solution, it will automatically disable the automatic update installation. If you don't want to use the solution, you can configure it based on your needs.



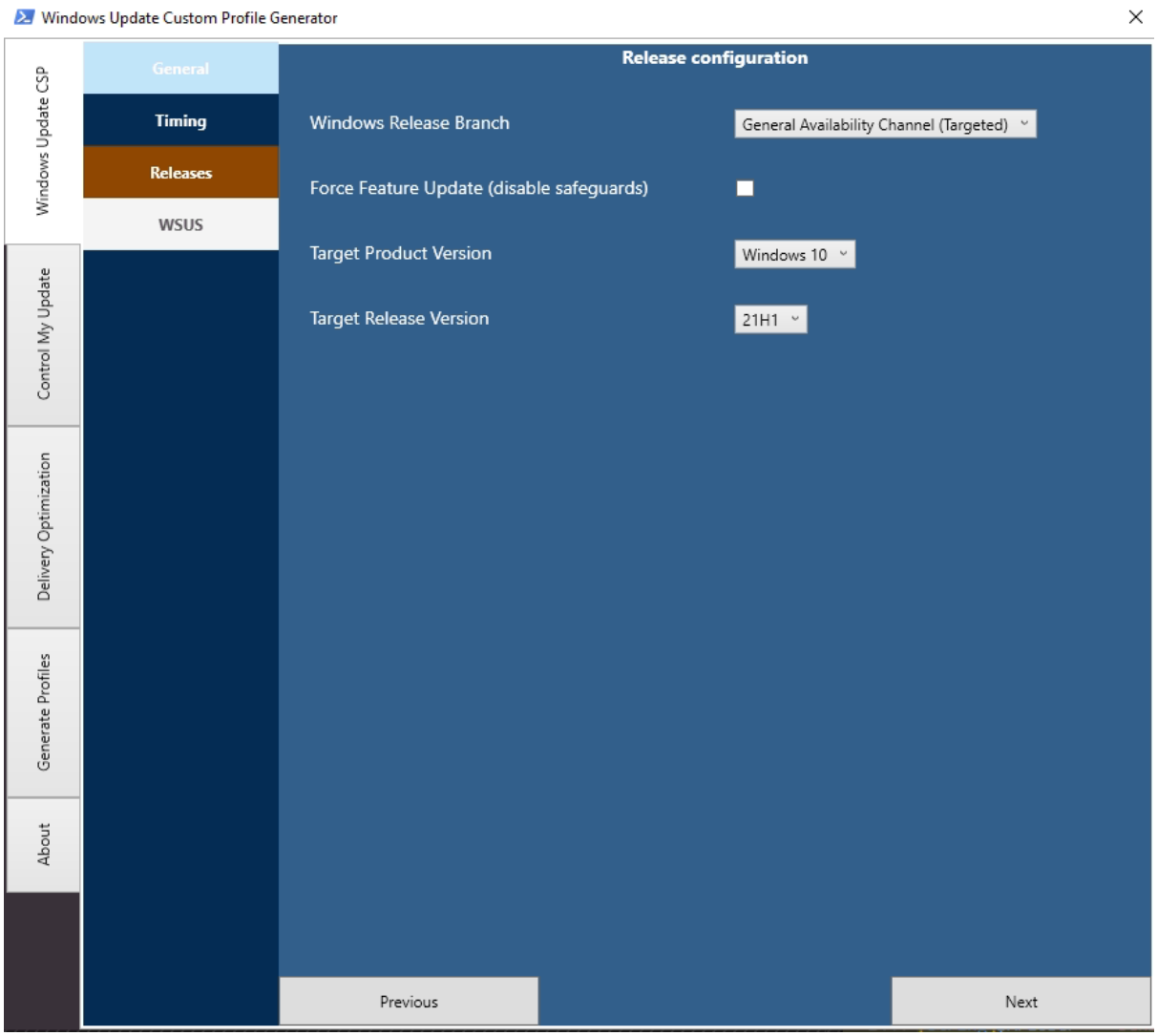
- Quality Update deferral (days)
  Based on the selection, quality updates getting deferred for x days.

- Feature Updates deferral (days)
  Based on the selection, feature updates getting deferred for x days.

- Restart deferral (days)
  Based on the selection, restart getting deferred for x days.

- Restart grace period (days)
  The grace period countdown starts from the time of the pending restart. The device will try to download and install the update at a time based on your other download and installation policies (the default is to automatically download and install in in the background). When the pending restart time is reached, the device will notify the user and try to update outside of active hours. Once the effective deadline is reached, the device will try to restart during active hours.

- Update Scan Interval (hours)
  Not applicable for Windows Update for Business (its always 22 hours).



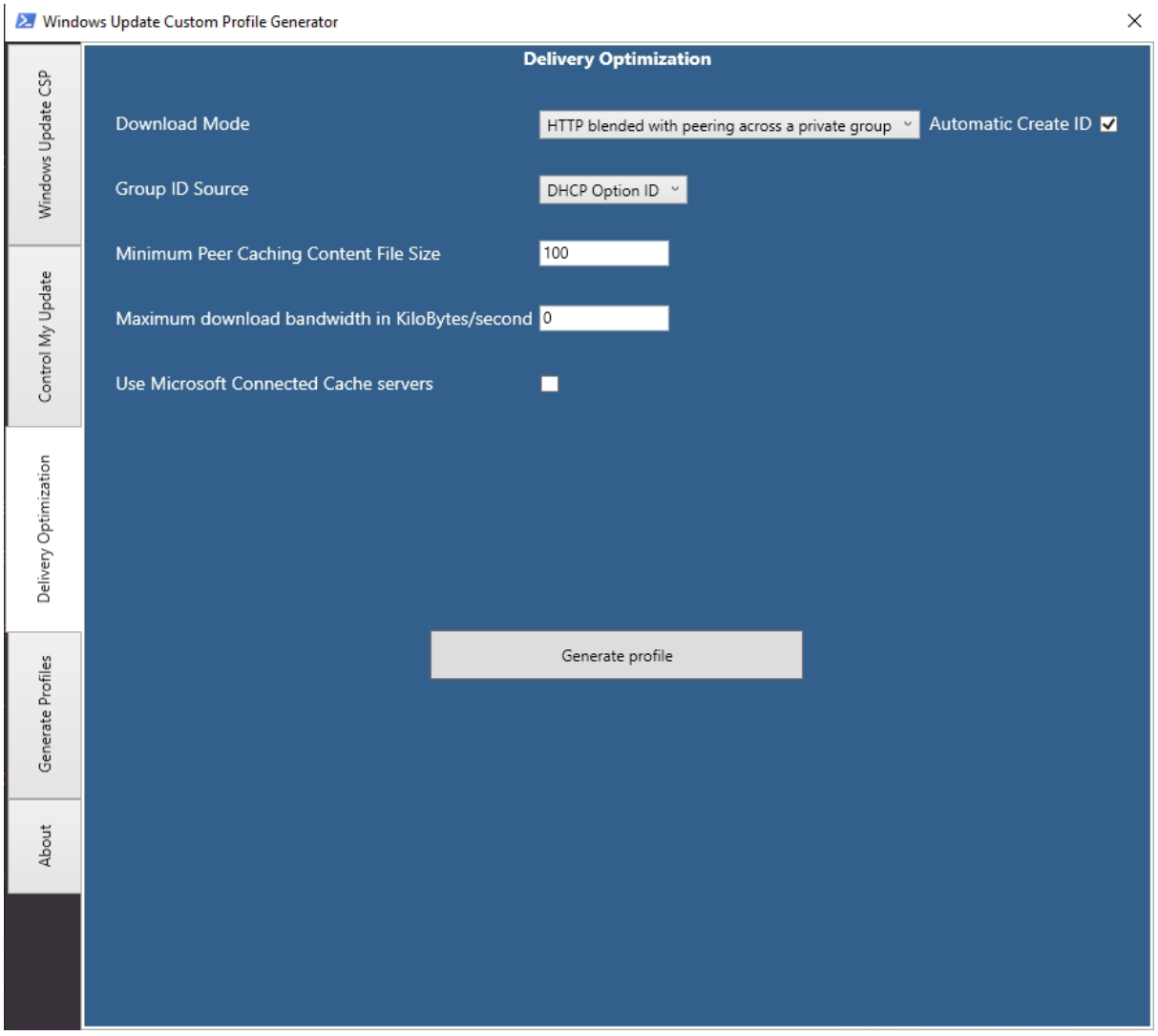
- Windows release branch
  Select the release branch – in case you create more than four rings, the ring 0 will be in Insider Slow – the rest in the selected branch.

- Force Feature Updates (disable safeguards)
  Disable safeguards helps to upgrade devices that are not 100% supported for the

feature update.

- Target Product Version
  Select the target product version. In case you want to update to Windows 11, you can use the product version and release version to upgrade the devices to Windows 11.
- Target Release Version
  Either you can use the target release version to make sure the device stays on a specific version, or the device updates to the selected version.

2.3. Delivery Optimization



- Download Mode
  Select the delivery optimization download mode

- Group ID Source
  If you want to use grouping and want to use the option to automatic get the group ID, choose the right source (make sure the group ID is configured properly).

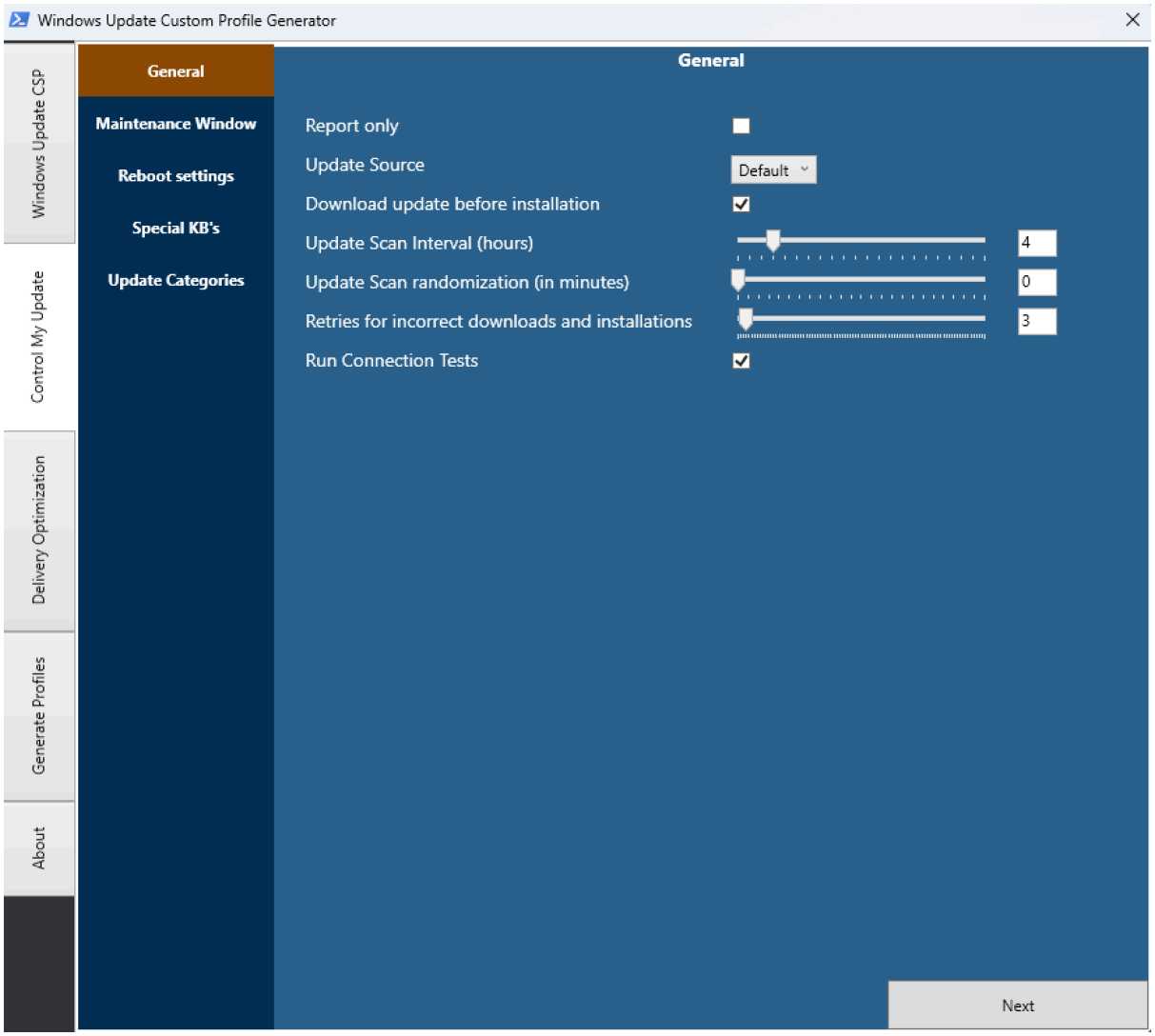
- Minimum Peer Caching content file size
  In case you want to change the minimum file size:
  Keep in mind that only four updates are shared on one device – if you select only 1MB then all updates are available for sharing and bigger updates are might not shared actively.

- Maximum download bandwidth in KiloBytes/second
  To limit the download speed, type in the maximum KB/s – 0 means unlimited.

- Use Microsoft connected cache servers
  Only supported for MEM customers.

2.4. Control My Update

The Control My Update (CMU) is the best-in-class Windows Update installation engine and provides enhanced features like blocking specific updates. All this is done via native Windows Update API's – there is no need for additional resources like PSWindowsUpdate



- Report only
  This will disable all other settings and will just use the CMU for getting the installed updates and writing them into registry. Together with Workspace ONE Sensors, this data can be collected and used in Workspace ONE Intelligence

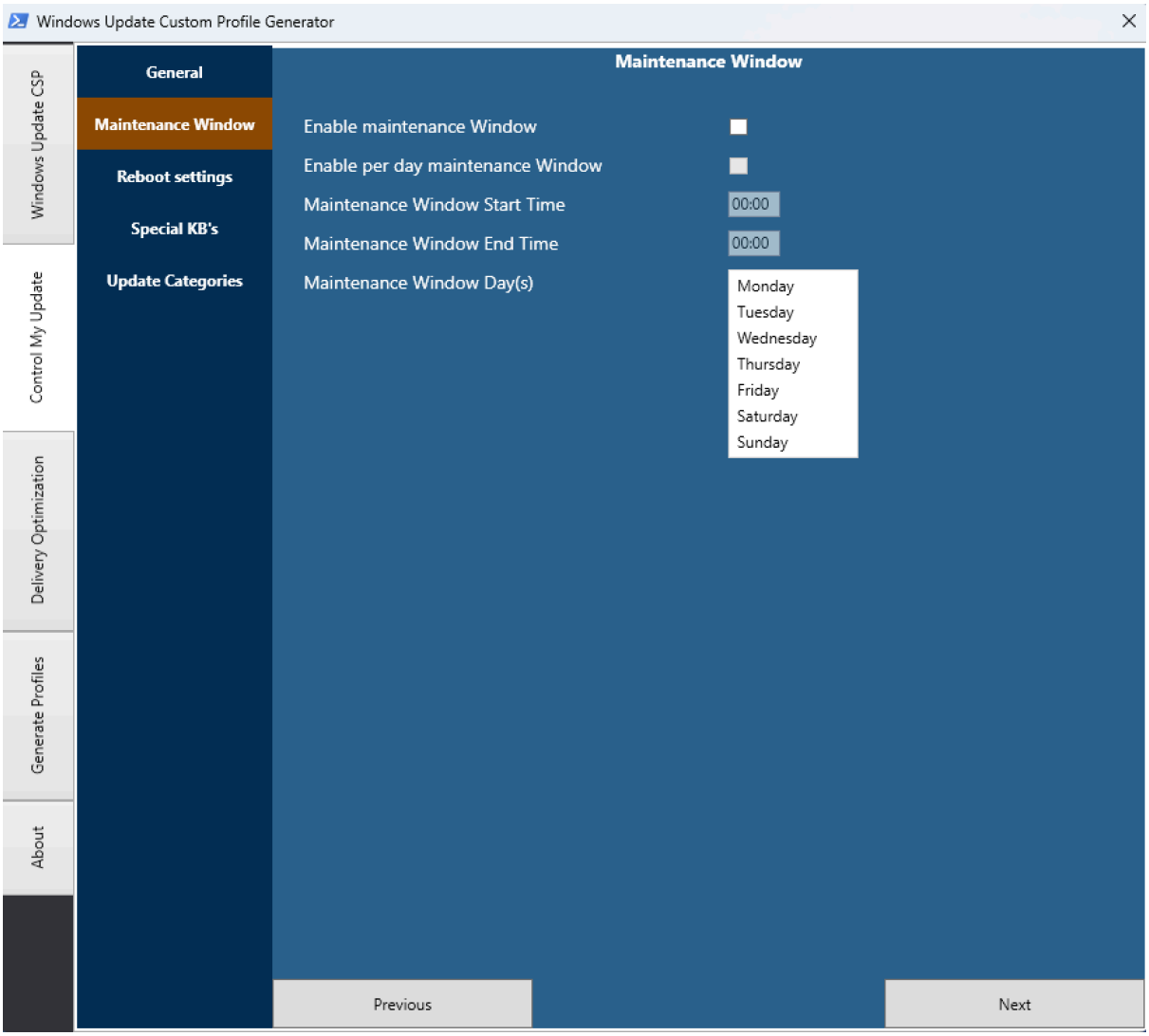
- Update Source
  You can select out of three options: Default, Microsoft Update and WSUS. Default will not change anything – Microsoft Update and WSUS will ignore the current setting that is configured on the system. For example, if your device is AD joined and you still have a GPO that sets the device to use WSUS, selecting Microsoft Update will overwrite this and the Windows Update Agent will use Microsoft Update. There will be no change in the registry or local policies – the CMU will just set the update source with every run.

- Download Update before installation
  To reduce the installation time, the updates can be downloaded before. This takes also affect when you use maintenance windows. Even outside of the maintenance window, the updates getting downloaded if you enable this option.

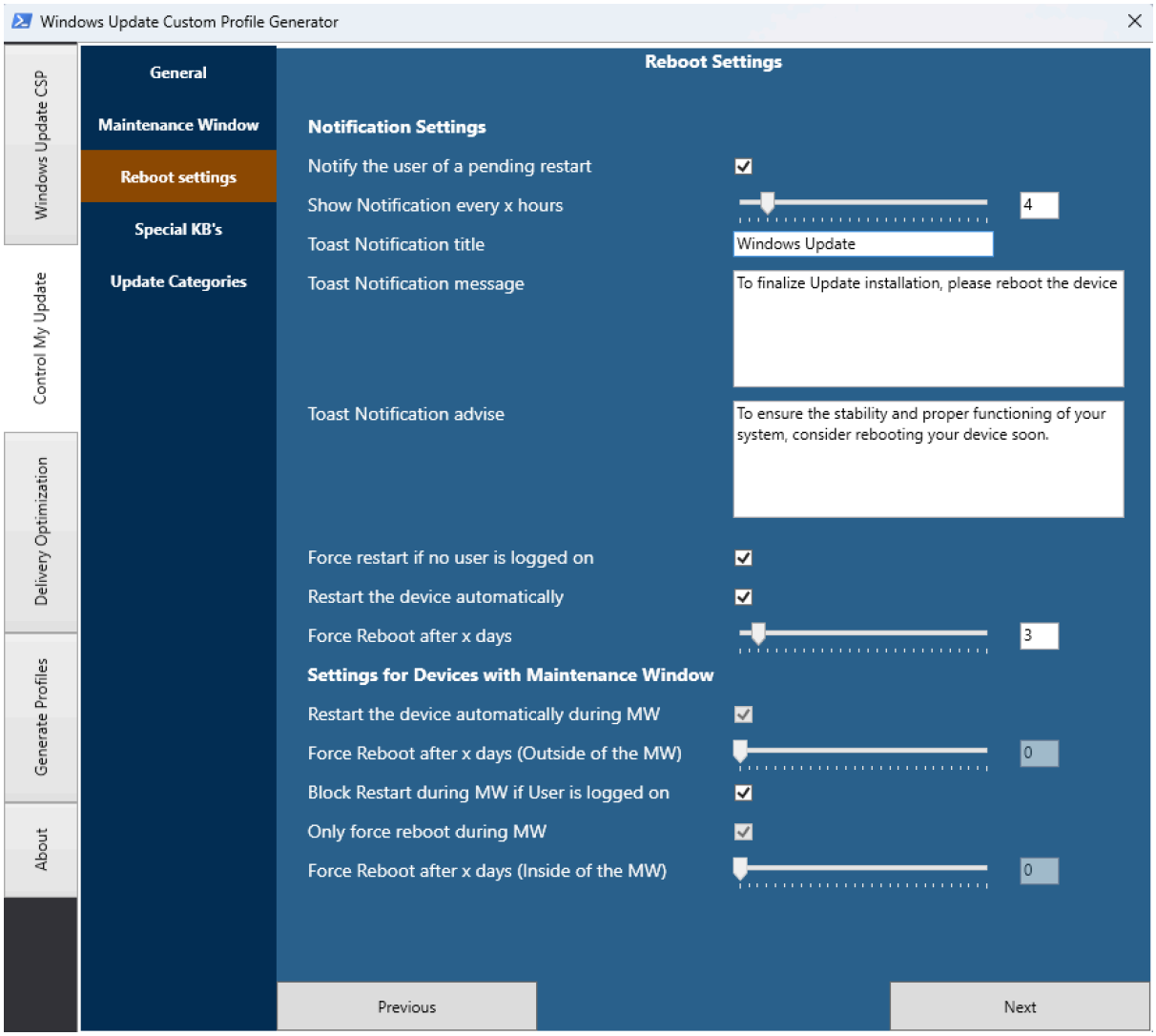
- Update scan Interval (hours)
  With this setting you can configure how often the device will search for new Updates.

- Update scan randomization (in minutes)
  The default update scan interval is set to one hour – if you want to increase it, you need to change the value in the “Windows Update” tab and then “Timing”.
  Scan randomization will add a random time on top of the scan interval. For example, if you have set the scan interval to 4 hours and the scan randomization to 60 minutes, the scan will be triggered between 4 and 5 hours. The randomization gets changed after every Windows Update scan.

- Retries for incorrect downloads and installations
  Define a number of retries for incorrect downloads and installations.
  For example, if you set the retry count to 3, the CMU will try to download and update 3 times if the download was not successful.



- Enable maintenance window
  If you want to use maintenance windows, you need first enable them and then select a time. If you only select a time but no day – the time is set for every day. You can select multiple days and can also use overnight maintenance windows where you start e.g., at 10pm on Monday and stop at 4am on Tuesday.
  Updates will be installed only during this time and the device will reboot automatically during maintenance windows.
  For the time – make sure you select a time in 24-hour format!



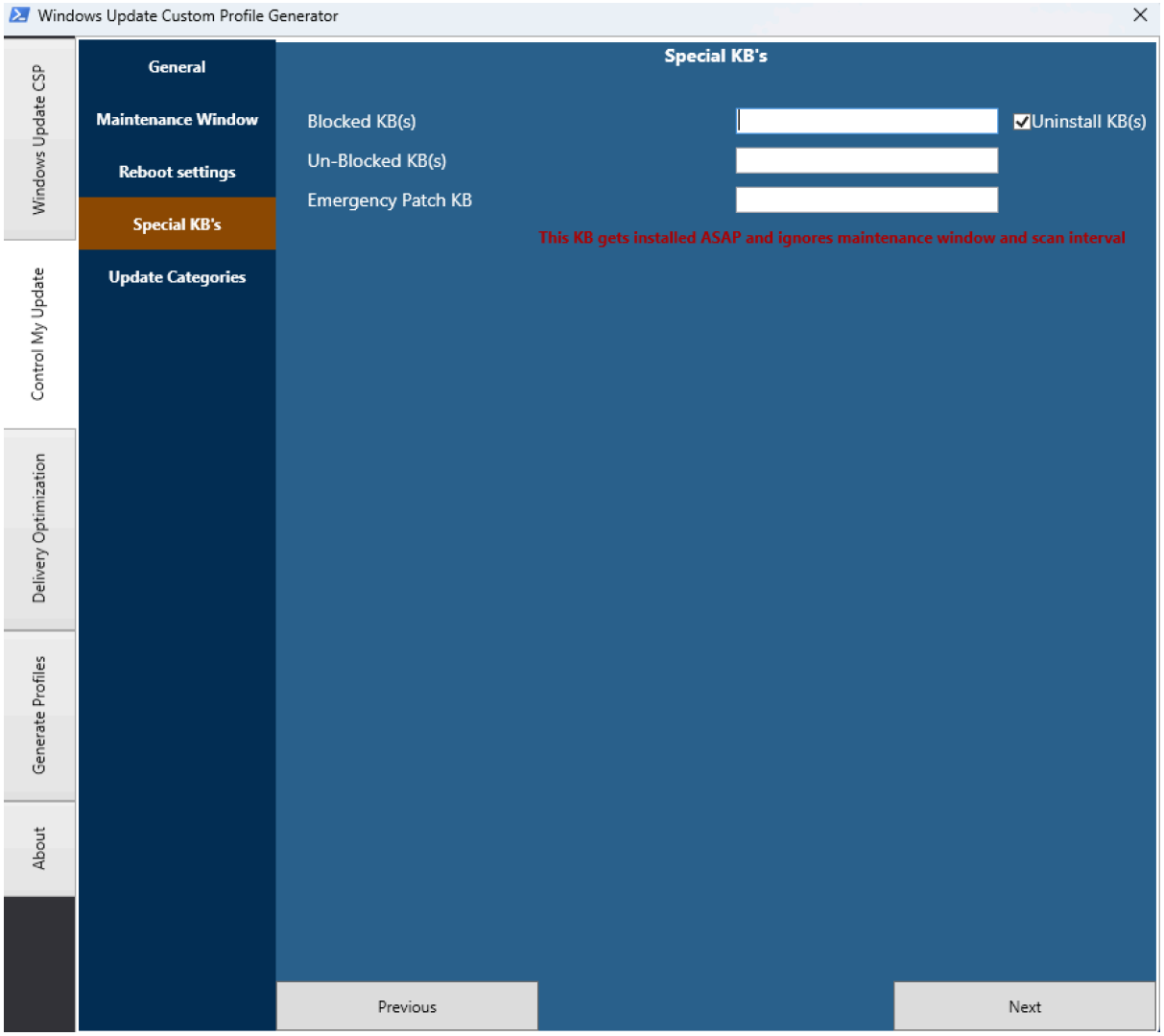
- Notify the user of a pending restart
  If you enable the notification, then the user will get a toast notification if the device needs to be rebooted to finish an update installation.

- Toast Notification title
  Define a title for the notification.

- Toast Notification message
  Define a text for the notification.

- Toast Notification advise
  Define details for the notification.

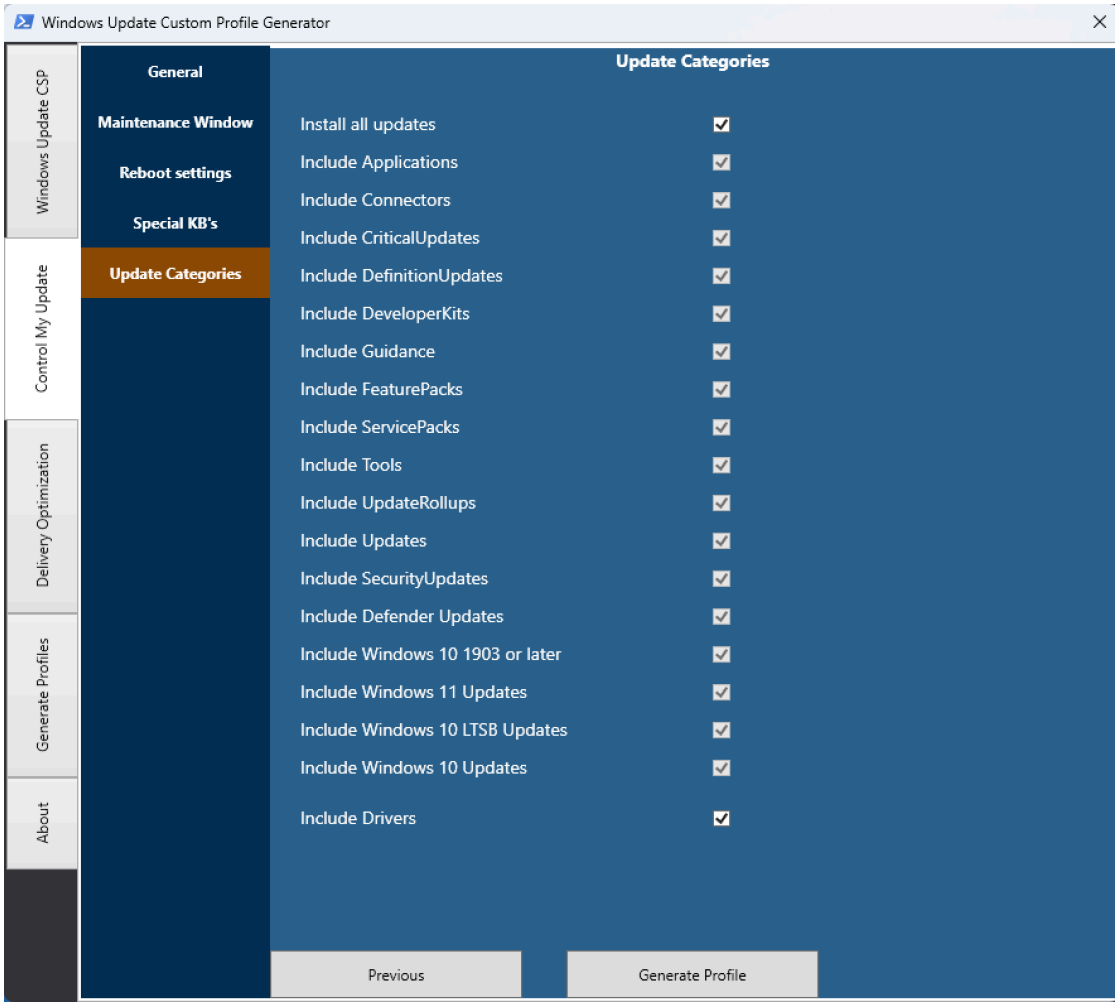
Restart the device automatically during MW
To make sure updates are installed completely some updates require a reboot. If you can restart the device automatically without user interaction, you can enable this setting and the device will always reboot during MW if a pending restart is detected.



- Blocked KB(s)
  Use this function to block specific KBs from installation. If, for example, the May CU causes issues on devices, you can add the KB article to the “Blocked KB(s)” and the update will not get installed. Check the “Uninstall KB(s)” to uninstall updates that were already installed.
  Multiple KBs can be specified and separated with a semicolon (e.g. KB123;KB456;KB789)
  Please note that if you block or uninstall one update, it may be that another update cannot be installed if the update has a dependency on the blocked update.

- Un-Blocked KB(s)
  To allow an update to get installed again, add the KB here. If the update is added in “Blocked KB(s)” and “Un-Blocked KB(s)”, the un-block will always win.

- Emergency Patch KB
  The emergency function can be used for high important patches. The KB that is added here, will be installed outside of the maintenance window and outside of the scan interval. You can only add one KB as emergency patch.
  The KB will only be installed if the update is available for installation on the device. It will not force an update to get installed if Windows Update does not offer the update.



- Update Categories

3. Control My Update

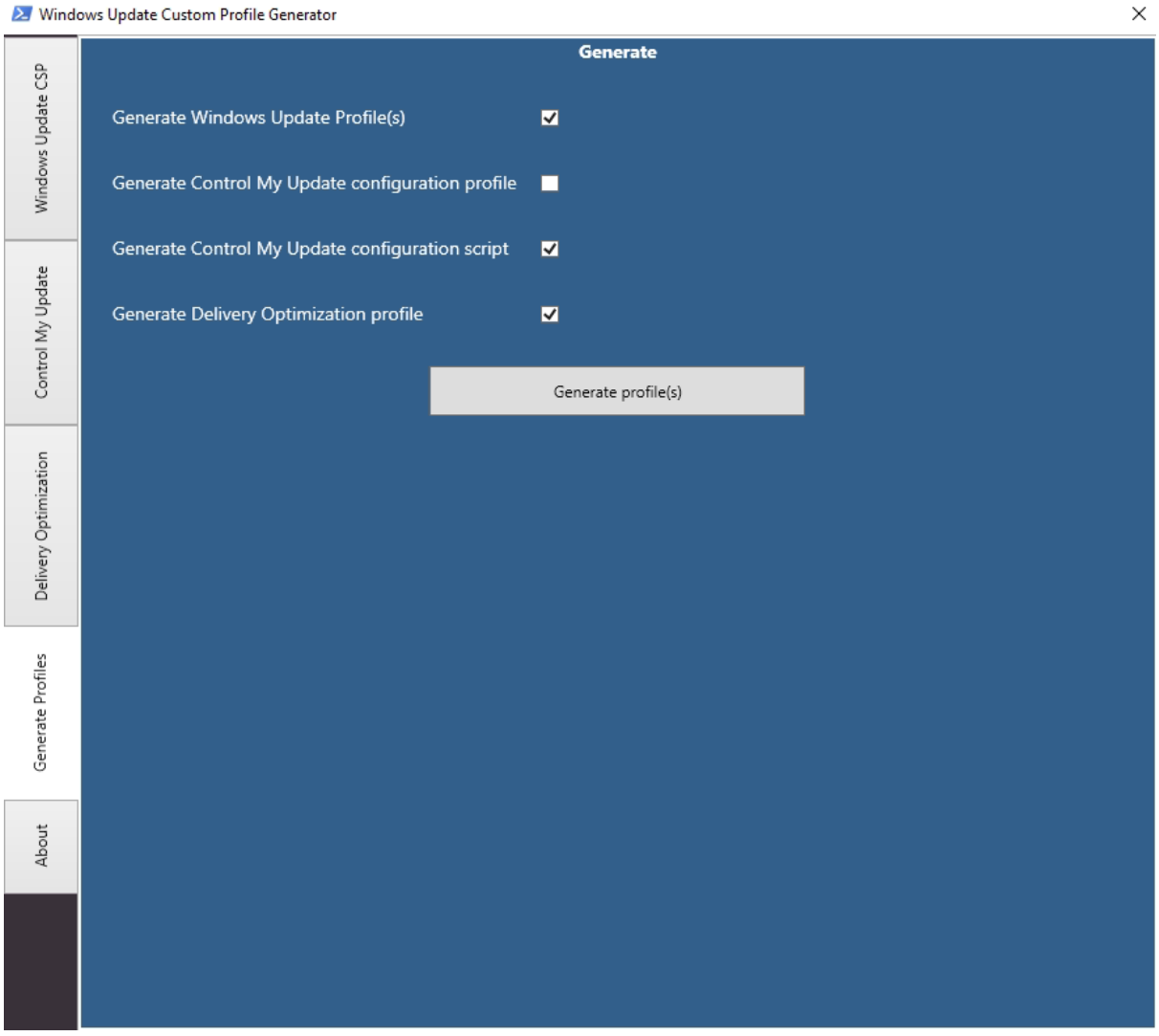
The whole Control My Update (CMU) is based on PowerShell. It relies on Windows Registry for saving and getting the configuration and also for the reporting.
To start with the solution, you need to go through the configuration and installation process which basically are the following steps:

1. Create a configuration script
2. Upload and deploy the application
3. Upload and deploy the sensors to collect the reporting data

For a better understanding here are the detailed steps and information for every step.

3.1. Create a configuration script

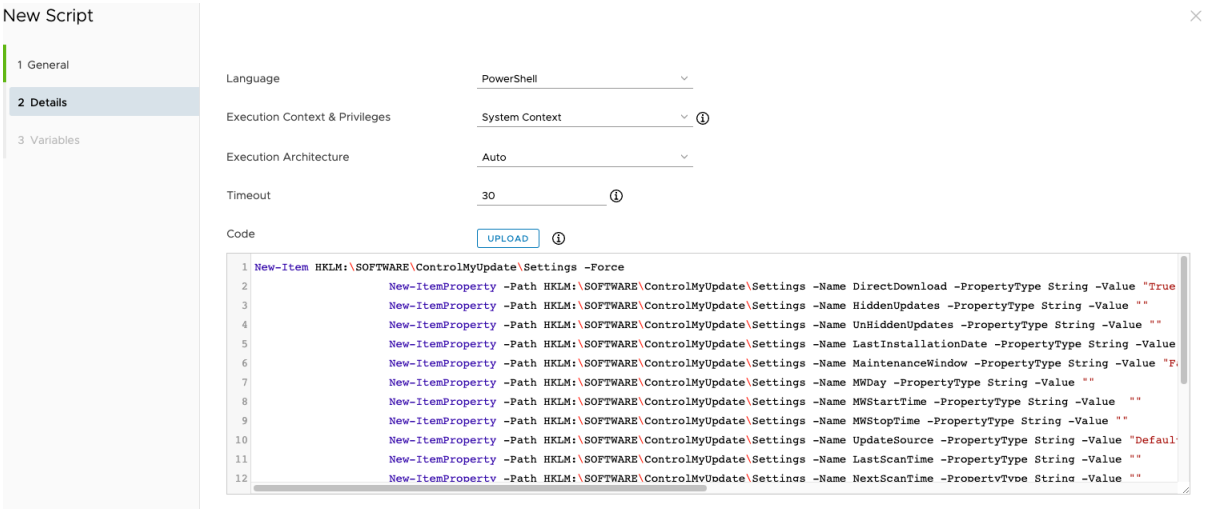
Open the Custom Profile Generator and configure your settings as needed. (for further details, please take a look at point no 2.4).
Now, generate the configuration script.
Navigate the “Generate” and select the “Generate Control My Update configuration script”.



The configuration profile option is a legacy option but also works and is supported.
You'll get a .ps1 file that will look like this:

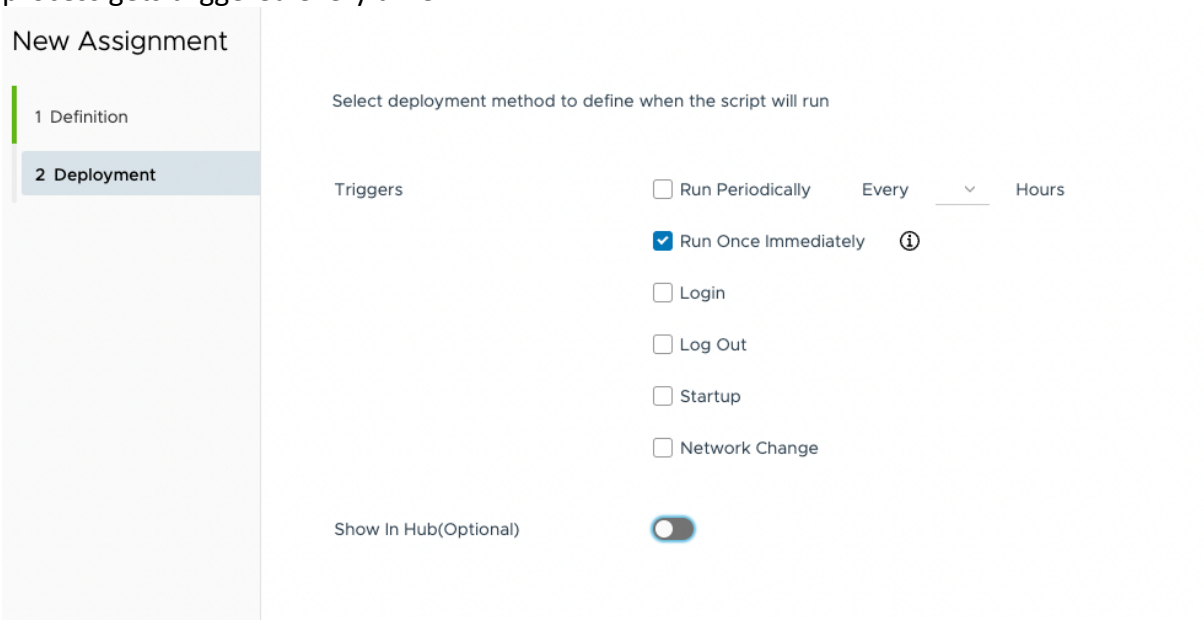
```
New-Item HKLM:\SOFTWARE\ControlMyUpdate\Settings -Force
New-ItemProperty -Path HKLM:\SOFTWARE\ControlMyUpdate\Settings -Name DirectDownload -PropertyType String -Value True
New-ItemProperty -Path HKLM:\SOFTWARE\ControlMyUpdate\Settings -Name EmergencyKB -PropertyType String -Value ""
New-ItemProperty -Path HKLM:\SOFTWARE\ControlMyUpdate\Settings -Name HiddenUpdates -PropertyType String -Value ""
New-ItemProperty -Path HKLM:\SOFTWARE\ControlMyUpdate\Settings -Name UnHiddenUpdates -PropertyType String -Value  ""
New-ItemProperty -Path HKLM:\SOFTWARE\ControlMyUpdate\Settings -Name LastInstallationDate -PropertyType String -Value ""
New-ItemProperty -Path HKLM:\SOFTWARE\ControlMyUpdate\Settings -Name MaintenanceWindow -PropertyType String -Value False
New-ItemProperty -Path HKLM:\SOFTWARE\ControlMyUpdate\Settings -Name MWDay -PropertyType String -Value ""
New-ItemProperty -Path HKLM:\SOFTWARE\ControlMyUpdate\Settings -Name MWStartTime -PropertyType String -Value  ""
New-ItemProperty -Path HKLM:\SOFTWARE\ControlMyUpdate\Settings -Name MWStopTime -PropertyType String -Value ""
New-ItemProperty -Path HKLM:\SOFTWARE\ControlMyUpdate\Settings -Name UpdateSource -PropertyType String -Value Default
New-ItemProperty -Path HKLM:\SOFTWARE\ControlMyUpdate\Settings -Name LastScanTime -PropertyType String -Value ""
New-ItemProperty -Path HKLM:\SOFTWARE\ControlMyUpdate\Settings -Name NextScanTime -PropertyType String -Value ""
New-ItemProperty -Path HKLM:\SOFTWARE\ControlMyUpdate\Settings -Name ScanInterval -PropertyType String -Value 1
New-ItemProperty -Path HKLM:\SOFTWARE\ControlMyUpdate\Settings -Name ScanRandomization -PropertyType String -Value 0
New-ItemProperty -Path HKLM:\SOFTWARE\ControlMyUpdate\Settings -Name ReportOnly -PropertyType String -Value False
New-ItemProperty -Path HKLM:\SOFTWARE\ControlMyUpdate\Settings -Name NotifyUser -PropertyType String -Value True
New-ItemProperty -Path HKLM:\SOFTWARE\ControlMyUpdate\Settings -Name ToastTitle -PropertyType String -Value "Windows Update"
New-ItemProperty -Path HKLM:\SOFTWARE\ControlMyUpdate\Settings -Name ToastText -PropertyType String -Value "A reboot is required to finish updating, please save your work and reboot at your convenience. If no action is taken reboot will happen automatically"
New-ItemProperty -Path HKLM:\SOFTWARE\ControlMyUpdate\Settings -Name ForceRebootNoMW -PropertyType String -Value 1
New-ItemProperty -Path HKLM:\SOFTWARE\ControlMyUpdate\Settings -Name AutomaticReboot -PropertyType String -Value ""
New-ItemProperty -Path HKLM:\SOFTWARE\ControlMyUpdate\Settings -Name RetryCount -PropertyType String -Value 0
New-ItemProperty -Path HKLM:\SOFTWARE\ControlMyUpdate -Name ScriptLogLevel -PropertyType String -Value Info
```

This script can be uploaded to Workspace ONE. Navigate to “Resources” -> “Scripts” and click on “Add” and “Windows”. Type in “Control My Update” and click “Next”.
Upload or copy+paste the configuration and select 64-Bit as Execution Architecure and change the timeout to something like 2 minutes.

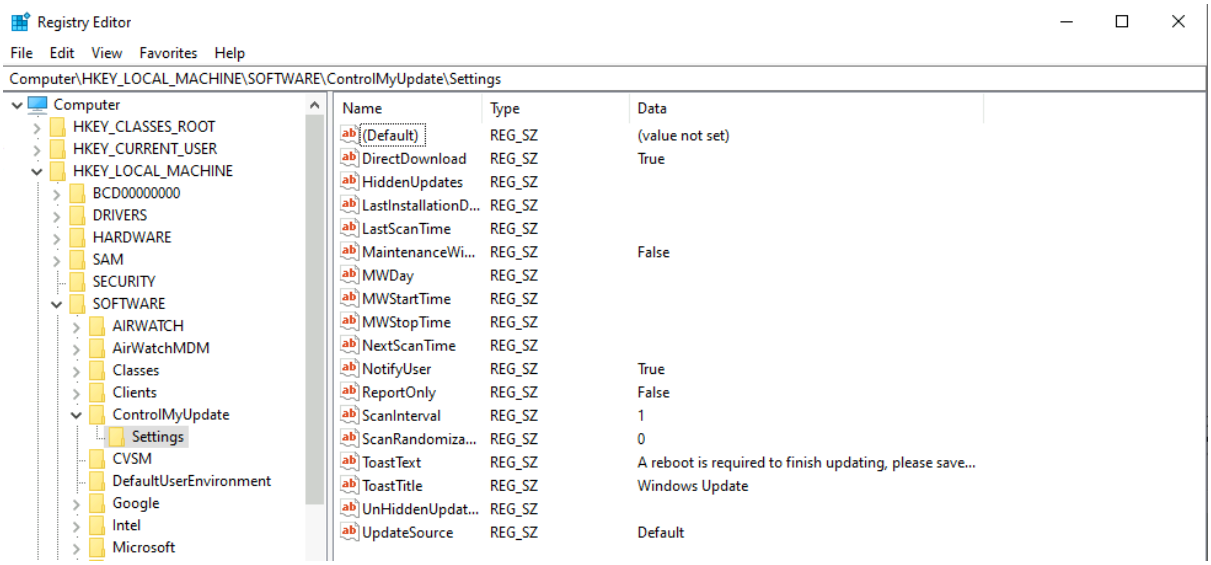


Right now, we are not using Workspace ONE Script variables, so click next and assign the script to your devices.

In the assignment, select “Run Once Immediately”. Feel free to let the settings getting reapplied, but in case you are deploying an emergency patch – please select only the “Run Once Immediately” option and disable everything else. Otherwise, the emergency update process gets triggered every time.



On the device, settings are found in the registry below
HKEY_LOCAL_MACHINE\SOFTWARE\ControlMyUpdate



Once the devices have received the configuration, the next step is to deploy the application.

3.2. Create and deploy the application

In the “Application” folder there is a ready to use .zip file, that you can use for uploading to Workspace ONE.

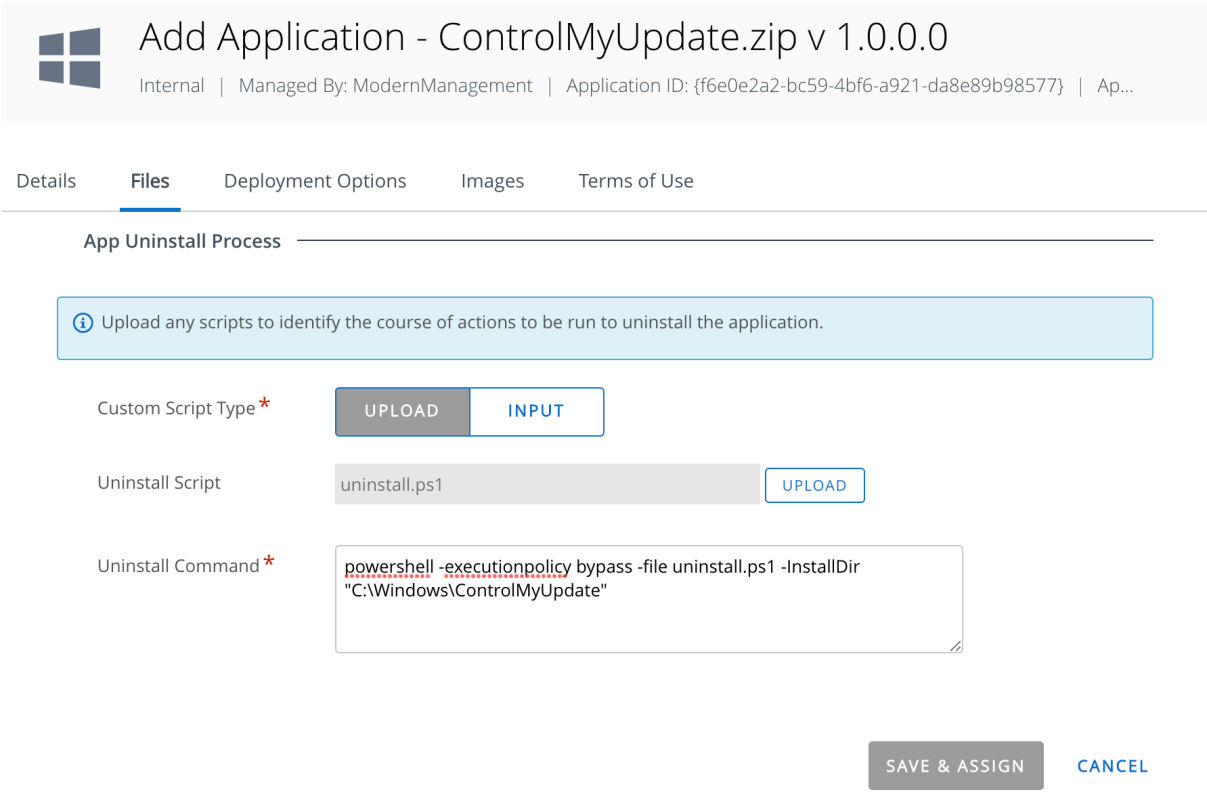
You find a file “installation.txt” in the Application folder. You can copy paste the commands out of this file.

3.2.1. Upload the application

Open the Workspace ONE console and navigate to “Resources” -> “Apps” -> “Native” and add a new application file. Select the ControlMyUpdate.zip file and upload it. Next, change the name to “Control My Update” and the version to 2.3. This will help you to identify updates on the script.

3.2.2. Uninstallation configuration

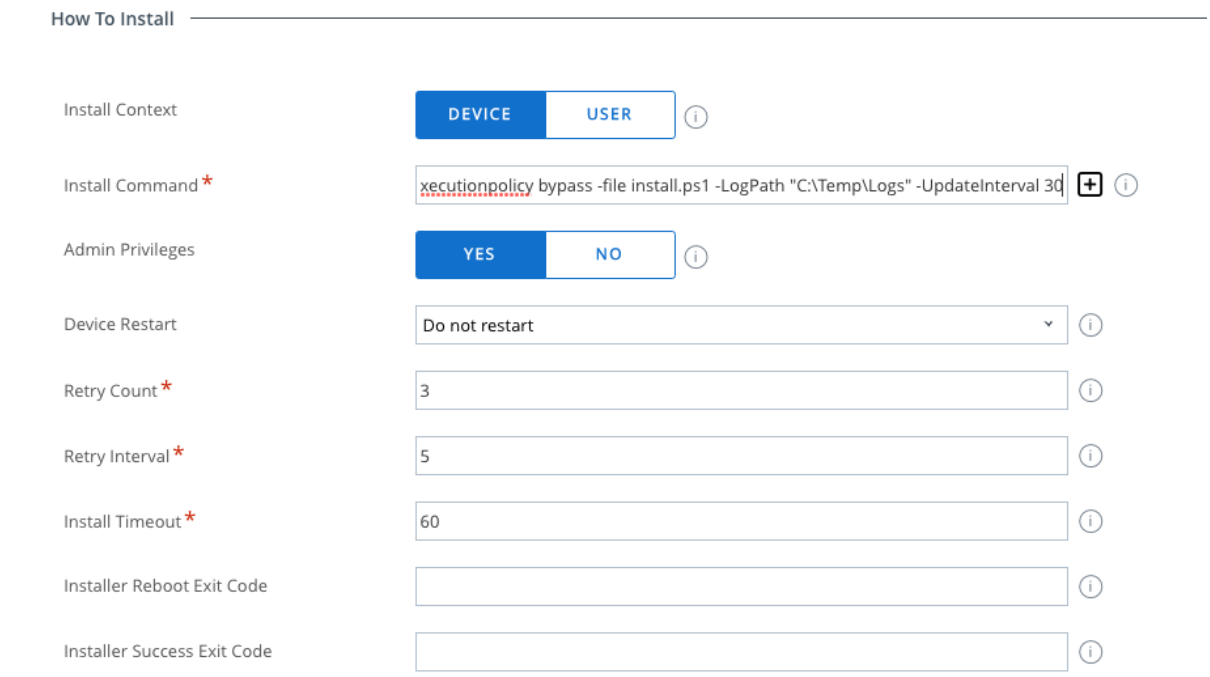
Navigate to the “Files” tab and scroll down to the “App Uninstall Process”. Select upload and upload the uninstall.ps1 file which is also located in the “Application” folder.



As “Uninstall Command” type:
powershell -executionpolicy bypass -file uninstall.ps1 -InstallDir "C:\Windows\ControlMyUpdate"

3.2.3. Install command configuration

Open the “Deployment Options” tab and scroll down to the “Install Command” and type in:



powershell -executionpolicy bypass -file install.ps1 -LogPath "C:\Temp\Logs" -UpdateInterval 30

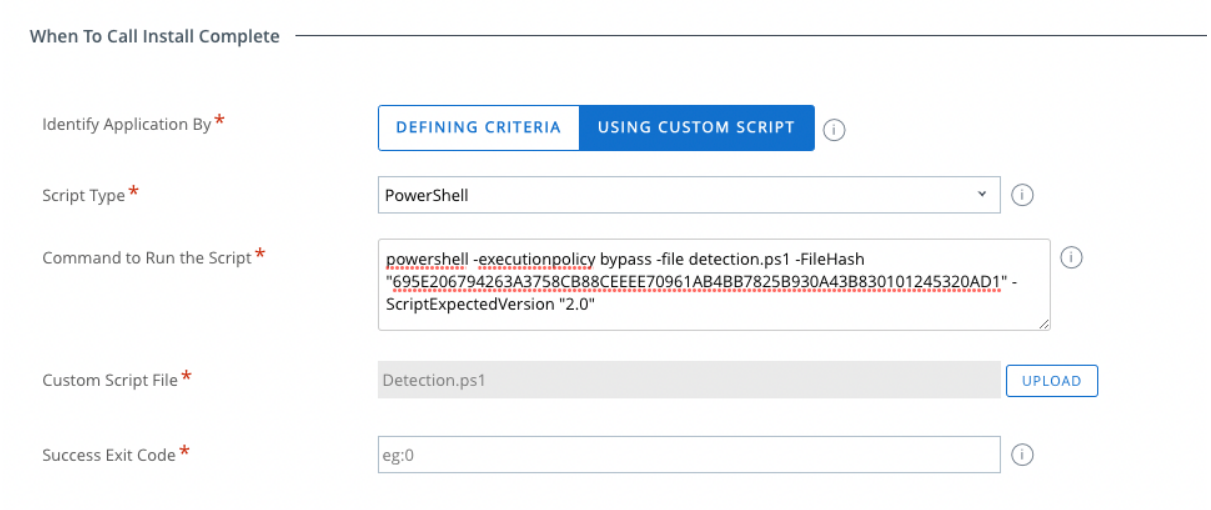
- LogPath
  Define a logpath – if you don't want to use logging, you can either delete the parameter or change it to ""
- UpdateInterval
  Define an interval how frequently the scheduled task should run. This doesn't mean that the device will search for updates every 30 minutes – this is configured via the Scan interval setting. But in case of maintenance windows or emergency updates, the setting makes sure that updates getting installed. Best practice value is 30 minutes.

Change the "Install Timeout" value to something like 5 minutes – just to make sure that we don't need to wait an hour if the installation fails.

Scroll further down to the section "When To Call Install Complete". Select "Using custom script" and select "PowerShell". Upload the "detection.ps1" file as "Custom Script File". In the "Command to Run the Script" textbox write the following:

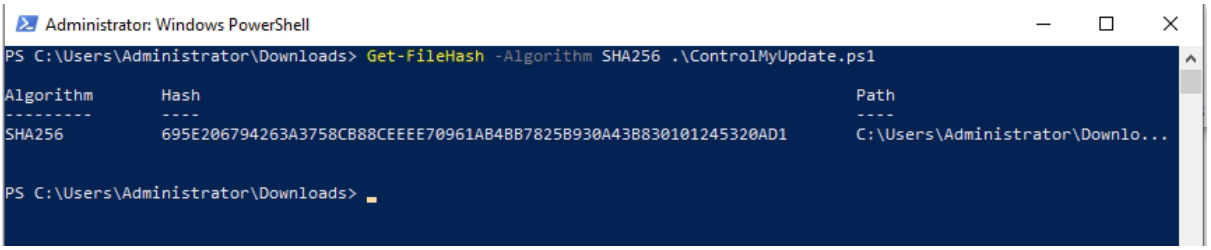
powershell -executionpolicy bypass -file detection.ps1 -FileHash "695E206794263A3758CB88CEEEE70961AB4BB7825B930A43B830101245320AD1" -ScriptExpectedVersion "2.1"

As "Success Exit Code" type "0".



To get the file hash, open PowerShell and navigate to the ControlMyUpdate.ps1.
Run the following command:

Get-FileHash -Algorithm SHA256 .\ControlMyUpdate.ps1



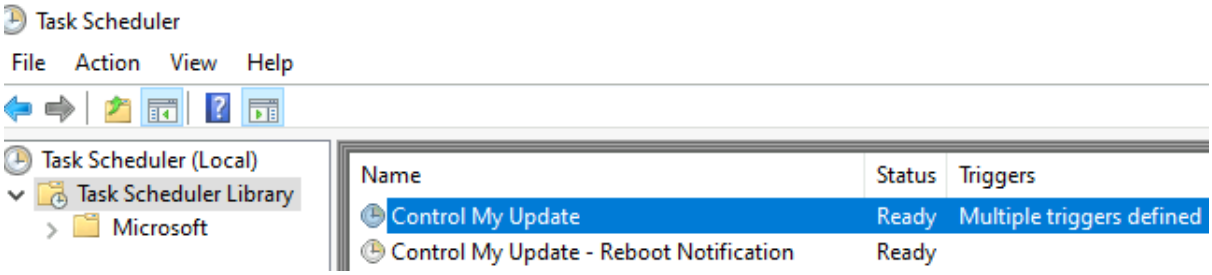
In case you make any changes to the file, you need to update the file hash. This is needed to secure the script, since its running in local system context.

3.2.4. Deployment and installation verification

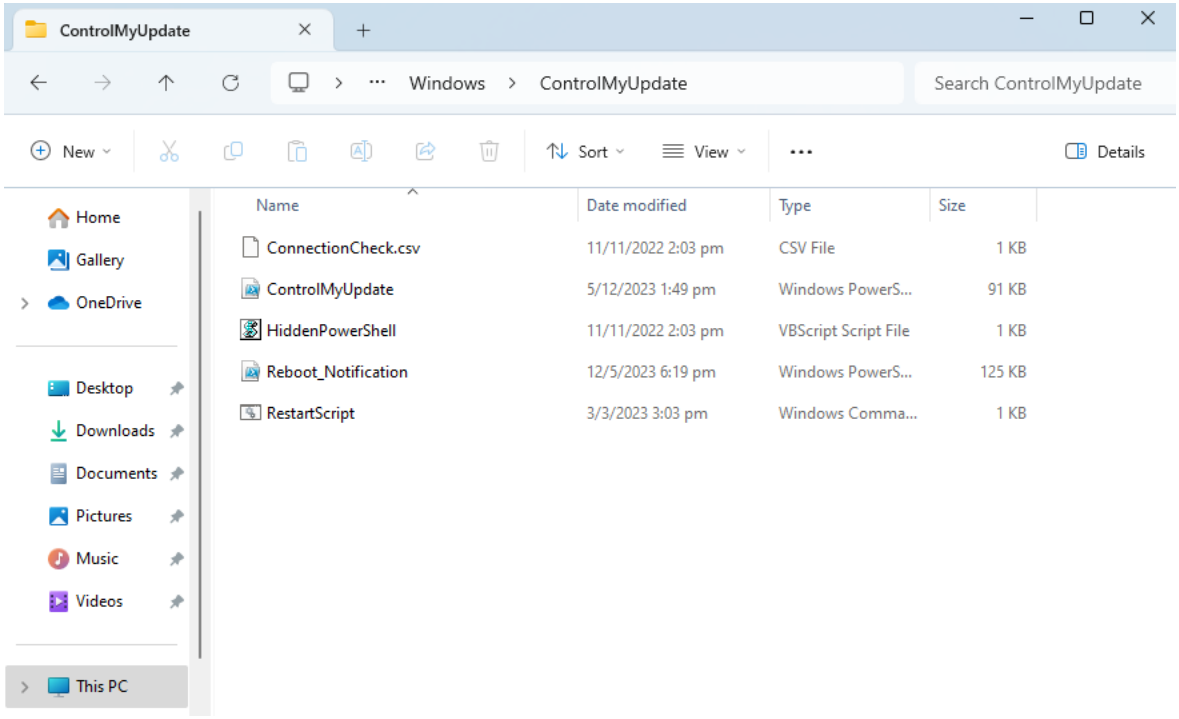
After you created the application, you now need to deploy the application to the devices. We highly recommend deploying the application only to a few devices to see how the devices behave. Make sure you targeted the same devices as in the configuration script deployment.
To verify the installation, you can check the following two things:

1. Open the Task Scheduler on the device and navigate to “Task Scheduler Library”.

   Verify that the “Control My Update” and “Control My Update - Reboot Notification “ tasks were created.



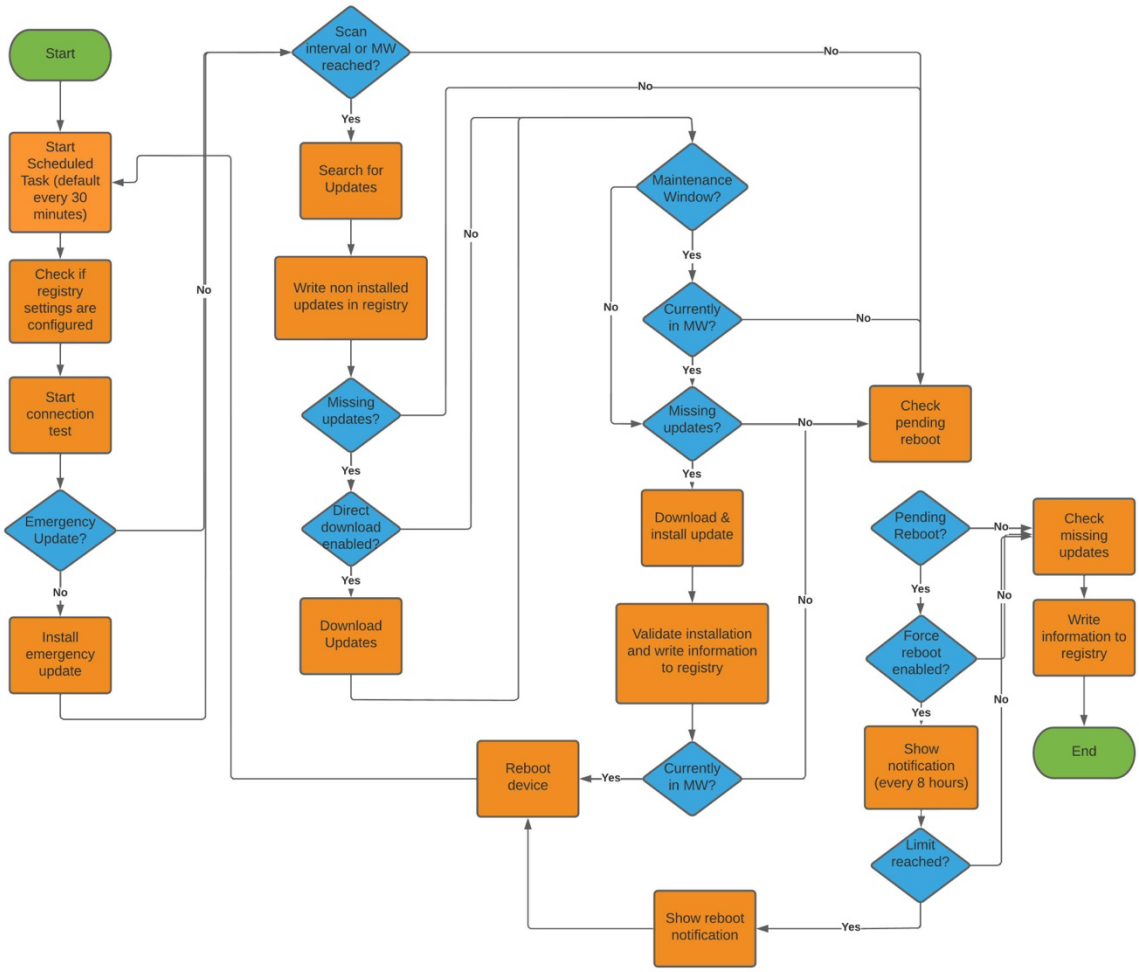
2. Check the filesystem and navigate to C:\Windows\ControlMyUpdate.



Verify that the folder and the listed files exists.
If both things are validated, the solution is ready to work and will start the first run 30 minutes (default if not changed) after the installation.

3.3. Control My Update Workflow

The flow chart will show you the logical workflow of the solution.

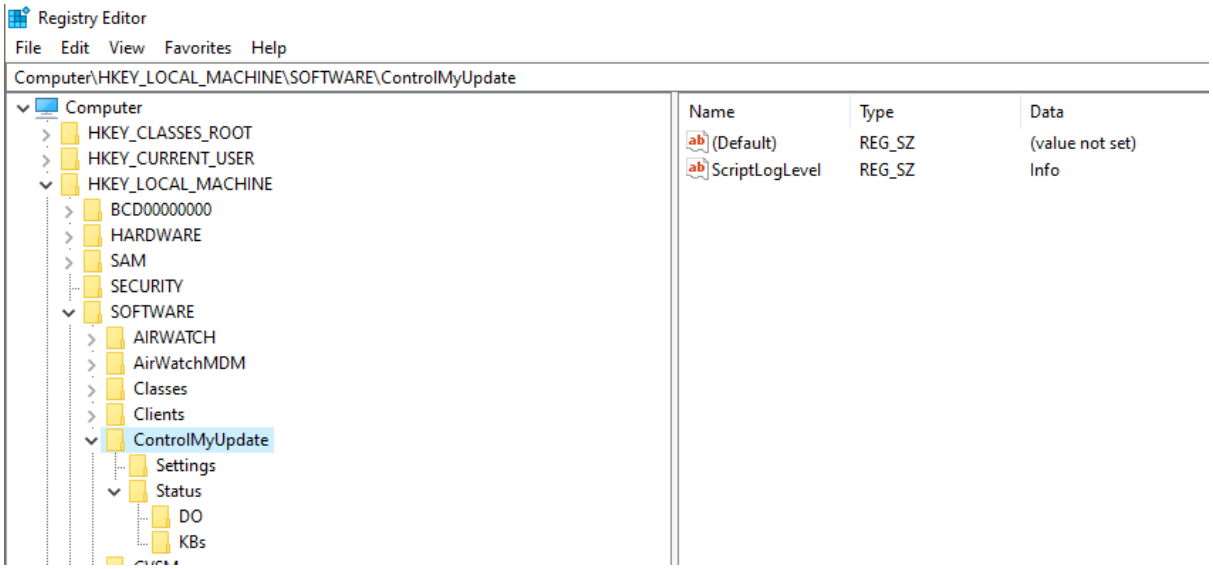


There are some important things that you need to know:

1. The solution will be triggered by a Scheduled Task. The Scheduled Task will start at every reboot and in the interval you selected in the installation command.
2. The solution will not search, download or install updates if the configured scan interval is not reached. The default scan interval is one hour + 0 minutes randomization. In case you change this, the script will start but will not do anything besides writing the current status to the registry.
3. Emergency updates will ignore the scan interval AND the maintenance window. If the scan interval or the maintenance window is reached, then the script will go on. Otherwise, the script will only write the current status to the registry.

3.4. Logging

We've implemented a full logging functionality. By default, the log level is set to "Info" and only the basic information will be added to the log.



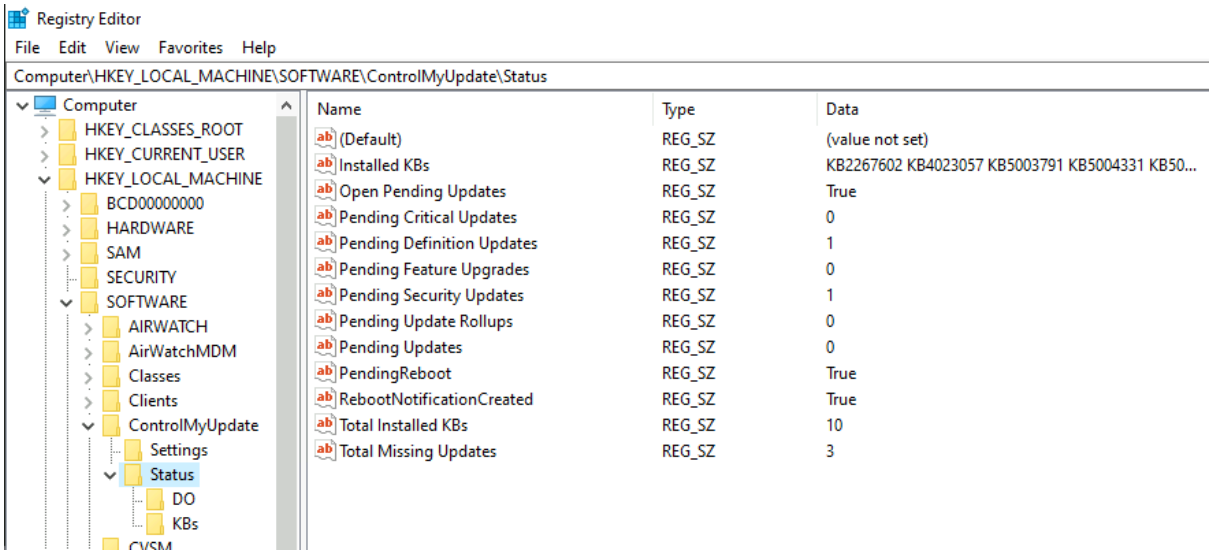
You can configure the log level in the registry root folder HKLM:\Software\ControlMyUpdate

Log level:

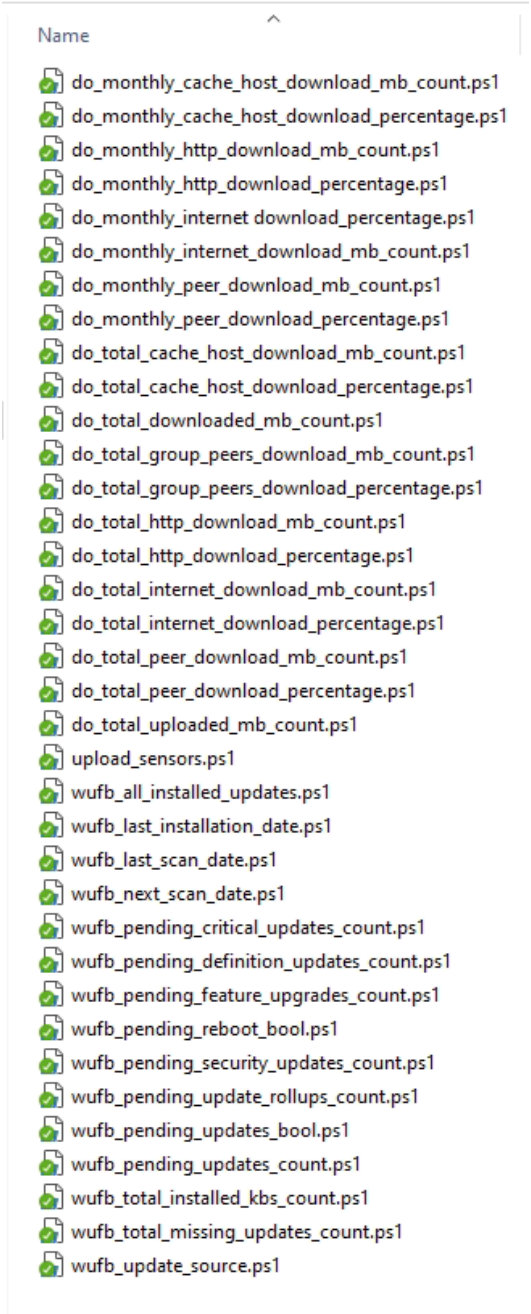
- Info
  General information
- Debug
  More detailed information
- Trace
  Development information to troubleshoot code issues

3.5. Reporting

Sensors are a VMware Workspace ONE function that provide the option to store the gathered information in Workspace ONE Intelligence and create nice Dashboards.
All the collected information is stored in the local registry of the device below the following path:
HKEY_LOCAL_MACHINE\SOFTWARE\ControlMyUpdate\Status
Which looks like this:



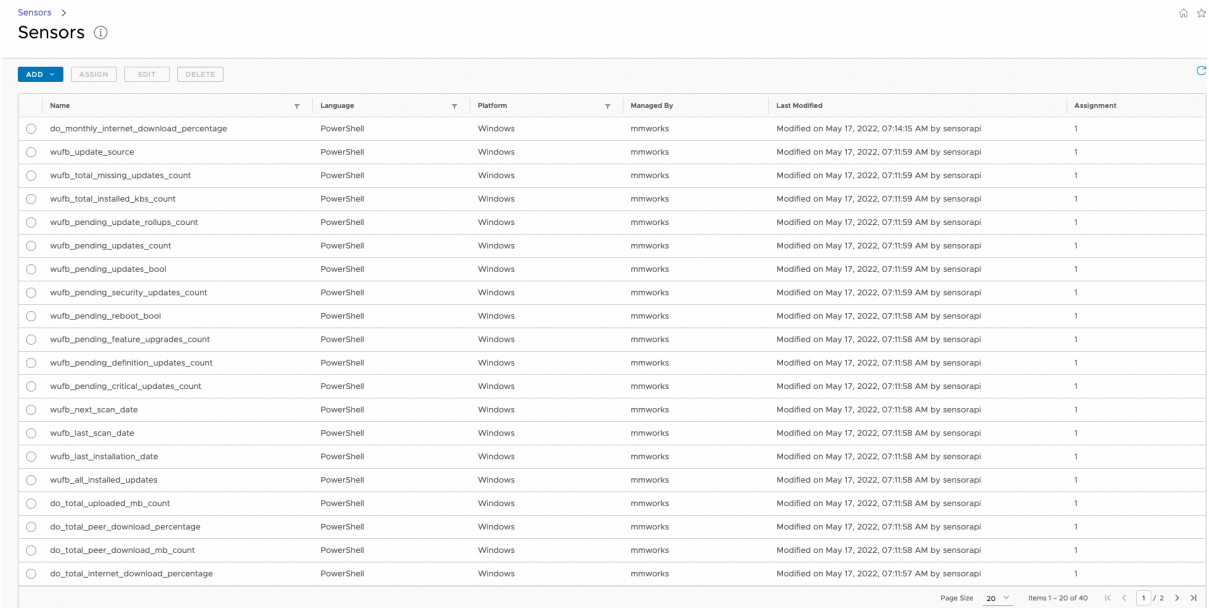
It will show you the current installed updates, the pending update status and current reboot status.
To collect that information, we provided you a bunch of preconfigured sensors:



In the list, there is a file named “upload_sensors.ps1”. If you don’t want to upload the sensors manually, you can use this script to upload all files directly to the console.
Make sure that you move this file out of the “Sensors” folder before you run it – otherwise it will be uploaded too.
Go to powershell, and navigate to the folder where you have the upload_sensors.ps1 file. Then run the following command in PowerShell. The source path points to the location of the Sensors folder. The sensors will all be assigned to the SmartGroupName specified.

. ./upload_sensors.ps1 -SourcePath "C:\Temp" -APIEndpoint "as137.awmdm.com" -APIUser "APIAdmin"-APIPassword "Password" -APIKey "123412341234 -OGID “1234” -SmartGroupName “SGName”

After you uploaded the data to your organization group, you should see the sensors in the UEM console:

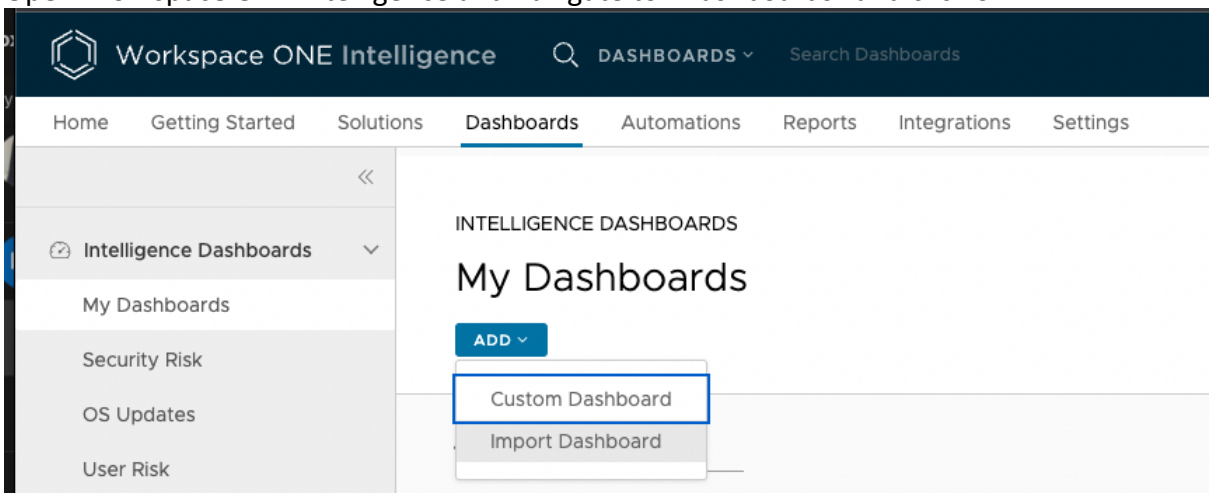


3.6. Monitoring

For monitoring Workspace ONE offers Workspace ONE Intelligence. We can use the data that was collected with the sensors and create a nice Dashboard out of this data.

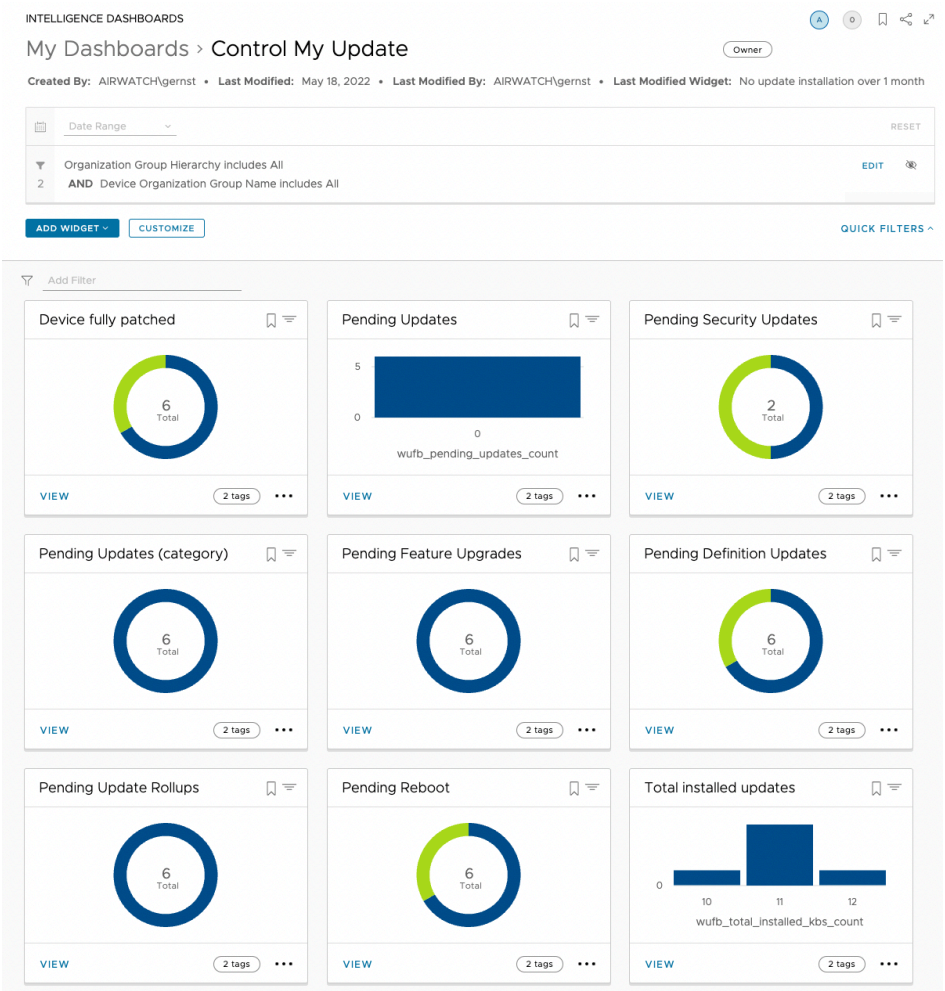
All you need is the “ControlMyUpdate_Dashboard.json“ and access to Workspace ONE Intelligence.

Open Workspace ONE Intelligence and navigate to “Dashboards“ and click on “ADD“.



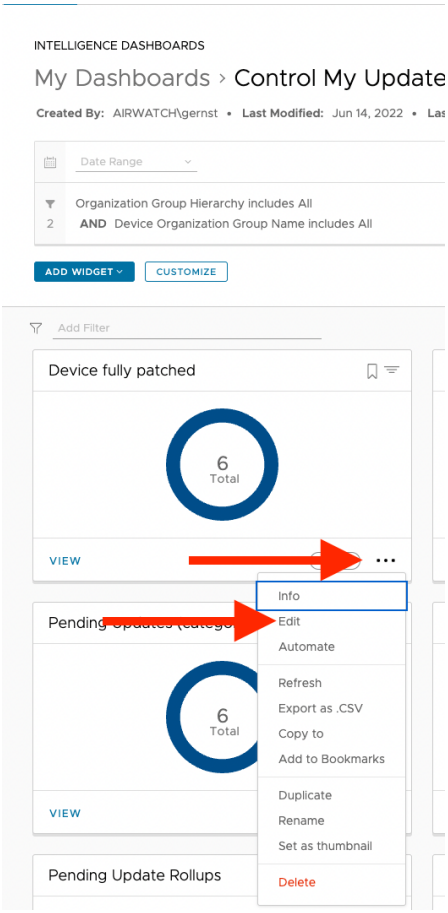
Select “Import Dashboard“ and upload the “ControlMyUpdate_Dashboard.json“.

After the import was done you should see a new Dashboard that looks like this:

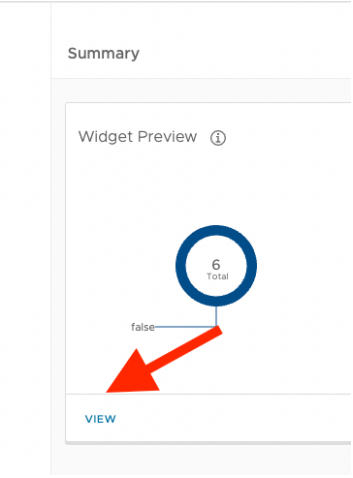


You have the option to change the view and you drill down the charts to get the needed device information.

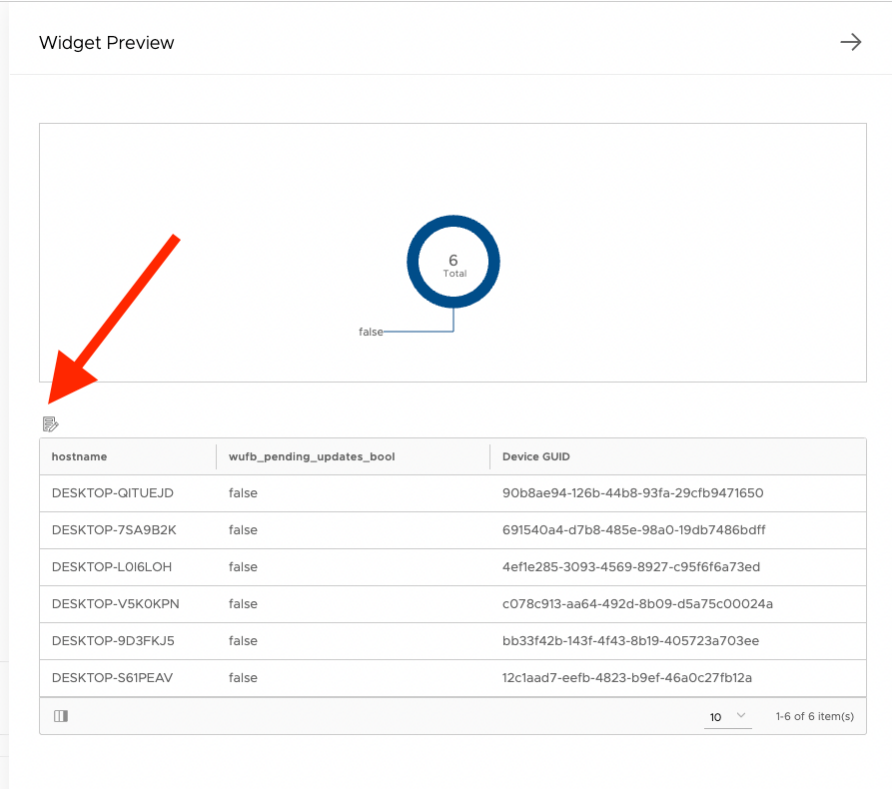
Unfortunately, we can't export the full Dashboard with all the needed columns. Right now, only the Device GUID will show in the details page of the widget.
You would need to add the needed columns in each widget.
Open the dashboard and select one widget with the edit function.



Now select the “view” button on the top right “summary” section.



You can now edit the columns. We recommend adding the hostname and the sensor that is used – in this example the pending reboot sensor named “wufb_pending_updates_bool” (this sensor needs to be changed at every widget).
Of course, you can add additional information.



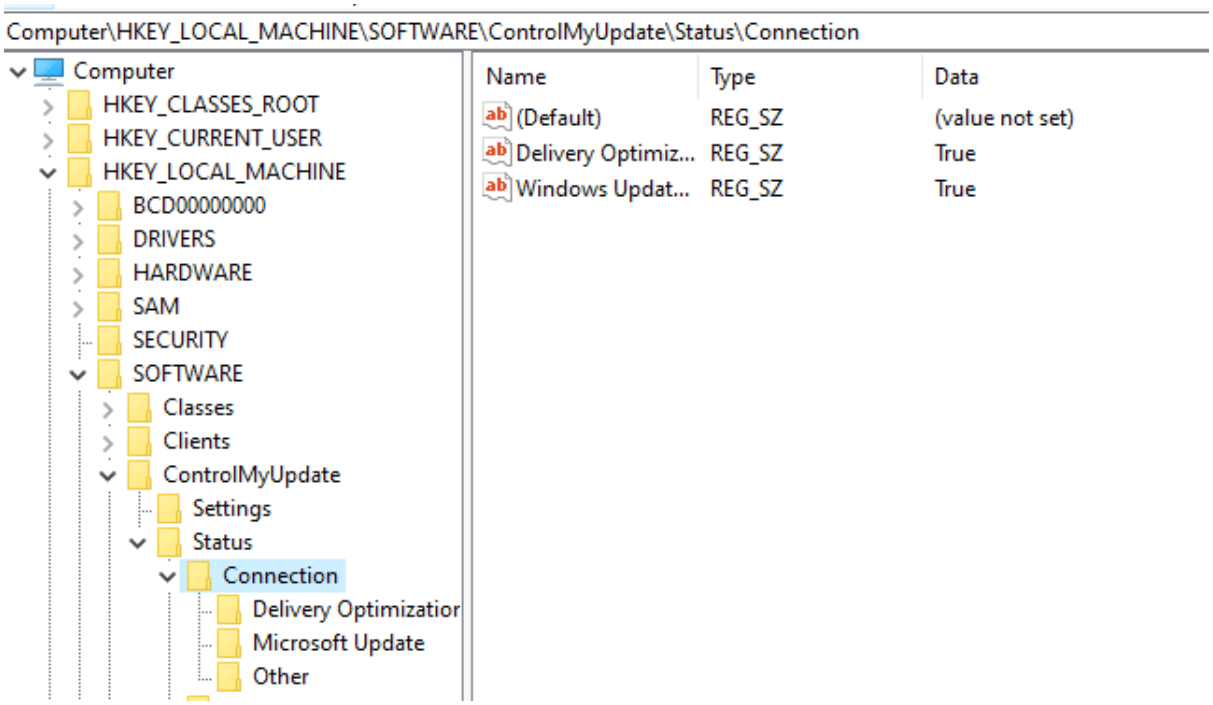
3.7. Connection Test

With version 2.1 we implemented a new connection test functionality. By default, CMU is testing some Microsoft Update and Delivery Optimization URL's. For this we are using a .csv file to provide the information which URL's and which ports are going to be checked.

```
URL,Port,Usecase
geover-prod.do.dsp.mp.microsoft.com,443,DO
geo-prod.do.dsp.mp.microsoft.com,443,DO
geo.prod.do.dsp.mp.microsoft.com,443,DO
geover.prod.do.dsp.mp.microsoft.com,443,DO
dl.delivery.mp.microsoft.com,80,DO
dl.delivery.mp.microsoft.com,443,DO
emdl.ws.microsoft.com,80,DO
update.microsoft.com,80,WU
update.microsoft.com,443,WU
tsfe.trafficshaping.dsp.mp.microsoft.com,443,WU
download.windowsupdate.com,80,WU
download.microsoft.com,80,WU
download.microsoft.com,443,WU
catalog.update.microsoft.com,80,WU
```

As you can see, you can add more URL's to the list.
Add a new line, type in the URL followed by the port and then the use case (WU = Windows Update, DO = Delivery Optimization and Other for all other services).

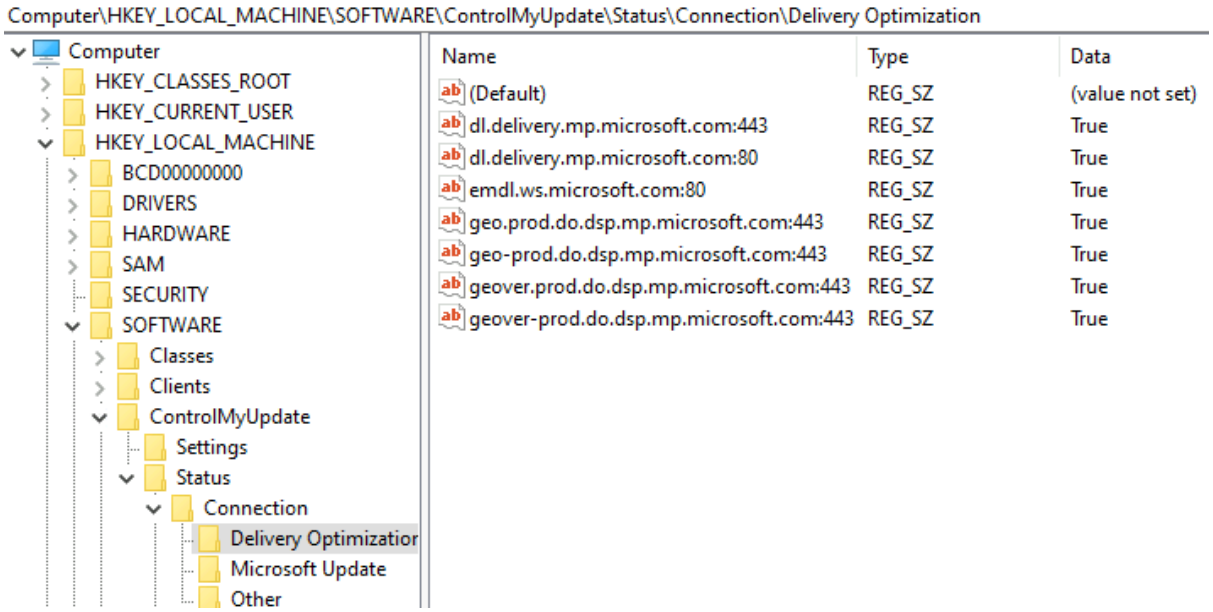
Those information are getting stored in the registry:



Below the Connection hive there is the overall connection status for Delivery Optimization and Windows Update.

True means that all connection tests were successful – false means that there was at least one error.

When you select e.g., Delivery Optimization, you'll see all tested URL's and ports:



4. FAQ

Can we run the Scheduled Task every minute?

Yes, but we highly recommend configuring the Schedule Task not to run less than every 15 minutes. Otherwise, the logfile will be spammed with messages – and even more important,

it doesn't make sense to trigger the process less than 15 min.
Also – if you don't use emergency updates and maintenance windows, best practice would be to configure the scheduled task to run every 60 minutes – or higher.
Best practice value is 30 minutes.

Do I need to set randomization?
No – you already have some kind of randomization because the application gets not installed and executed at the same time on all devices. The randomization is just an additional function to make sure that the devices not reaching out at the same time – this is only recommended for slow internet connections.

Why do I need the scan interval?
You don't want that your devices will check every 30 minutes for updates. Microsoft is not releasing the updates so frequently, so it doesn't make sense to have a scan interval lower than one hour.
Best practice value is 4 hours.

Why is the profile generator not supported for all Windows releases?
Microsoft made a lot of Windows Update changes since the release of Windows 10 and especially in the 20H2 release. Right now, the GUI is only support for 20H2 and higher. There might be something up in later releases to select the Windows release before the configuration but right now it's only for 20H2 and later.

Will VMware provide support for this?
NO! This is a fling that was developed in our free time. VMware is not responsible for this solution!

Is this solution secure?
The PowerShell script will run in system context. To secure the script integrity we implemented the hash value check, and the script is stored in the Windows folder.
This means, the "attacker" needs administrative rights to change the script. Even if the script was changed, the hash value is different and Workspace ONE will reinstall the original script again.