

# Force reload Intune Powershell Scripts

By [Michael Mardahl](#)  May 7, 2019  [Enterprise Mobility](#), [Intune](#), [Modern Management](#), [PowerShell](#)  [Comments](#)

Last Updated on September 15, 2019

3 min read

Force Microsoft Intune Management Extension to reload specific PowerShell scripts, either continuously or at logon.

'' UPDATE: You should also read my article on "Hacking the Intune Management Extension", as it explains different ways to achieve this goal.

## The current limitations of Intune Management Extension

Microsoft Intune (MDM) only supports an initial deployment of a PowerShell script to the end users. Once deployed successfully (or failed 3 times), it will never run again for that user.

This is a problem for many Intune Administrators as they try and create scripts to solve some of the limitations within Intune MDM on Windows 10.

*Like my solution for managing [Local Administrators on AAD Joined Devices with AAD Security Groups](#).*

## Workaround / Solution

The following is me trying to explain my thoughts around this solution. If you just want to have a look at it, you can find it on GitHub. Feel free to fork/pull and generally help out with the project. 😊

## SideCarBehaviourScript (SCBS) repository on GitHub

As you might know, you can manually force the re-run of PowerShell scripts by deleting specific keys in the HLKM registry. However, the scripts are obscure and require GraphAPI – and a lot of fiddling – to make sense of, as nicely demonstrated by Ben Reader on his blog post <http://powers-hell.com/2018/04/16/how-to-force-intune-configuration-scripts-to-re-run/>.

## Rethinking

I found that approach to be a bit troublesome to automate sensibly.

I decided to look at it the other way around.

I looked for a way to make the registry keys easily identifiable, so I could delete them as required.

It turns out that adding a "Write-Output" command to the beginning of my configuration scripts, actually added the output directly into the registry.



This made it very easy for me to identify which scripts I wanted to either "reload continuously" or "execute at log on".

## PowerShell script to the rescue (again!)

Keeping this behaviour in mind, I built three scripts:

### [SideCarBehaviour-AtStartup.ps1](#)

- Removes registry entries for scripts that need to run each time a user logs on.

A different perspective helps... 

- This effectively makes it work for multiple users on the same computer, as the Intune Management Extensions will run at logon, find the policy missing, and re-run it.

### [SideCarBehaviour-Reload.ps1](#)

- Removes registry entries for scripts that need to run at a set interval (90 minutes default).

### [Configure-ScheduledTasks.ps1](#)

- This script "installs" the two scheduled tasks required for each script to run at the right time.
- It must be run with elevated permissions, and will point the tasks to the folder where it's executed from. Be sure to put it someplace practical, like <program files (x86)> \ SideCarBehaviourScript

I deploy it using the [PowerShell App Deployment Toolkit](#), wrapped with the [MSIWrapper](#).

An example [Deploy-Application.ps1](#) is in the GitHub repo.

## What you MUST do – to make this work

After deploying the solution to the Windows 10 device,

you **MUST** add one of the required output lines to all the scripts whose run behaviour you need to change.

**The following bits of PowerShell should be added early in the script thereby making it the first output.**

- To have Intune re-run the script at logon:

```
Write-Output "<scriptfile name>"
```

```
Write-Output "###SideCarBehaviour##ATSTARTUP"
```

- To have Intune re-run the script continuously:

```
Write-Output "<scriptfile name>"
```

```
Write-Output "###SideCarBehaviour##RELOAD"
```

**gain:** These bits of code should be put in YOUR own scripts, *not the ones you download from me.*



## Final Words

Hope you found this solution useful and decide to contribute via GitHub.

NB: This is all work in progress, and is provided AS IS so please test as much as possible!

Please follow me on Twitter [@michael\\_mardahl](#) to learn more about me and the knowledge I share.

## About Author



### **Michael Mardahl**

Michael works as a Cloud Architect with APENTO in Denmark. Specializing in customer journeys from classic Infrastructure to Cloud consumption. He has been in the IT industry for more than 20 years, and has experience from a broad range of IT projects. When not at work, Michael enjoys the value of spending time with family and friends, and BLOG's passionately about Enterprise Mobility whenever he

has time to spare.