

Windows 10 Toast Notification Script documentation

Contents

Configuration.....	3
Toast - Feature.....	3
UpgradeOS - Feature	3
PendingRebootUptime - Feature.....	3
PendingRebootCheck - Feature	3
ADPasswordExpiration - Feature.....	4
TargetOS - Option	4
MaxUptimeDays - Option	4
PendingRebootUptimeText - Option	4
PendingRebootCheckText - Option	4
ADPasswordExpirationText - Option	4
ADPasswordExpirationDays - Option	4
RunPackageID - Option.....	5
RunApplicationID - Option.....	5
RunUpdateID - Option	5
Deadline - Option	5
DynamicDeadline - Option	5
CreateScriptsAndProtocols - Option.....	5
UseSoftwareCenterApp - Option.....	6
UsePowershellApp - Option	6
CustomAudio - Option	6
LogoImageName - Option	6
HeroImageName - Option	6
ActionButton - Option	6
DismissButton - Option.....	6
SnoozeButton - Option	6
Scenario - Option	7
Action - Option	7
Installation.....	8
Configuration Manager	8

Group Policy Preferences	9
Create Files	9
Scheduled Task	10
Intune	11
Logging.....	11
Examples.....	Error! Bookmark not defined.
Additional information	12

Configuration

The script is customizable through the config-toast.xml. The config-toast.xml can be located locally next to the New-ToastNotification.ps1 file or located on a network share.

The script will try to use the file locally if no other path is set in the -Config parameter.

New-ToastNotification.ps1 -Config [\\YourPath\ToastNotificationScript\Configs\config-toast-pendingreboot.xml](#)

Setting a remote config file enables you to quickly modify the purpose of the Toast Notification without pushing new files to the computer running the toast. This also enables you to have several configurations, for running multiple purposes. Just point to another configuration when required.

The config-toast.xml can be hosted online as well:

New-ToastNotification.ps1 -Config "https://krpublicfiles.blob.core.windows.net/toastnotification/config-toast-whfb.xml"

Toast - Feature

This enables or disables the entire toast notification.

Possible values: True | False

<Option Name="Toast" Enabled="True" />

UpgradeOS - Feature

This enables the toast for use with Windows Servicing. When this is set to True, the build of the running OS is compared with the target OS.

Possible values: True | False

<Feature Name="UpgradeOS" Enabled="False" />

PendingRebootUptime - Feature

This enables the toast for use for checking computer uptime. If this is set to True, the MaxUptimeDays is considered. If toast is displayed if the MaxUptimeDays is exceeded.

Possible values: True | False

<Feature Name="PendingRebootUptime" Enabled="False" />

PendingRebootCheck - Feature

This enables the toast for use for checking for pending reboots.

Pending reboots are checked in WMI and registry. The toast is displayed if pending reboots are found. WMI is only checked if the computer where the toast is running has the SCCM client installed.

Possible values: True | False

<Feature Name="PendingRebootCheck" Enabled="True" />

ADPasswordExpiration - Feature

This enables the toast for use for checking for expiring Active Directory password.

Active Directory password expiration date is checked with local AD for the logged on user. If the password is about to expire, a toast notification will be displayed, reminding the user to change their password.

Possible values: True | False

```
<Feature Name="ADPasswordExpiration" Enabled="False" />
```

TargetOS - Option

This sets the expected target OS build when running the toast with the UpgradeOS feature. This is to prevent the toast from displaying on computers already upgraded to the target OS. No need to remind users of an upgrade they already completed.

Possible values: 19041 = 2004 | 18363 = 1909 | 18362 = 1903 | 17763 = 1809

```
<Option Name="TargetOS" Build="18351" />
```

MaxUptimeDays - Option

When using the toast for checking for pending reboots, the value of this option is also evaluated. If the uptime of the computer exceeds the value set in this option, the toast will be displayed. This option is only considered when having PendingRebootCheck set to Enabled.

```
<Option Name="MaxUptimeDays" Value="-1" />
```

PendingRebootUptimeText - Option

This option adds an additional group with text about the uptime of the computer.

Possible values: True | False

```
<Option Name="PendingRebootUptimeText" Enabled="False"
```

PendingRebootCheckText - Option

This option adds an additional group with text about the pending reboot in registry/WMI.

Possible values: True | False

```
<Option Name="PendingRebootCheckText" Enabled="False"
```

ADPasswordExpirationText - Option

This option adds an additional group with text about the expiring password.

Possible values: True | False

```
<Option Name="ADPasswordExpirationText" Enabled="False" />
```

ADPasswordExpirationDays - Option

This option defines how many days in advance the toast notification will start appearing to remind users change their password.

```
<Option Name="ADPasswordExpirationDays" Value="90" />
```

RunPackageID - Option

Run any Configuration Manager packageID directly from the action button. Configure the value to the packageID of the package or task sequence in Configuration Manager.

```
<Option Name="RunPackageID" Enabled="False" Value="KR100907" />
```

RunApplicationID - Option

Run any Configuration Manager applicationID directly from the action button. Configure the value to the application ID of the application in Configuration Manager.

```
<Option Name="RunApplicationID" Enabled="True" Value="ScopeId_A9117680-D054-482B-BC97-532E6CBD0E6B/Application_fd55f35c-4e34-4490-a3ec-ee0b79233ec6" />
```

RunUpdateID - Option

Run any Configuration Manager updateID directly from the action button. Configure the value to the KB-article number of the update in Configuration Manager.

```
<Option Name="RunUpdateID" Enabled="False" Value="4561600" />
```

RunUpdateTitle - Option

This option is used in conjunction with RunUpdateID to further specify which update that is being run from the action button. This is especially needed when several updates with the same KB-article is deployed to the device. If only one update with the specified KB-article is deployed to the client, using this option is not required.

```
<Option Name="RunUpdateTitle" Enabled="False" Value="" />
```

Deadline - Option

Adds an additional group with text containing information about a potential deadline.

```
<Option Name="Deadline" Enabled="False" Value="5 June 2019" />
```

DynamicDeadline - Option

Adds an additional group with text and date containing the deadline for the deployed task sequence or feature update. Configure the value to the package id of the task sequence or KB article of the feature update.

Possible values: True | False

```
<Option Name="DynamicDeadline" Enabled="True" Value="KR100907" />
```

CreateScriptsAndProtocols - Option

This option creates custom scripts and protocols for use with installing packages, task sequences, applications, and software updates directly via the action button. Script and protocol for initiating a reboot from the action button is created as well.

Scripts are created in ProgramData\ToastNotificationScript

Custom protocols are created in HKCU\Software\Classes

Possible values: True | False

```
<Option Name="CreateScriptsAndProtocols" Enabled="True" />
```

UseSoftwareCenterApp - Option

This option set the app doing the toast notification to Software Center. This should only be used, if the computer has the ConfigMgr client installed.

```
<Option Name="UseSoftwareCenterApp" Enabled="True" />
```

UsePowershellApp - Option

This option set the app doing the toast notification to Powershell. Use this option, if the computer running the toast doesn't have the ConfigMgr client installed.

```
<Option Name="UsePowershellApp" Enabled="False" />
```

CustomAudio - Option

This option adds custom audio to the toast notification. The text set in the config.xml will be read aloud following the actual toast notification.

```
<Option Name="CustomAudio" Enabled="False" />
```

LogoImageName - Option

This option allows you to change the image used as logo in the toast notification. By default, the script looks for the image in the image folder. The image can be hosted online as well in blob storage or similar using the http or https protocol.

```
<Option Name="LogoImageName"
```

```
Value="https://krpublicfiles.blob.core.windows.net/toastnotification/ToastLogoImage.jpg" />
```

```
<Option Name="LogoImageName" Value="ToastLogoImageWindows.jpg" />
```

HeroImageName - Option

This option allows you to change the image used as logo in the toast notification. By default, the script looks for the image in the image folder. The image can be hosted online as well in blob storage or similar using the http or https protocol.

```
<Option Name="HeroImageName"
```

```
Value="https://krpublicfiles.blob.core.windows.net/toastnotification/ToastHeroImage.jpg" />
```

```
<Option Name="HeroImageName" Value="ToastHeroImageWindows.jpg" />
```

ActionButton - Option

Enables or disables the action button. Also sets the name displayed on the button.

```
<Option Name="ActionButton" Enabled="True" Value="Understood" />
```

DismissButton - Option

Enables or disables the dismiss button. Also sets the name displayed on the button.

```
<Option Name="DismissButton" Enabled="True" Value="Not now" />
```

SnoozeButton - Option

Enables or disables the snooze button. Also sets the name displayed on the button. Enabling this button always displays all 3 buttons.

```
<Option Name="SnoozeButton" Enabled="True" Value="Snooze" />
```

Scenario - Option

This option controls how long the toast notification is displayed. Reminder is displayed until the user dismisses the toast. Short is about 5 seconds. Long is about 25 seconds.

Possible values are: reminder | short | long | alert

```
<Option Name="Scenario" Type="reminder" />
```

Action - Option

This option sets the action for the action button. There are limited possibilities here. One option being is launching Software Center going directly to an available application.

```
<Option Name="Action" Value="softwarecenter:SoftwareID=ScopeId_A9117680-D054-482B-BC97-532E6CBD0E6B/Application_4920c244-5b1e-4434-9117-fded57f5c80f" />
```

The toast notification action button is limited to protocol based actions. The script is therefore able to create such custom protocols automatically. See CreateScriptsAndProtocols - Option. You can create those yourself as well, enabling the toast notification to almost launch anything through the action button, if the necessary scripts and protocols are in place.

```
<Option Name="Action" Value="ToastRunApplicationID:" />
```

```
<Option Name="Action" Value="ToastRunPackageID:" />
```

```
<Option Name="Action" Value="ToastRunUpdateID:" />
```

```
<Option Name="Action" Value="ToastReboot:" />
```

Installation

Configuration Manager

The easiest approach is to run this directly from ConfigMgr using a package. In this scenario, there is no need to copy down any files locally to the computer.

- Simply create a package containing all the files as source files. You can leave out the config.xml and point this to an UNC path to avoid updating DPs every time you need to change the config.xml
- Create a program running powershell and the New-ToastNotification.ps1

Run Toast Notification Properties

OpsMgr Maintenance Mode

General Requirements Environment Advanced Windows Installer

Name: Run Toast Notification

Your use of software deployed by ConfigMgr may be subject to license terms. You should review any applicable license terms prior to deploying software.

User description:

Command line: powershell.exe -ExecutionPolicy Bypass -NoLogo Browse...

Start in:

Run: Hidden

After running: No action required

Category:

OK Cancel Apply

Command line example: powershell.exe -ExecutionPolicy Bypass -NoLogo -NonInteractive -NoProfile -WindowStyle Hidden -File .\New-ToastNotification.ps1 -Config [\\YourUNCPath\ToastNotificationScript\Configs\config-toast-pendingreboot.xml](#)

Group Policy Preferences

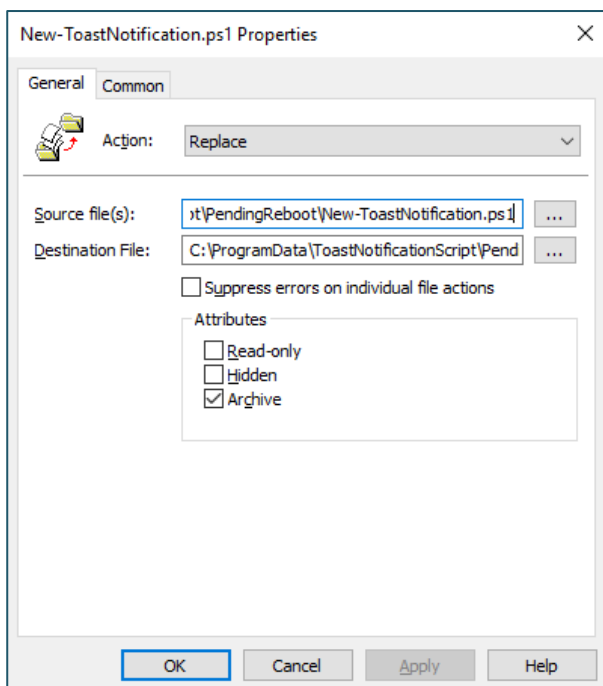
Create Files

You can also use GPP to both create a scheduled task and copy the files down locally. I suggest that the files exist locally when using this approach to avoid being dependent on network connectivity.

In this scenario, you create new files locally using Group Policy Preferences. Point to the source files, which should be a network path and point to the destination path, which should be a local path. This needs to be done for each file you need to have locally.

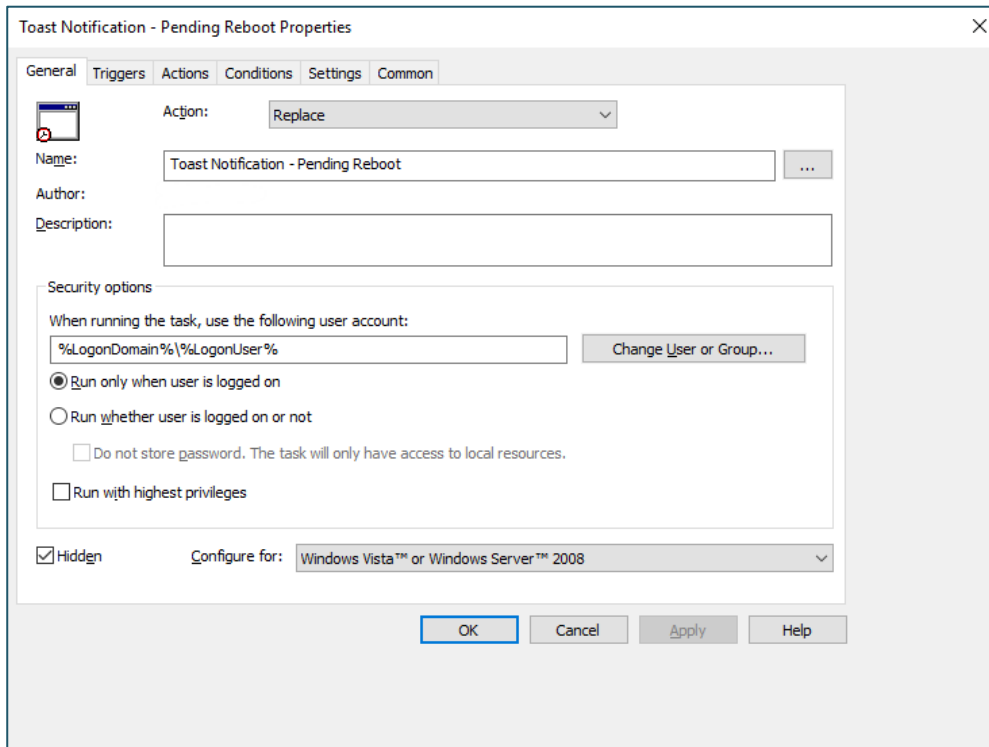
Source: [\\YourFileShare\ToastNotificationScript\PendingReboot\New-ToastNotification.ps1](#)

Destination: C:\ProgramData\ToastNotificationScript\PendingReboot\New-ToastNotification.ps1



Scheduled Task

Create a new scheduled task, running in the logged on user's context.



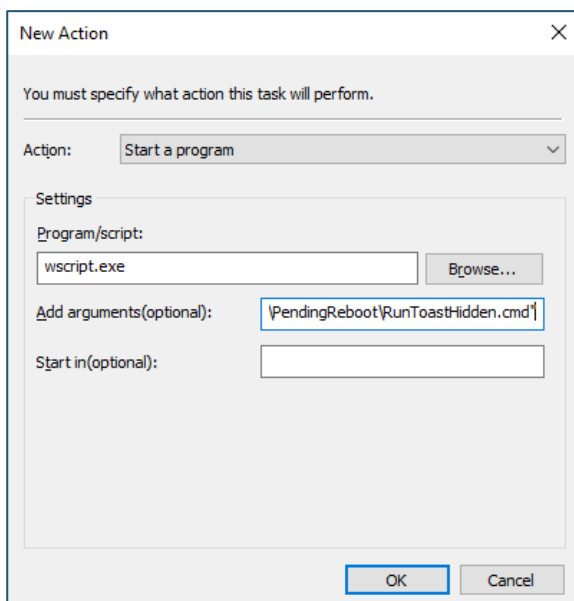
If you want the scheduled task to run completely silent, you will need to make use of 2 additional files which I also included in the download; Hidden.vbs and RunToastHidden.cmd

This is due to the nature of Powershell being a console application and the task running as the logged on user. Simply selecting Hidden and running Powershell.exe hidden is not enough.

Program/script: wscript.exe

Arguments: "C:\ProgramData\ToastNotificationScript\PendingReboot\Hidden.vbs"

"C:\ProgramData\ToastNotificationScript\PendingReboot\RunToastHidden.cmd"



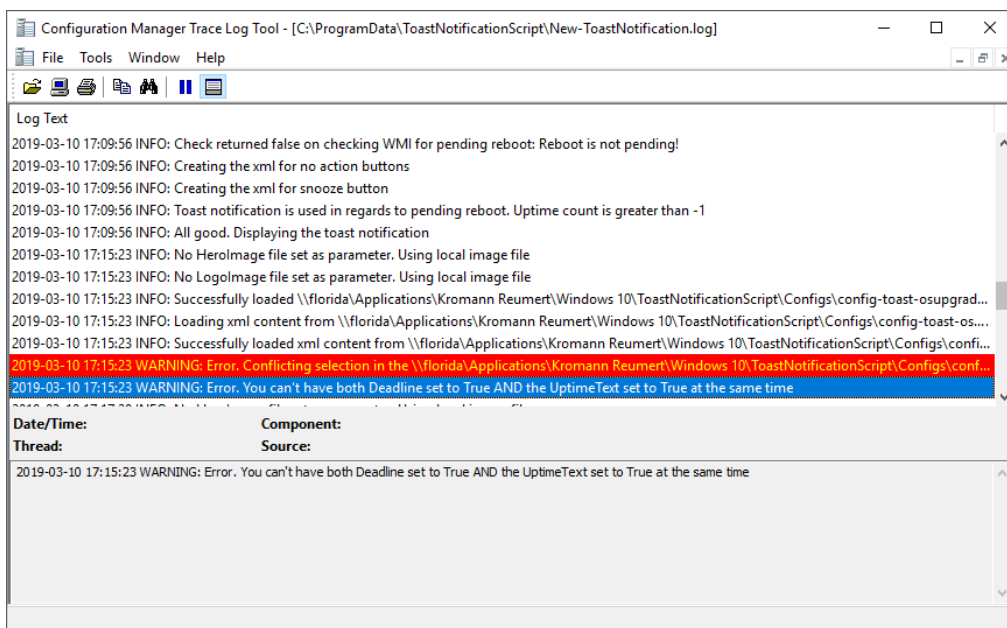
Intune

The Windows 10 Toast Notification Script supports being used with Endpoint Analytics Proactive Remediations. See this blog post: <https://www.imab.dk/windows-10-toast-notification-script-update-support-for-use-with-endpoint-analytics-proactive-remediations/>

Logging

All actions regarding the toast notification are logged to AppData\ToastNotificationScript\New-ToastNotification.log

This is useful when troubleshooting issues or if curious to get a better understanding of what is done.



Additional information

Author: Martin Bengtsson

Blog: www.imab.dk

Twitter: @mwbengtsson

Version: 2.0.0

Date: 11 March 2019

Updated: 22 August 2020