

Intune Powershell scripting examples

Petri Paavola
Microsoft MVP - Windows and Intune

Petri Paavola

Microsoft MVP –
Windows and Intune
Senior Modern Management Principal

Petri.Paavola@yodamiitti.fi

Skills

- › Powershell / Graph API
- › AI
- › Windows Autopilot + Intune + Intune for Education
- › Windows 10&11 Deployment and Management
- › Traditional on-prem deployment and management
- › Consulting & Training



[@petripaavola](https://twitter.com/petripaavola)

<https://github.com/petripaavola>

[Intune.ninja](https://intune.ninja)

[Powershell.ninja](https://powershell.ninja)

Over 23 years of work experience

Current (10+ years):

- › Yodamiitti Oy / Owner Consulting / Training

Past:

- › Aalto university / IT-services Responsible for Workstation service





Key takeaways:

- **Käytännön vinkkejä skriptien tekoon päätelaitehallinnassa**
- **Tosielämän esimerkkejä ja skriptipohjia**



Oliko teillä mielessä skriptaustarpeita

Saatamme tehdä jonkin ehdotuksen nyt, tai sitten emme 😊



Käytännön vinkkejä skriptaukseen

Näkökulma käytännön tekemisessä Windows ja päätelaitehallinnassa



DEMOJA

Esitys on enimmäkseen demoa





Skriptausehdotuksia etukäteen

2	anonymous	host-koneella triggeröityy poistoprosessi (tähän kohtaan ei tarvitse keskittyä sessiossa) 2. host-kone huutelisi jonnekin että "mää lähden nyt" 3. "jonnekin" olisi esim. Azure Logic App, joka sisältäisi tilien poistomagian 4. Poistomagia poistaisi huudelleen host-koneen tilit Tässä voidaan keskittyä myös vain kohtaan 3 Ehdotus 2: Käyttäjälle oletusohjelmat asettava skripti Company Portaliin / Software Centeriin. Ajettavissa käyttäjän oikeuksin ja asettaisi oletusohjelmat pakettiin kuuluvan xml-tiedoston määritysten mukaisesti Case tämän taustalla: kone otettu Autopilotilla käyttöön niin, että käyttäjäprofiilin muodostuessa ei koneella ole vielä MS Officea tai Libre Officea asentuneena (tulee vasta jälkikäteen jaeltuna) -> näiden tukemat tiedostotyypit jää ainakin osittain vaille oletus-assosiaatiota.
3	anonymous	Autopilot HW hashin exporttaus asennetulta koneelta jonnekin, mistä se saadaan otettua takas käyttöön jos kone siivotaan Autopilot kannasta jostain syystä ja halutaan takaisin.
4	anonymous	tietokoneen käyttöraporttia Powercfg /batteryreport tekee kattavaa käyttökelposta raporttia voisko tästä rakentaa raporttia anonymisoida ja uploada jonnekin keskitetyksi. Sitten analysoida sitä powerBI:llä. Sessioiden pituudet, buutit, sleep-pituudet. motivaattorina Ajankäyttö, työhyvinvointi, varsinaisen työajan ulkopuolella tehty työ. Vai onko jotain muuta mielekiintoista ratkaisua tämän aiheen ympärillä.
5	anonymous	Configuration Manager: Powershell scripti joka hakee käyttäjälle julkaistut ohjelmat (available) ja asentaa ne, esim task sequence:a käyttäen. Git hubista löytyy esimerkki joka toimii siten että se hakee ne, en ole onnistunut kohdistamaan package id:tä deployment id:stä kullekin paketille joten asennus osuus ei toimi.
6	anonymous	1. Automatic Primary user update for Microsoft entra ID join devices. 2. Automatically check the unlicensed intune user who uses the device and gets the email daily 3. Automatic Intune backup and documentation. 4. Microsoft configuration high availability.



Skriptausvinkkejä tosielämän use case

www.wpninjas.fi

- AutopilotBranding -skripti
- IntuneWin32 -asennusskriptit
 - EXE ja MSI
 - Logitus
- Intune Win32 Detection Check -skripti
- Intune Win32 Requirements -skripti
- Tulostusajurien ja tulostimien asennus skriptillä
 - 32 vs. 64-bittinen ajoympäristö
- Intune Powershell -skriptit
- Intune Remediations -skriptit
- Intune-hallintaa Powershellillä
 - Case Grouptag ja/tai PrimaryUser
- Skriptien dokumentaatio ja helpit
- Copilotit & ChatGPT & muut työkalut skriptauksen avuksi?



AutopilotBranding -skripti

- AutopilotBranding on alunperin Michael Niehausin tekemä skripti/konsepti, jolla konfiguroidaan asioita Powershellin kautta
 - Samoja asioita on mm. ConfigMgr:ssa konfiguroitu Task Seqencen komennoissa
- Ajatus on ajaa skripti Enrollment Status Page Device-vaiheessa eli skripti ajetaan ennen kuin käyttäjä kirjautuu koneelle, koska skripti konfiguroi asioita mm. Default User -profiiliin
- Yksi tärkeä tai ehkä tärkein komento on esimerkiksi **Set-TimeZone**, jolla asetetaan, mutta ei pakoteta oikea aikavyöhyke
- Michaelin alkuperäinen skripti löytyy <https://github.com/petripaavola/Intune/tree/master/Printing>
- Petrin AutopilotBranding-skripti on tulossa <https://github.com/petripaavola/Intune/tree/master/AutopilotBranding-Yodamiitti>
- Huom! Löysin bugin "kaikista" maailman AutopilotBranding-skripteistä, niin menee hetki, että saan GitHubiin jaettavan version



IntuneWin32 -asennuskriptit

- Powershell-logitus
 - **Start-Transcript & Stop-Transcript** helpoin tapa tehdä lokitusta, jos et käytä muita (esim. CMTrace-yhteensopivia) lokitusmenetelmiä
- Muista, että Intune Win32 -sovellusjakelu ajetaan 32-bittisessä ympäristössä
 - Esimerkkiskriptien alussa on logiikka, joka käynnistää skriptin automaattisesti uudelleen 64-bittiseen ympäristöön
 - Natiivia 64-bittisiä ympäristöä tarvitaan esimerkiksi ajuriasennuksissa **pnputil.exe** ja **dism.exe** kanssa
- Skripteissä on vahva logitus ja vikatilanteen hallinta (Try-Catch)



IntuneWin32 -asennuskriptit EXE

- Lataa Intune Powershell EXE -asennuksen -esimerkkiskripti https://github.com/petripaavola/Intune/blob/master/Apps/install_Audacity_3.5.1-EXE-example.ps1
- Intune install command:
Powershell.exe -ExecutionPolicy Bypass -File install_Audacity_3.5.1-EXE-example.ps1

```
# Install Audacity + MP3 Lame encoder
# IntuneWin32 EXE Powershell script install example
#
#
# Petri.Paavola@yodamiitti.fi
# Windows MVP - Windows and Intune
# 2024-11-01
#
# Original script source:
# https://github.com/petripaavola/Intune/blob/master/Apps/install_Audacity_3.5.1-EXE-example.ps1
#
# Intune install command:
# Powershell.exe -ExecutionPolicy Bypass -File install_Audacity_3.5.1-EXE-example.ps1

#
#
# # #
# #
#

# Configure these values
$SoftwareName = 'Audacity'
$version = "3.5.1"

$InstallerFilePath = "$PSScriptRoot\audacity-win-3.5.1-64bit.exe"
$Arguments = @(
    '/verysilent'
    '/norestart'
)

#
# #
# # #
#
#

# Start script in 64bit environment if script was started in 32bit environment
# Intune Win32 application install process starts in 32bit process by default (2024-11-01)
# Few commands will require native bit command to work. Examples pnputil.exe and dism.exe
# Below 64bit workaround is provided by Oliver Kieselbach
# Original example:
# https://github.com/okieselbach/Intune/blob/master/ManagementExtension-Samples/IntunePSTemplate.ps1
if (-not [System.Environment]::Is64BitProcess)
{
    # start new PowerShell as x64 bit process, wait for it and gather exit code and standard error output
    $sysNativePowerShell = "$($PSHOME.ToLower().Replace("syswow64", "sysnative"))\powershell.exe"
```




IntuneWin32 -asennuskriptit MSI

www.wpninjas.fi

- Lataa Intune Powershell MSI -asennuksen -esimerkkiskripti
https://github.com/petripaavola/Intune/blob/master/Apps/install_LibreOffice_7.0.6-MSI-example.ps1
- Intune install command:
Powershell.exe -ExecutionPolicy Bypass -File install_LibreOffice_7.0.6-MSI-example.ps1

```
# Install LibreOffice 7.0.6 MSI
# IntuneWin32 MSI Powershell script install example
#
# UI Languages: FI, EN, SE
# Proofing languages: English, German, French, Swedish and Spanish
#
#
# Petri.Paavola@yodamiitti.fi
# Microsoft MVP - Windows and Intune
# 2024-11-01
#
# Original script source:
# https://github.com/petripaavola/Intune/blob/master/Apps/install_LibreOffice_7.0.6-MSI-example.ps1
# Intune install command:
# Powershell.exe -ExecutionPolicy Bypass -File install_LibreOffice_7.0.6-MSI-example.ps1
#
# Intune uninstall command:
# msieexec /x {9F9A9C01-5A65-4C2E-A243-FC88C81BC35F} /qn /!v C:\Windows\Logs\Uninstall_LibreOffice_7.0.6_MS
#
# MSI ProductCode={9F9A9C01-5A65-4C2E-A243-FC88C81BC35F}
#
# This is MSI application install example script for MSI-files
# In this case there is so long parameter that it will not fit in Intune install command line

#
# # #
# # #
#
# Configure these values
$SoftwareName = "LibreOffice"
$version = "7.0.6"
$MSIFilePath = "$PSScriptRoot\LibreOffice_7.0.6_Win_x64.msi"

#
# # #
# # #
#
# Start script in 64bit environment if script was started in 32bit environment
# Intune Win32 application install process starts in 32bit process by default (2024-11-01)
# Few commands will require native bit command to work. Examples pnputil.exe and dism.exe
# Below 64bit workaround is provided by Oliver Kieselbach
# Original example:
# https://github.com/okieselbach/Intune/blob/master/ManagementExtension-Samples/IntunePSTemplate.ps1
if (-not [System.Environment]::Is64BitProcess)
{
    # start new PowerShell as x64 bit process, wait for it and gather exit code and standard error output
    $sysNativePowerShell = "$($PSHOME.ToLower()).Replace("syswow64", "sysnative")\powershell.exe"
```



Intune Win32 Detection Check -skripti

- Intune Win32 Custom Detection script
- Jos sovellus löytyy, niin
 - Kirjoita mitä tahansa StdOuttiin. Esim. **Write-Host "Application found"**
 - Poistu skriptistä paluuarvolla 0 **Exit 0**
- Jos sovellusta ei löydy
 - Poistu skriptistä paluuarvolla ei-nolla. Esim. **Exit 1**
- Ääkköset voivat aiheuttaa haasteita natiivi-UTF8 -skripteille
 - Itse teen englanniksi, enkä ikinä käytä ääkkösiä näissä Detection Check -skripteissä
- [https://github.com/petripaavola/Intune/blob/master/Apps/Audacity 3.5.1 Intune Win32 Custom DetectionCheck Script.ps1](https://github.com/petripaavola/Intune/blob/master/Apps/Audacity%203.5.1%20Intune%20Win32%20Custom%20DetectionCheck%20Script.ps1)

```
# Intune Win32 App Audacity Custom Detection check script for version 3.5.1
#
# This returns compliant with equal to 3.5.1 version
#
#
# Petri.Paavola@yodamiitti.fi
# Windows MVP - Windows and Intune
# 2024-11-01
#
# Original script source:
# https://github.com/petripaavola/Intune/blob/master/Apps/Audacity%203.5.1%20Intune%20Win32%20Custom%20DetectionCheck%20Script.ps1

# Check if file exists
if(-not (Test-Path "C:\Program Files\Audacity\audacity.exe")) {
    # File does not exist
    exit 1
}

$FileVersionString = [System.Diagnostics.FileVersionInfo]::GetVersionInfo("C:\Program Files\Audacity\audacity.exe").FileVersion

# Convert , -> .
$FileVersionString = $FileVersionString.Replace(',','.')

#The below line trims the spaces before and after the version name
$FileVersionString = $FileVersionString.Trim();

# Cast variable to [version]
$FileVersion = [version]$FileVersionString

# We could also get version with this command in most other cases
#$file = Get-ChildItem "C:\Program Files\Audacity\audacity.exe"
#[version]$FileVersion = $file.versioninfo.fileversion

if ($FileVersion -eq "3.5.1.0" ) {
    # App detected

    # Write the version to STDOUT by default
    # Write anything to StdOut and exit 0 to make application show as detected
    Write-Host "$FileVersion"
    exit 0
}
else {
    # App NOT detected

    #Exit with non-zero failure code
    exit 1
}
```



Intune Win32 Requirements -skripti

- Requirements-skripti palauttaa arvoja eri muuttujatyypeillä Intunelle

Select output data type * ⓘ

Operator * ⓘ

Value

- String
- Date and Time
- Integer
- Floating Point
- Version
- Boolean

- Requirement-rulesssa konfiguroidaan miten paluarvoja tulkitaan, jotta päätellään onko sovellus **Applicable** vai **Not applicable**

Add a Requirement rule

Create a requirement.

Requirement type ⓘ

Script name *

Script file ⓘ

Script content

```
# Requirement script DEMO
# This is not actually doing anything

# Do some custom checks

# Return true or false
$true
```

Run script as 32-bit process on 64-bit clients ⓘ

Run this script using the logged on credentials ⓘ

Enforce script signature check ⓘ

Select output data type * ⓘ

Operator * ⓘ

Value



- Suurin kompastuskivi tulostimien asennusskripteissä on tulostusajurin asennus
- Koska Intune Win32 -sovellusasennus ajetaan 32-bittisessä ympäristössä ja **pnputil.exe** (ja mm. **dism.exe**) **vaatii natiivin 64-bittisen ajoympäristön**
- Workaround on lisätä asennusskriptin alkuun tarkistus, joka käynnistää skriptin itsestään 64-bittisenä
 - Toinen workaround olisi ajaa skriptissä pnputil.exe sysnative-polusta:
C:\Windows\SysNative\pnputil.exe /add-driver "\$PSScriptRoot\Drivers\eSf6u.inf" /install
 - Tai laittaa Intunen Win32 install command ajamaan 64-bittinen Powershell komennolla:
%windir%\Sysnative\WindowsPowerShell\v1.0\powershell.exe -ExecutionPolicy Bypass -File install_Audacity_3.5.1-EXE-example.ps1
- Lue teoriaa tulostimiin ja bittisyyteen liittyen Rudy Oomsin blogista
<https://call4cloud.nl/deploy-printer-drivers-intune-win32app/>
- Petrin tulostusajuri ja suoratulostuksen asennusskripti tulee GitHubiin
 - Kävi taas niin, että kun piti jakaa skripti, niin pitihän se tehdä sitten "uusiksi" paremmin eli skripti on tulossa, jos GitHub näyttää tyhjää 😊
- <https://github.com/petripaavola/Intune/tree/master/Printing>

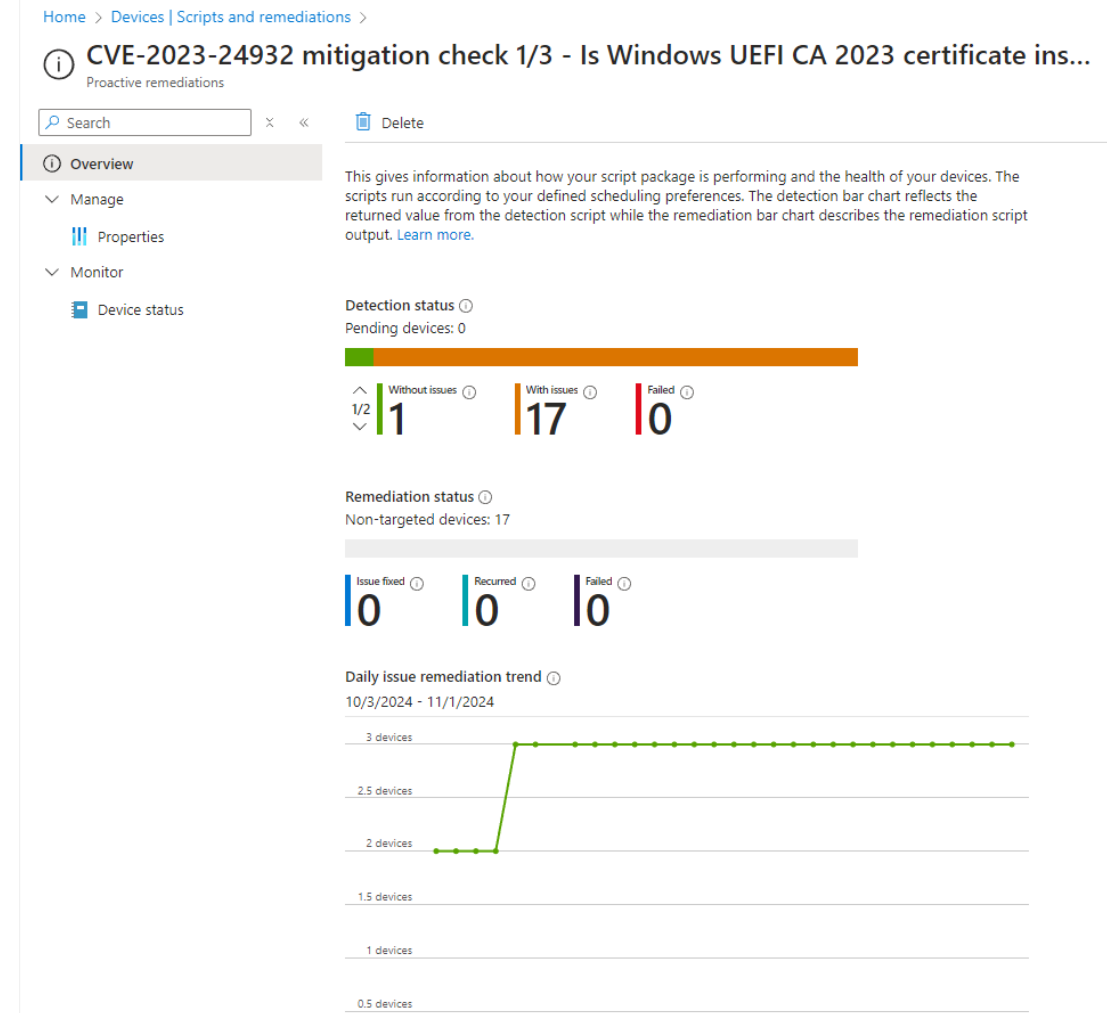


- Huomio, että laitteille kohdennetut Powershell-skriptit ajautuvat
 - Kerran SYSTEM:nä eli konetilinä
 - **Kerran jokaista uutta kirjautunutta käyttäjää kohden**
- "Lähes aina" Powershell-skriptit on ajateltu ajettavan vain kerran, eikä ole tiedostettu, että skripti ajautuu useamman kerran
- Remediation-skripti tarjoaa aidon Run once -vaihtoehdon
- Powershell-skripti ajautuu ensimmäisenä Autopilot-enrollmentissa (vaikka niin ei luvata)
- Powershell-skripti on blocking-tyyppinen Autopilot-enrollmentille eli jumissa oleva Powershell-skripti pysäyttää myös koko Autopilot enrollmentin

2024-09-17 04:27:46.2688944	Info			##### Device ESP phase #####	
2024-09-17 04:27:46.7488838	Info	User	Logon	APVM-8726300880\defaultuser0	
2024-09-17 04:28:11.1399126	Success	Powershell script	Execute	Import certificate - Nordic Virtual Summit 2024 for user System	21
2024-09-17 04:31:20.7350431	Success	Powershell script	Execute	Powershell script simulate long runtime (180 seconds) for user System	190
2024-09-17 04:31:20.7350431	Warning	Powershell script	Execute	Long Powershell script runtime found (190 seconds)	190
2024-09-17 04:31:43.1444553	Success	Powershell script	Execute	Powershell Script Runas System for user System	22
2024-09-17 04:32:02.7813937	Info	Apps	Required in ESP	4 Apps to install at ESP phase	
2024-09-17 04:32:02.7813937	Info	Win32App	Required in ESP	AutopilotBranding Yodamiitti v2.2	
2024-09-17 04:32:02.7813937	Info	Win32App	Required in ESP	Notepad++ 8.5.5 x64	
2024-09-17 04:32:02.7813937	Info	Win32App	Required in ESP	7-Zip 23.01 (x64 edition)	
2024-09-17 04:32:02.7813937	Info	Win32App	Required in ESP	Nordic Virtual Summit 2024 Application Install - SuperSecret install command	
2024-09-17 04:32:15.5053224	Not Detected	Win32App	Required Install	AutopilotBranding Yodamiitti v2.2	
2024-09-17 04:32:50.8268926	Success	Win32App	Required Install	AutopilotBranding Yodamiitti v2.2 (0 Success)	24



- Mahdollista ajastaa aito Run once tai ajaa x tunnin/päivän välein
- Detection- ja Remediation-skriptit
 - Remediation ei ole pakollinen
 - Usein tehdäänkin vain Detection, joka samalla konfiguroi halutun Asian
- Remediation-skriptejä voi ajaa yksittäiselle laitteelle On-Demand Intune-konsolista (laite action)
- Hyvä tapa tarkistaa jokin asia ja korjata tarvittaessa
- Hyvä tapa tarkistaa jokin asia ja saada siitä raportti (ei tehdä mitään korjausta)
- Paaaaaljon esimerkkiskriptejä löytyy mm. <https://github.com/JayRHa/EndpointAnalyticsRemediationScripts>





- Helpoin keino tehdä Intune-hallinnan Powershell-skriptejä on käyttää Petrin **ClipboardTools-EdgeDebuggerMGGraphPowerShellScript** -työkalua, joka löytyy ClipboardTools-moduulista
- Asenna ClipboardTools Powershelliin komennolla:
Install-Module -Name ClipboardTools -Scope CurrentUser
- Mene Edgessä (tai Chromessa) Intunessa halutun sivun lähelle (eli ennen kuin menet sivulle)
- F12 -> Developer Tools
- Mene halutulle Intune-sivulle tai tee haluttu Intune-toiminto (esim. tallenna Grouptag-change)
- Etsi Developer Tools -> Network kohdasta oikea Graph API -kutsu
- Right click -> Copy as Powershell
- Powershell-ikkunassa aja komento:
ClipboardTools-EdgeDebuggerMGGraphPowerShellScript
- Pastea työpöydälle mankeloitu Powershell-pätkä skripti-tiedostoon -> PROFIT! 😊



Intune-hallintaa Powershellillä

Microsoft Intune admin center

Home > Devices | Windows > Windows | Windows devices > APVM-8726300880 >

Run remediation (preview)

APVM-8726300880

Deploy a remediation script package to this device using both a detection and remediation scripts. [Learn more about the remediation script packages.](#) To manage the script packages available on this screen, go to [Proactive remediations.](#)

Search

Script package name ↓	Description	Details
<input type="checkbox"/> Detect Lenovo BIOS password state		View details
<input type="checkbox"/> Disable Built-in Administrator account		View details
<input checked="" type="checkbox"/> Enable Built-In Administrator account		View details
<input type="checkbox"/> Gather device Inventory		View details
<input type="checkbox"/> Get Monitor Serial Numbers and Details		View details
<input type="checkbox"/> List WLAN Profiles		View details
<input type="checkbox"/> Proactive Remediations DEMO		View details
<input type="checkbox"/> Restart stopped Office C2R svc	If service is stopped, try to start it. If not auto-start, change...	View details
<input type="checkbox"/> Set Local Admin from CSV (case Lukio-ko...		View details
<input type="checkbox"/> Update stale Group Policies	If GP refresh was > 7 days ago, then gpupdate. IT can cust...	View details
<input type="checkbox"/> Yodamiitti Proactive Remediations DEMO	Detection and Remediation script DEMO	View details
<input type="checkbox"/> Yodamiitti Proactive Remediations RunOn...		View details

Run remediation

graph

Preserve log Disable cache No throttling

Fetch/XHR Doc CSS JS Font Img Media Manifest WS Wasm Other

50000 ms 100000 ms 150000 ms 200000 ms 250000 ms

Name Headers Payload Preview Response Initiator

19 / 94 requests 24.3 kB / 30.0 kB Request Headers



Intune-hallintaa Powershellillä

Microsoft Intune admin center

Home > Devices | Windows > Windows | Windows devices >

APVM-8726300880

Search

Overview

- Manage
 - Properties
- Monitor
 - Hardware
 - Discovered apps
 - Device compliance
 - Device configuration
 - App configuration
 - Local admin password
 - Recovery keys
 - User experience
 - Device diagnostics
 - Group membership
 - Managed Apps
 - Filter evaluation
 - Enrollment
 - Remediations (preview)
 - Device query

Run remediation pending...

Essentials

Device name: APVM-8726300880

Management name: etunimi.sukunimi_Windows_9/17/2024_8:51 PM

Ownership: Corporate

Serial number: 0858-3566-8785-7423-8726-3008-80

Phone number: ---

Device manufacturer: Microsoft Corporation

Primary user: [Etunimi Sukunimi](#)

Enrolled by: [Etunimi Sukunimi](#)

Compliance: Compliant

Operating system: Windows

Device model: Virtual Machine

Last check-in time: 11/1/2024, 5:10:03 PM

Remote assistance: Remote Help

Device actions status

Network

graph

Headers

Request URL: [https://graph.microsoft.com/beta/deviceManagement/managedDevices\('f7f148ee-e6e1-4146-93f7-6e8406b48a95'\)/initiateOnDemandProactiveRemediation](https://graph.microsoft.com/beta/deviceManagement/managedDevices('f7f148ee-e6e1-4146-93f7-6e8406b48a95')/initiateOnDemandProactiveRemediation)

POST

204 No Content

40.126.53.24:443

strict-origin-when-cross-origin

Copy

- Copy URL
- Copy as cURL (cmd)
- Copy as cURL (bash)
- Copy as PowerShell
- Copy as fetch
- Copy as fetch (Node.js)
- Copy response
- Copy stack trace
- Copy all listed URLs
- Copy all listed as cURL (cmd)
- Copy all listed as cURL (bash)
- Copy all listed as PowerShell
- Copy all listed as fetch
- Copy all listed as fetch (Node.js)



```
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows  
  
PS C:\Users\pmpaavol> ClipboardTools-EdgeDebuggerMGGraphPowerShellScript  
PS C:\Users\pmpaavol>
```



Intune-hallintaa Powershellillä

```
new 1 x
1 # Connect to Graph API using Microsoft Graph module
2
3 # Install Microsoft.Graph.Authentication module with command
4 # Install-Module -Name Microsoft.Graph.Authentication -Scope CurrentUser
5 #Connect-MgGraph
6
7 Connect-MgGraph -Scopes "DeviceManagementManagedDevices.Read.All", "DeviceManagementApps.Read.All",
8 "DeviceManagementConfiguration.Read.All", "User.Read.All", "Group.Read.All", "GroupMember.Read.All", "Directory.Read.All"
9
10 $Uri =
11 "https://graph.microsoft.com/beta/deviceManagement/managedDevices('f7f148ee-e6e1-4146-93f7-6e8406b48a95')/initiateOnDemandPro
12 activeRemediation"
13
14 $Body = @"
15 {
16   "ScriptPolicyId": "3d7a62e1-001e-4365-9a84-95ff77d79dac"
17 }
18 "@
19
20 # Note! There seems to be either a bug or a feature with POST requests with Invoke-MgGraphRequest
21 # In some testings command succeeds but does not return anything to PowerShell pipeline
22 # but with -Debug option you can see that the data has been fetched from Graph API
23 # For now one workaround is to save data to text file which we'll do here
24
25 # Get data from Graph API
26 # Original request
27 #MgGraphRequest = Invoke-MgGraphRequest -Uri $Uri -Body $Body.ToString() -Method 'POST' -OutputType PSObject
28
29 # Workaround to save data to random named text file first with parameter -OutputFilePath
30 $OutputFilePath = "$($pwd.path)/MgGraphRequest_$(Get-Random).json"
31
32 $MgGraphRequest = Invoke-MgGraphRequest -Uri $Uri -Body $Body.ToString() -Method 'POST' -OutputFilePath $OutputFilePath
33
34 # Read and convert json data from temporary text file
35 $MgGraphRequest = Get-Content $OutputFilePath -Raw | ConvertFrom-Json
36
37 # Remove temporary file
38 Remove-Item -Path $OutputFilePath
39
40 if($MgGraphRequest) {
41   # Print results
42   # Usually you want to check values from $MgGraphRequest.value
43   Write-Host "Print variable MgGraphRequest values:"
44   $MgGraphRequest
45 }
46 } else {
47   Write-Host "Did not get any results from Graph API!" -ForegroundColor Yellow
48 }
49
50
```

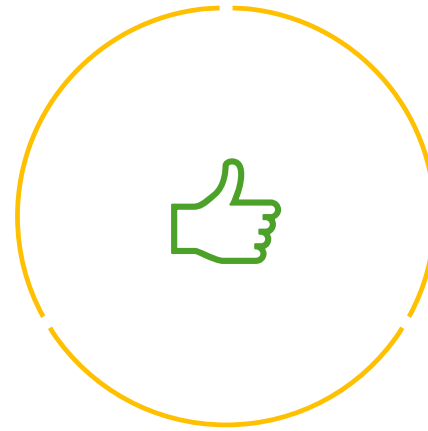


Check out my GitHub for Community Tools downloads and documentation

[Powershell.ninja](https://github.com/petripaavola)



<https://github.com/petripaavola>



Thank You

