TunedIn:London

TUNEDIN
MEETUPS

TUNEDIN
MEETUPS

Ben Whitmore

Intune Management Extension Deep Dive

TunedIn: London

Microsoft Reactor, Paddington, London
8th December 2023

Thank you to our Sponsors & Host

Microsoft

TUNEDIN
MEETUPS

TunedIn: London

8th December 2023

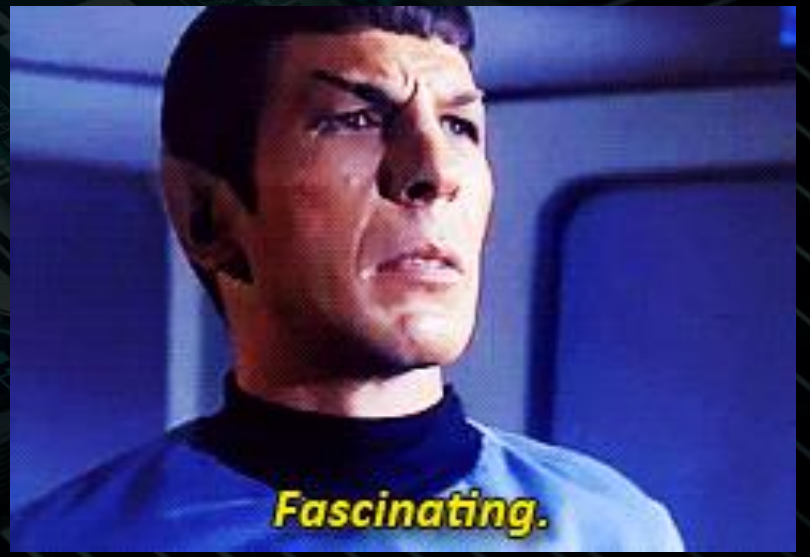
Agenda

- What is the IME?
- How is the IME installed?
- How does the IME process app policy?
- How does the IME process scripts?
- Digging into win32app policy event state messages
- Try again, how the IME handles failures and retries
- Invoke IME actions remotely like a boss
- Inventory
- Q&A

What is the IME

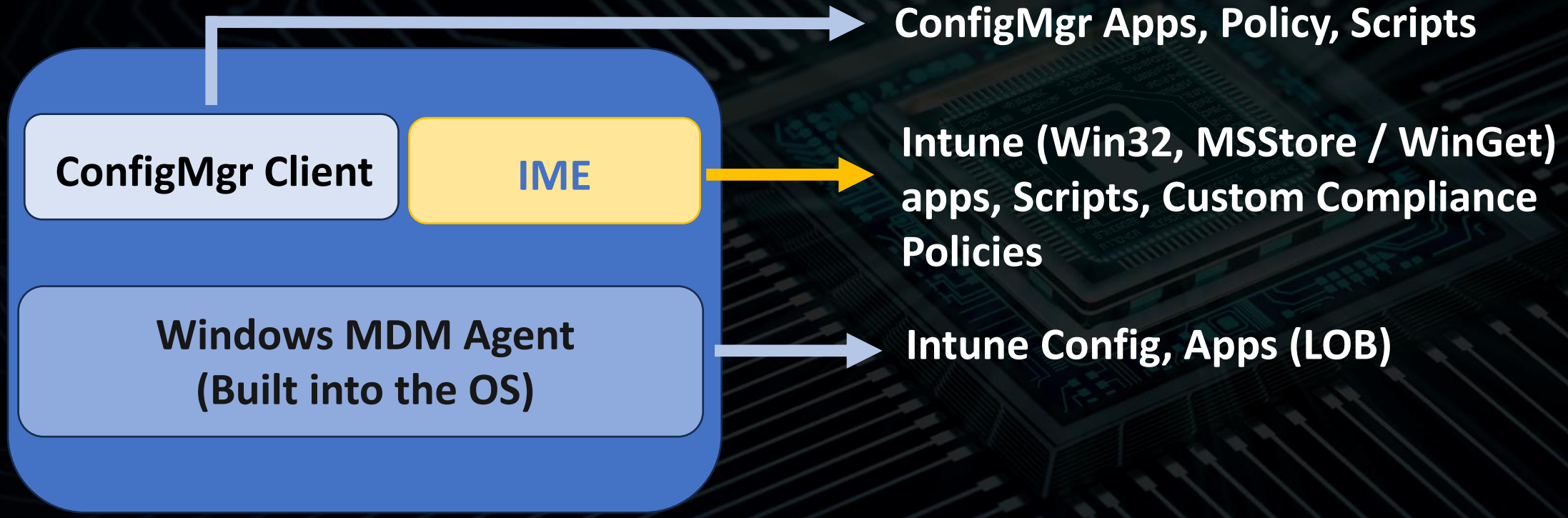
- A component installed by Intune and leveraged by Intune
- Used to deploy and execute PowerShell scripts and install Win32 applications on Windows devices that are enrolled in Intune

Microsoft 365 Apps for enterprise - en-us	Microsoft Corporation
Microsoft ASP.NET Core 6.0.23 - Shared Framework (x...	Microsoft Corporation
Microsoft Azure Storage Explorer version 1.31.2	Microsoft Corporation
Microsoft Edge WebView2 Runtime	Microsoft Corporation
Microsoft Intune Management Extension	Microsoft Corporation
Microsoft OneDrive	Microsoft Corporation
Microsoft PowerBI Desktop (x64)	Microsoft Corporation
Microsoft System CLR Types for SQL Server 2019	Microsoft Corporation
Microsoft Teams	Microsoft Corporation



What is the IME?

Where does the IME sit in the Microsoft RMM **agent** stack?



What is the IME?

IntuneManagementExtension.log

- Activities and processes related to the execution of scripts
- Installation of apps deployed through Microsoft Intune
- Provides insights into how the IME is functioning

AgentExecutor.log

- Execution of scripts deployed through Microsoft Intune

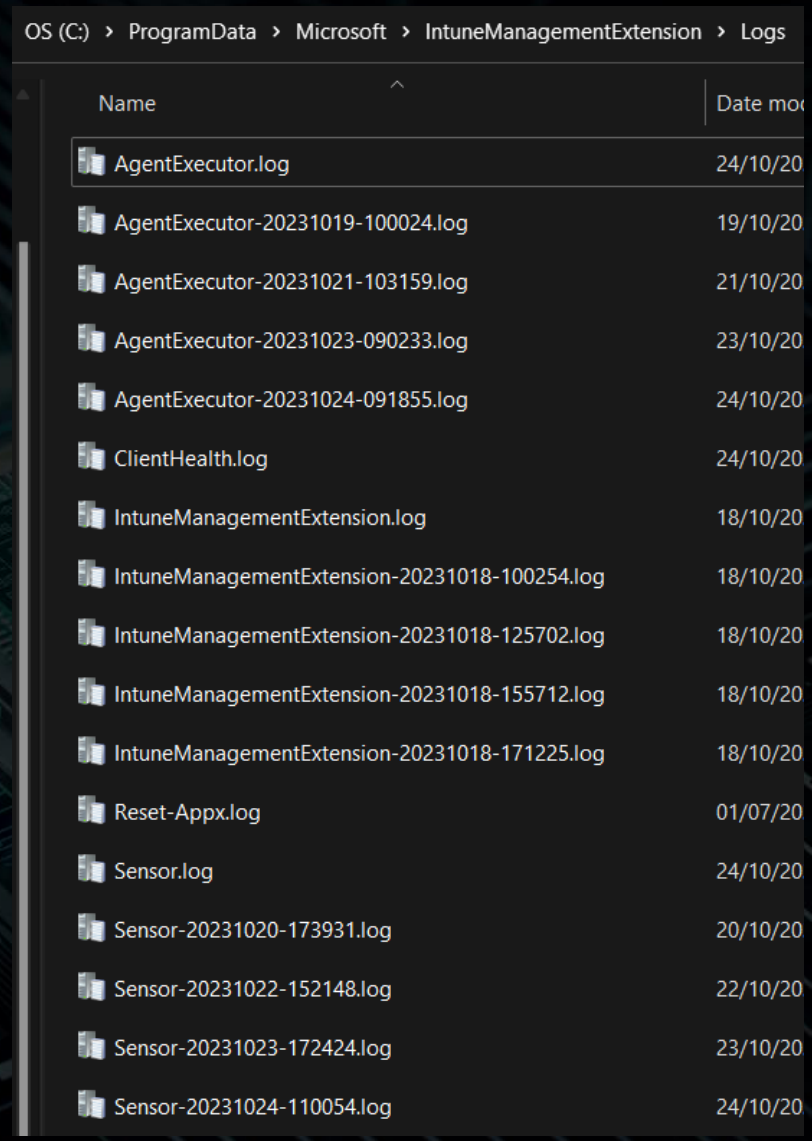
HealthScripts.log

- Proactive remediation scripts deployed through Microsoft Intune

ClientHealth.log

- Client health activities for the IME i.e. Is the service is running

Win32AppInventory.log is back!



C:\ProgramData\Microsoft\IntuneManagementExtension\Logs

What is the IME?

Increase IME Logging

HKLM\SOFTWARE\Microsoft\IntuneWindowsAgent\Logging

LogMaxHistory – Default is 3
LogMaxSize – Default is 3145728

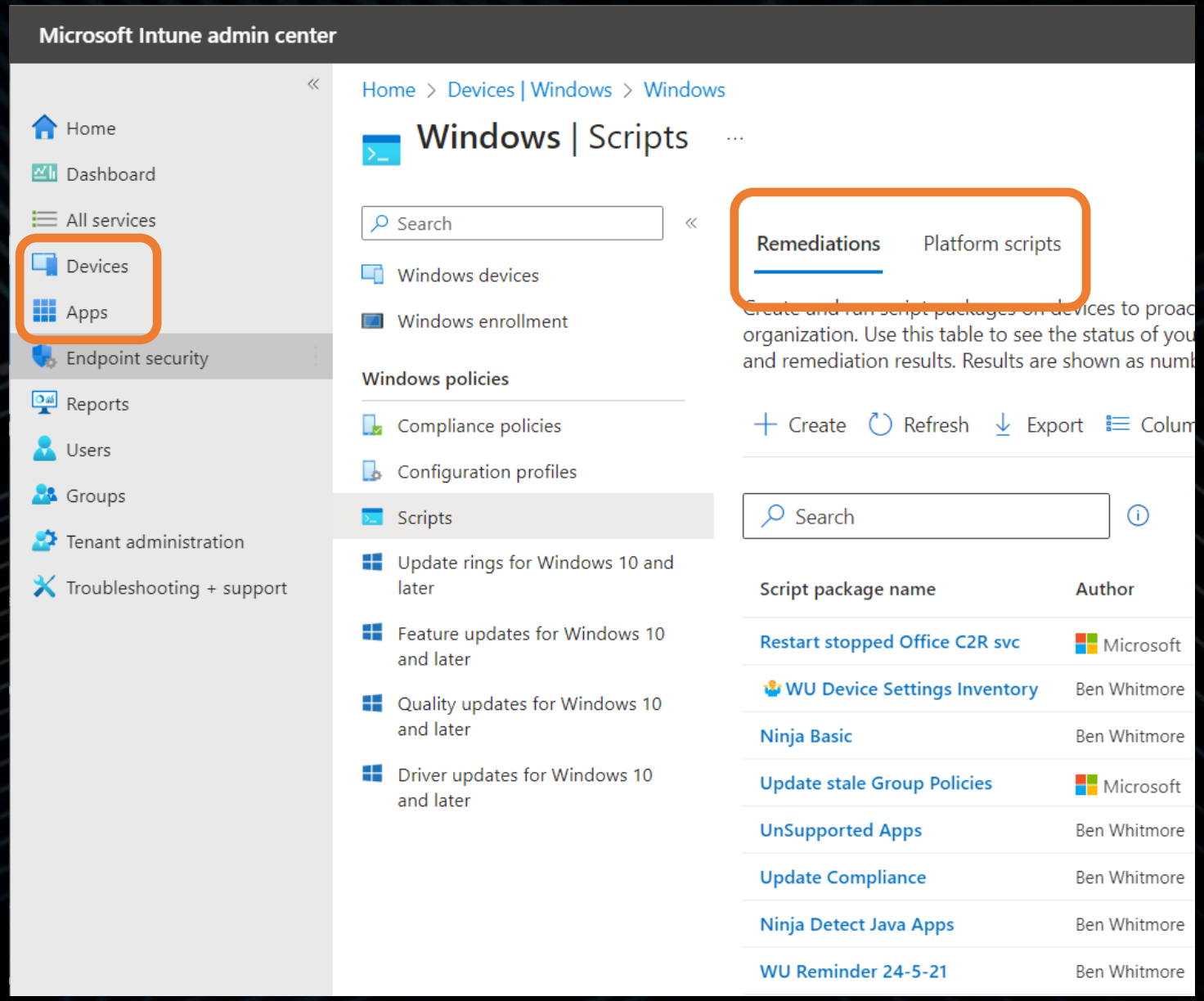
(Used to be 2 x 2MB logs)

https://github.com/okieselbach

How is the IME installed?

The IME is installed when a managed device is targeted with either a:-

1. PowerShell Script or Proactive Remediation
2. Win32 app or Microsoft Store app
3. Custom compliance settings



How is the IME installed?

The IME is installed, from an MSI, via the OMA-DM channel using the:-

EnterpriseDesktopAppManagement Configuration Service Provider (CSP)

This CSP is used to handle enterprise desktop application management tasks, such as querying installed enterprise applications, installing applications, or removing applications.

Device/MSI/{ProductID}

Scope	Editions	Applicable OS
✅ Device ✅ User	✅ Pro ✅ Enterprise ✅ Education ✅ Windows SE ✅ IoT Enterprise / IoT Enterprise LTSC	✅ Windows 10, version 1511 [10.0.10586] and later

Device — Copy

```
./Device/Vendor/MSFT/EnterpriseDesktopAppManagement/MSI/{ProductID}
```

The MSI product code for the application.

Description framework properties:

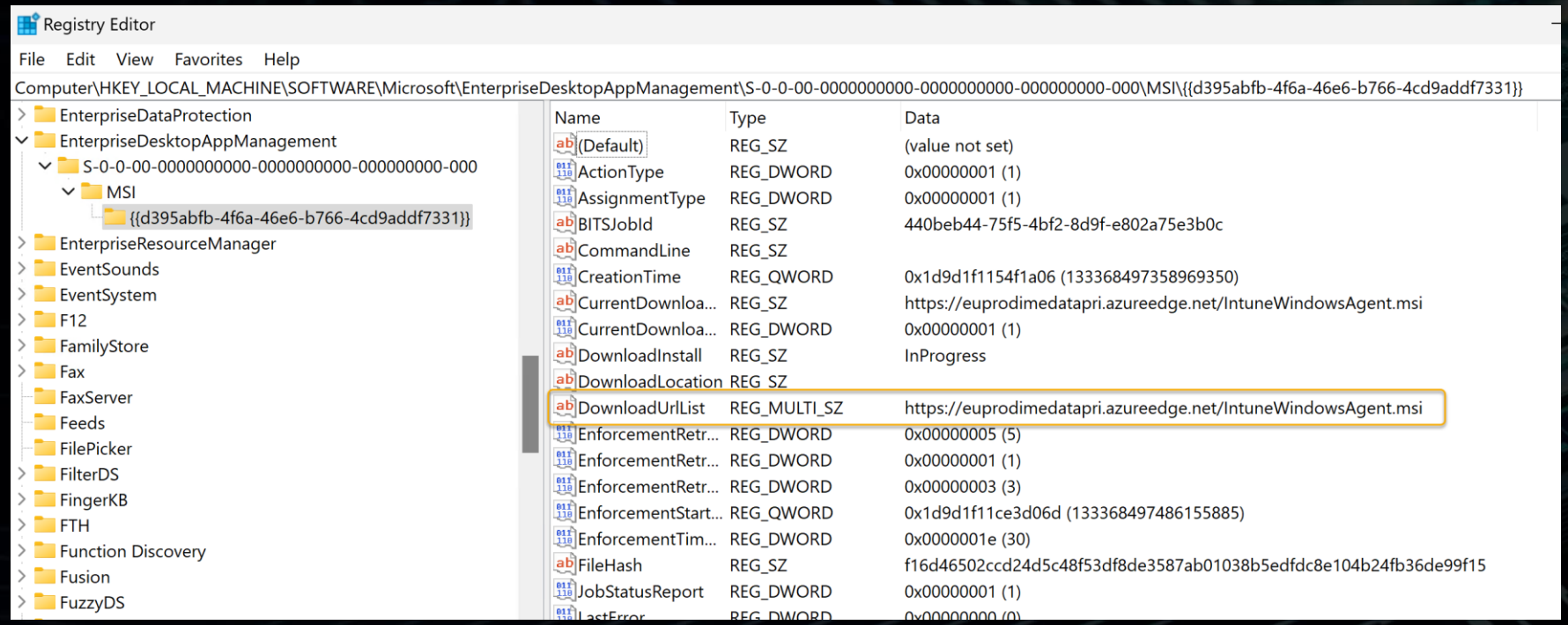
Property name	Property value
Format	node
Access Type	Add, Delete, Get
Atomic Required	True
Dynamic Node Naming	UniqueName: The MSI product code for the application.

https://learn.microsoft.com/en-us/windows/client-management/mdm/enterprisedesktopappmanagement-csp

How is the IME installed?

You can track the install via the local registry

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\EnterpriseDesktopAppManagement\
S-0-0-00-0000000000-0000000000-000000000-000\MSI



New versions are rolled out automatically

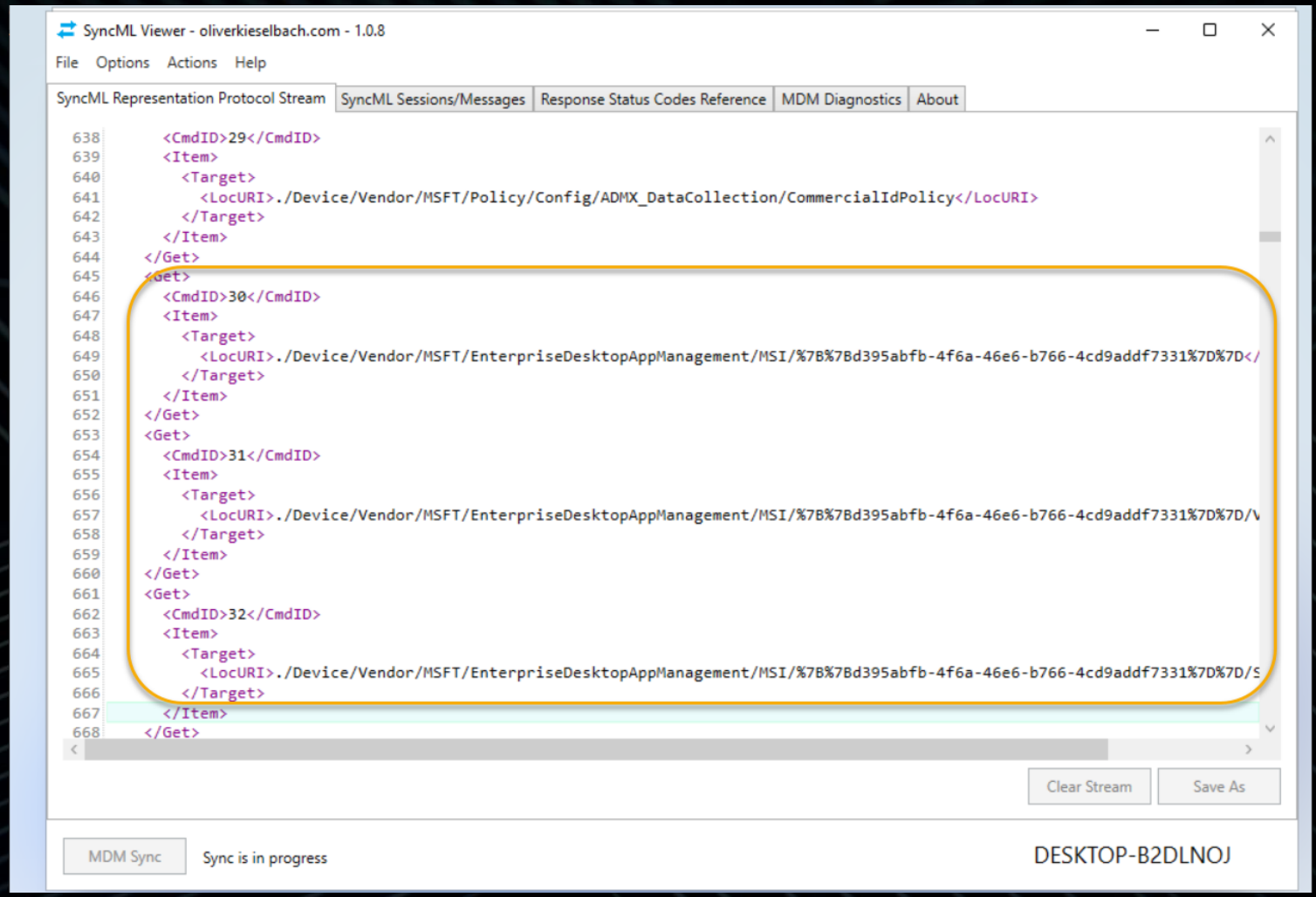
Snapshots are a curse

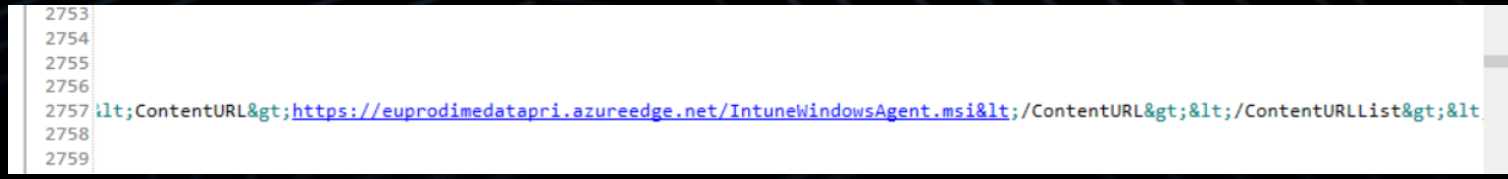
https://euprodimedatasec.azureedge.net/IntuneWindowsAgent.msi

How is the IME installed?

View the SyncML message where OMA-DM initiates the MSI install for the IME

https://github.com/okieselbach/SyncMLViewer

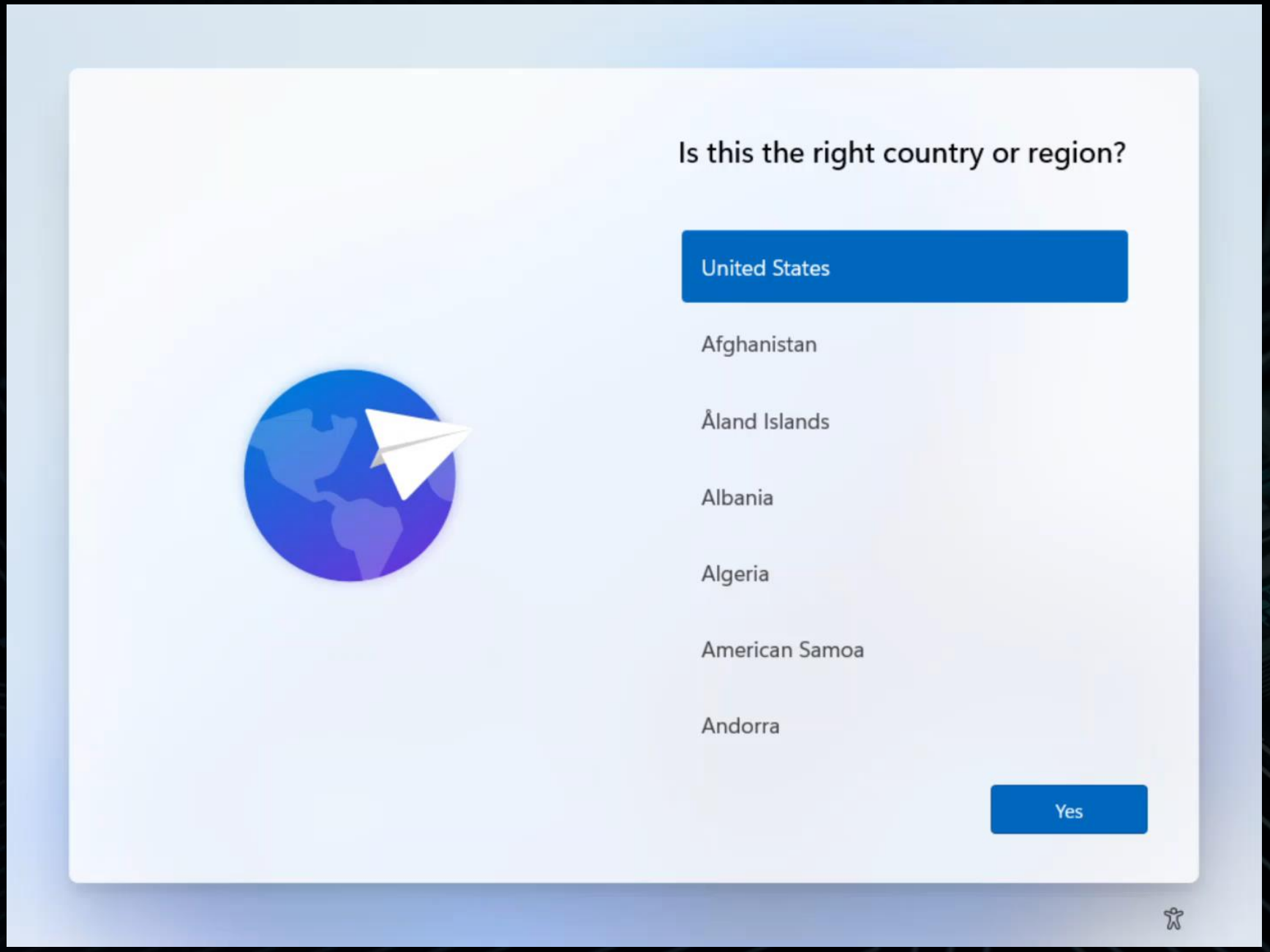




TUNEDIN MEETUPS

TunedIn:London

How is the IME installed?


Is this the right country or region?
United States
Afghanistan
Åland Islands
Albania
Algeria
American Samoa
Andorra
Yes

How does the IME process app policy?

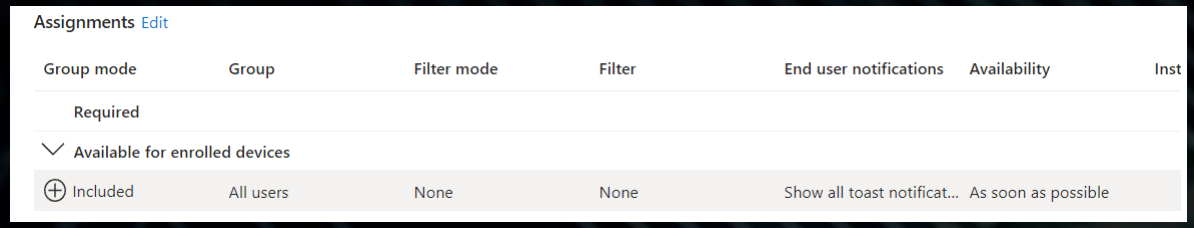
Policy is deployed

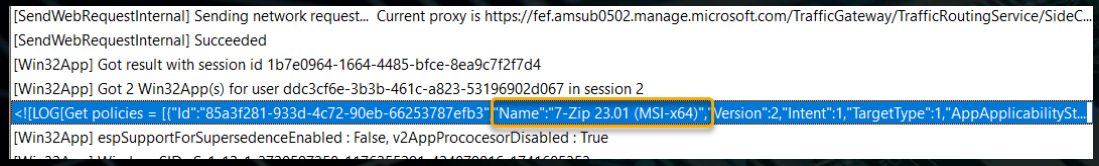
Policy reaches the device

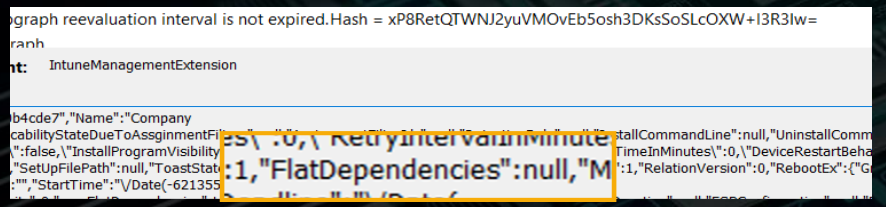
Dependencies checked

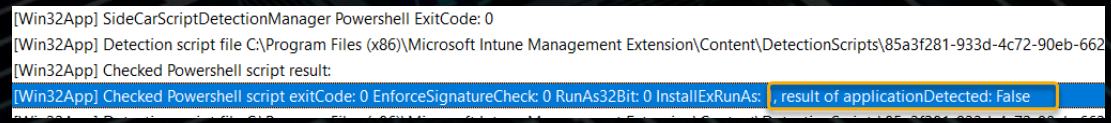
Detection rule checked

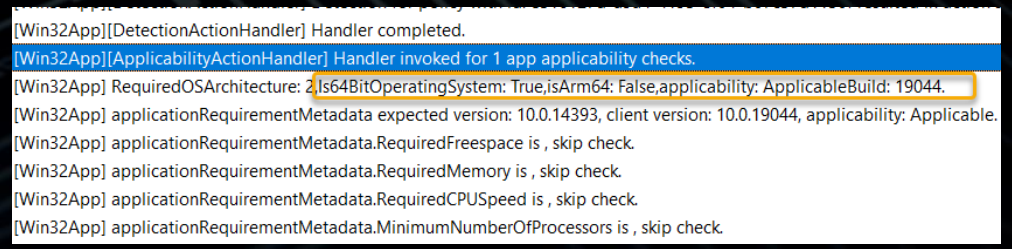
Applicability and requirements checked











How does the IME process app policy?

IME begins content download (content phase 1)

```
[Win32App] DownloadType = 2, DownloadUrl = http://swdb02-mscdn.manage.microsoft.com/4cfb7791-c1d9-4686-adce-60b9cffc46e6/f2d49725-e826-484d-8...
[Win32App] Content cache miss for app (id = 85a3f281-933d-4c72-90eb-66253787efb3, name = 7-Zip 23.01 (MSI-x64)), start downloading...
[Win32App DO] start creating a new download job, FileId = 501FCB7D-A970-4E34-A753-4B48FE5D8BEF_4cfb7791-c1d9-4686-adce-60b9cffc46e6_f2d49725-e8...
[Win32App DO] destination path = C:\Program Files (x86)\Microsoft Intune Management Extension\Content\Incoming\85a3f281-933d-4c72-90eb-66253787efb3...
DO Job Id = 8da972a9-e339-404c-a320-119d4a99be5b
```

IME validate the package and decrypts content (content phase 2)

```
[Win32App DO] DO download and decryption is successfully done
[Win32App] Downloaded file size 2,649,804.00
[Win32App] Downloaded file time 5,903.00
[Win32App] Start unzipping.
[Win32App] Unzipping file on session 2 from C:\Program Files (x86)\Microsoft Intune Management Extension\Content\Staging\85a3f281-933d-4c72-90eb-66253...
Cleaning up staging content C:\Program Files (x86)\Microsoft Intune Management Extension\Content\Staging\85a3f281-933d-4c72-90eb-66253787efb3_2
```

Content cleaned up and moved (content phase 3)

```
[Win32App] Unzipping file on session 2 from C:\Program Files (x86)\Microsoft Intune Management Extension\Content\Staging\85a3f281-933d-4c72-90eb-66253... IntuneManagementEx 29/08/2023 12:37 32 (0x20)
Cleaning up staging content C:\Program Files (x86)\Microsoft Intune Management Extension\Content\Staging\85a3f281-933d-4c72-90eb-66253787efb3_2 IntuneManagementEx 29/08/2023 12:37 32 (0x20)
[Win32App] Saving DownloadTime for user: ddc3cf6e-3b3b-461c-a823-53196902d067 IntuneManagementEx 29/08/2023 12:37 32 (0x20)
Date/Time: 29/08/2023 12:37:35  Component: IntuneManagementExtension
Thread: 32 (0x20)  Source:
[Win32App] Unzipping file on session 2 from C:\Program Files (x86)\Microsoft Intune Management Extension\Content\Staging\85a3f281-933d-4c72-90eb-66253787efb3_2\85a3f281-933d-4c72-90eb-66253787efb3_2
```

C:\Windows\IMECache

App install begins (Detection run again)

```
[Win32App][DownloadActionHandler] Handler completed with action status: Success.
[Win32App][DetectionActionHandler] Handler invoked for 1 app detections.
[Win32App][DetectionActionHandler] Detection running for policy with id: 85a3f281-933d-4c72-90eb-66253787efb3.
[Win32App] ProcessDetectionRules starts
Date/Time: 29/08/2023 12:37:35  Component: IntuneManagementExtension
Thread: 32 (0x20)  Source:
[StatusService] Sending an update to user ddc3cf6e-3b3b-461c-a823-53196902d067 via callback for app: 85a3f281-933d-4c72-90eb-66253787efb3. Applicability: Applicable Status: Installing, ErrorCode: null
```

App install continues

```
[Win32App] Handler invoked with execution type: Install for policy -90eb-66253787efb3 and version: 2.
[Win32App] 461c-a823-53196902d067.
[Win32App][GRSManager] Saving GRS values to storage.
[Win32App][GRSManager] Saved GRS value: 08/29/2023 11:37:36 to path: ddc3cf6e-3b3b-461c-a823-53196902d067\GRS\nmdKzsJy7z12yxRyn9wU1hnVYI92zxkV...
[Win32App][Win32AppExecutionExecutor] Using content at: C:\Windows\IMECache\85a3f281-933d-4c72-90eb-66253787efb3_2.
[Win32App] ExecuteWithRetry Parsing InstallEx...
[Win32App] ===Step=== Execute retry 0
[Win32App] ===Step=== InstallBehavior RegularWin32App, Intent 3, UninstallCommandLine PatchMyPC-ScriptRunner.exe /UninstallPackage
```

How does the IME process app policy?

Reboot Manager checks exit code for reboot requirement and content cleaned up

Cleaning up staged content C:\Windows\IMECache\85a3f281-933d-4c72-90eb-66253787efb3_2

[Win32App][RebootManager] Removing user with id: ddc3cf6e-3b3b-461c-a823-53196902d067 from the available check-in schedule.

Detection reevaluated

[Win32App] Detection script file C:\Program Files (x86)\Microsoft Intune Management Extension\Content\DetectionScripts\85a3f281-933d-4c72-90eb-66253787...

[Win32App] Checked Powershell script result: Detected

[Win32App] Checked Powershell script exitCode: 0 EnforceSignatureCheck: 0 RunAs32Bit: 0 InstallExRu... f applicationDetected: True

[Win32App] Detection script file C:\Program Files (x86)\Microsoft Intune Management Extension\Con...

Compliance state set in registry and sent to the Intune service

[Win32App][ReportingManager] Detection state for app with id: 85a3f281-933d-4c72-90eb-66253787efb3 has been updated. Report delta: {"DetectionState":{"...

[Win32App][ReportingManager] Sending status to company portal based on report: {"ApplicationId":"85a3f281-933d-4c72-90eb-66253787efb3","ResultantApp...

[Win32App][StatusServiceReportsStore] Saved AppInstallStatusReport for user ddc3cf6e-3b3b-461c-a823-53196902d067 for app 85a3f281-933d-4c72-90eb-66...

[StatusService] Sending update from StatusServicePublisher.

Toast Success or Company Portal update

Microsoft Intune Management Extension

Byteben Lab software distr...

7-Zip 23.01 (MSI-x64) was i... successfully.

12:37

Installed

7-Zip 23.01 (MSI-x64)

Igor Pavlov

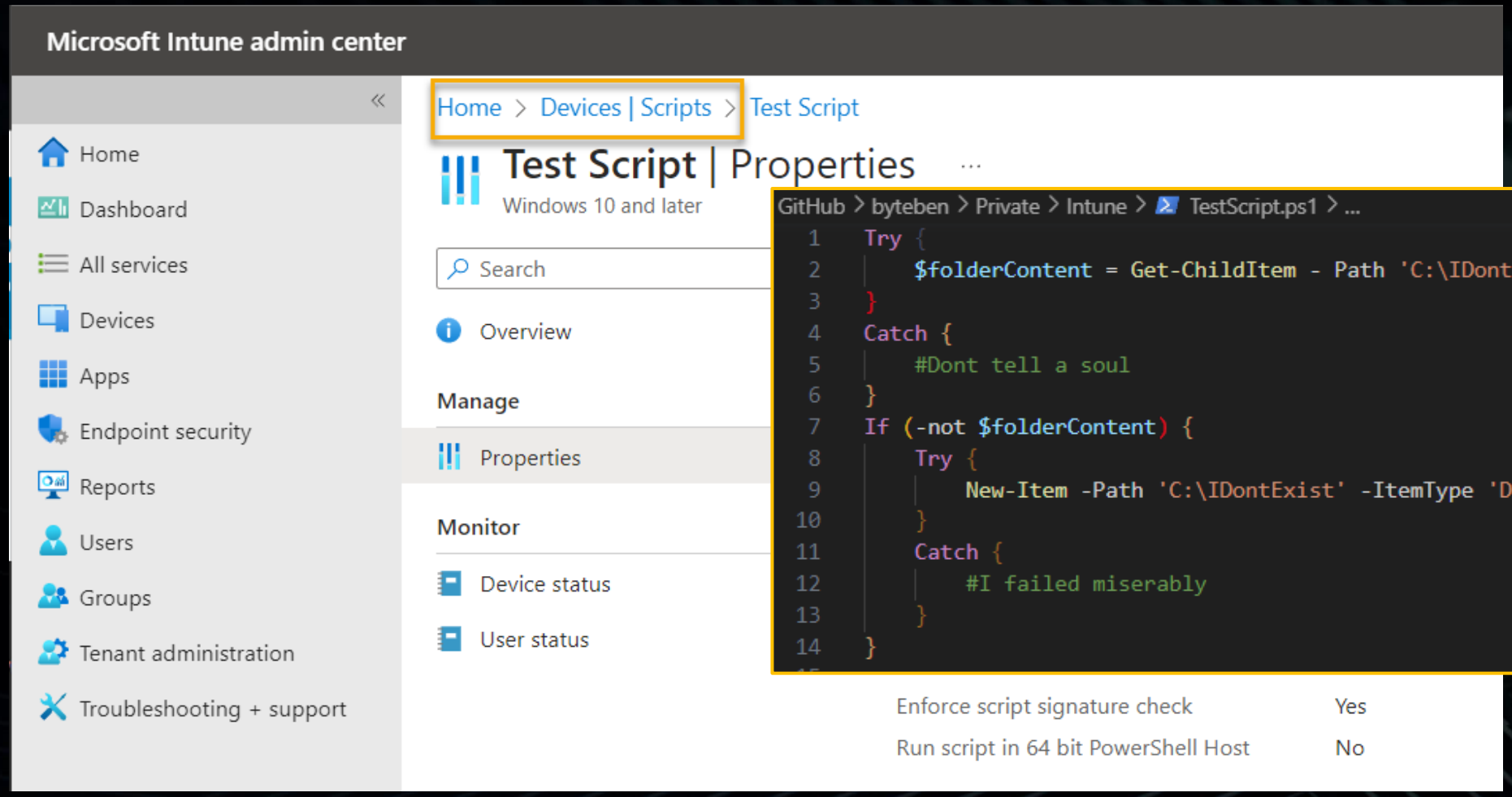
Reinstall ...

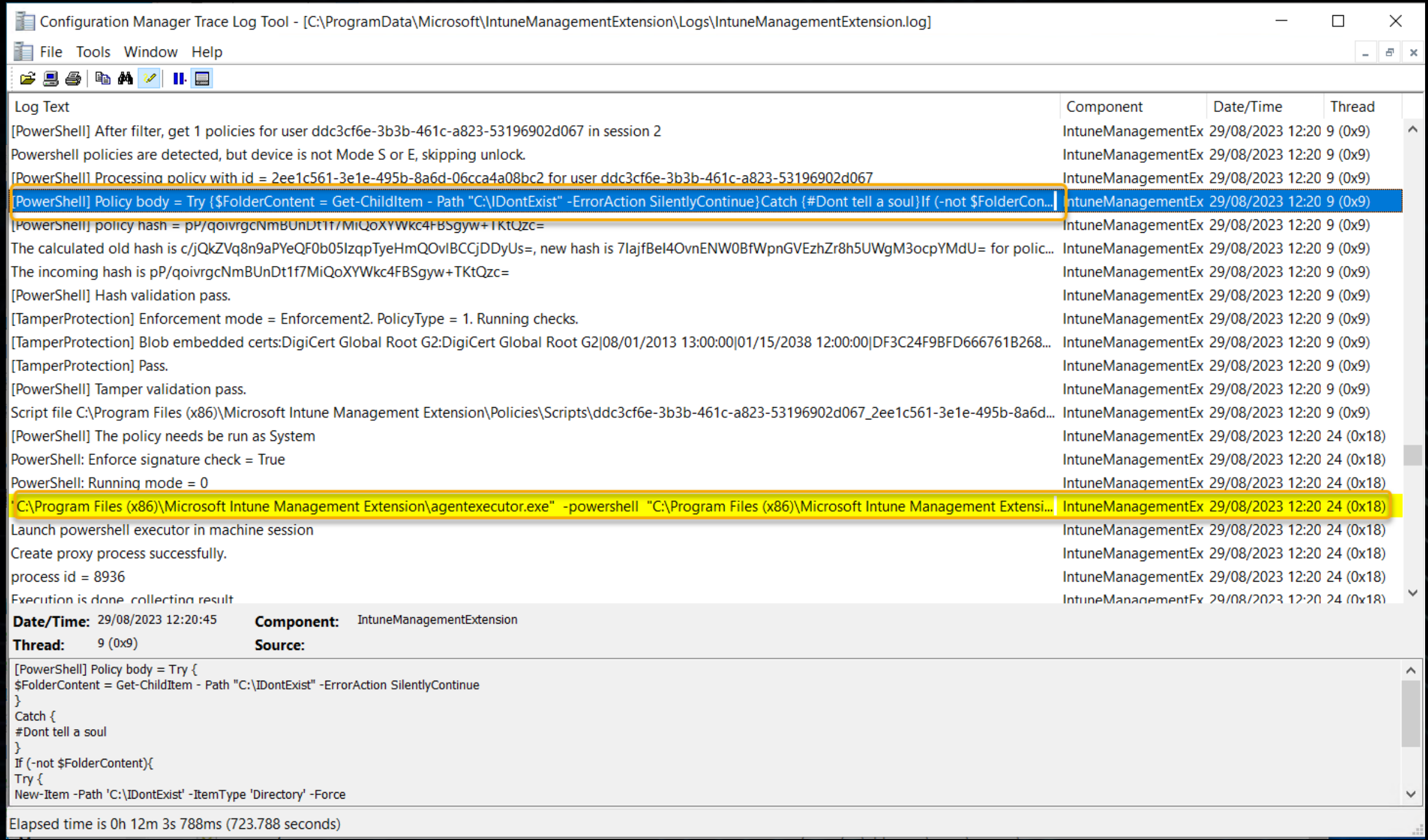
Drink Coffee

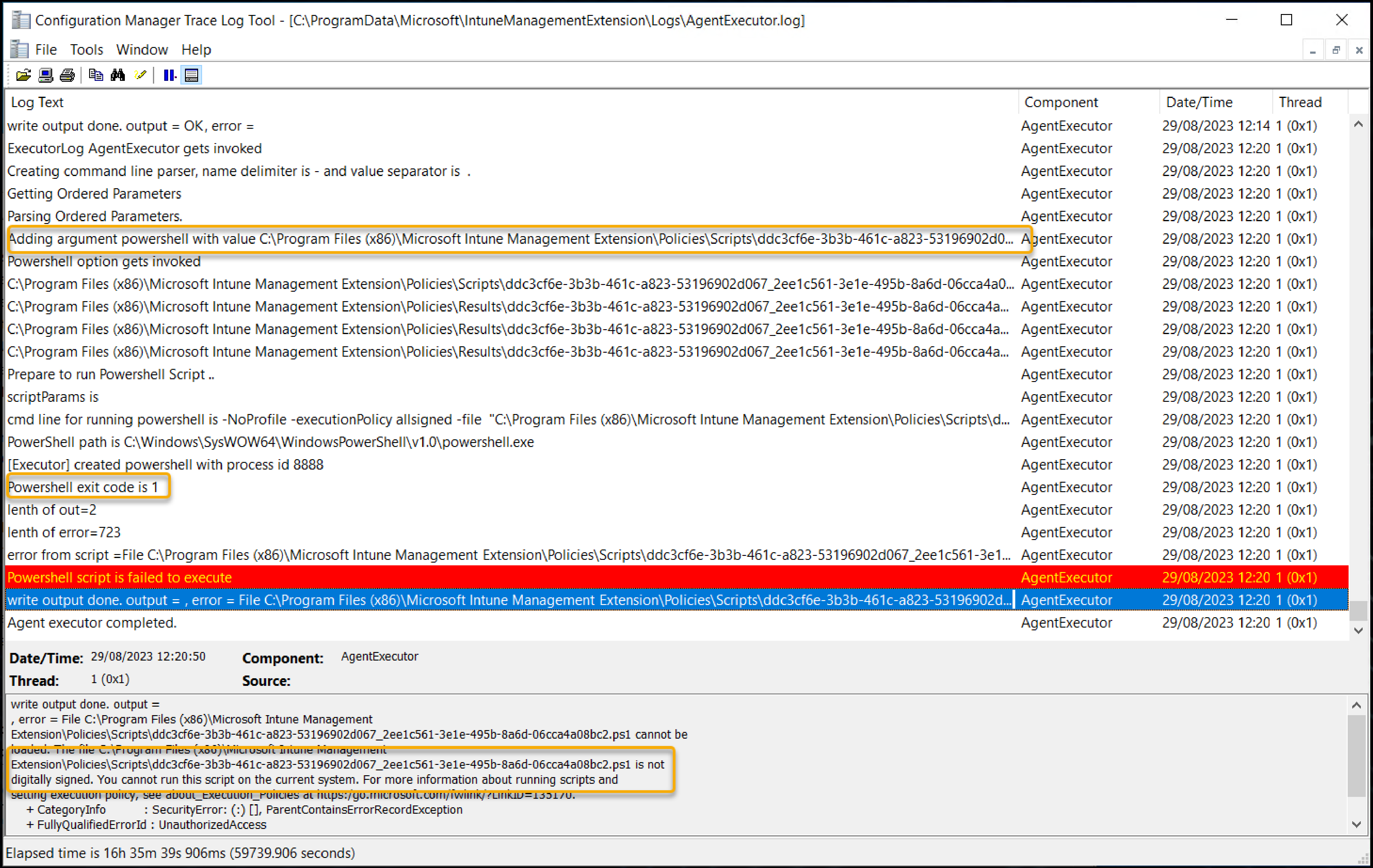
TUNEDIN MEETUPS

TunedIn:London

How does the IME process scripts?

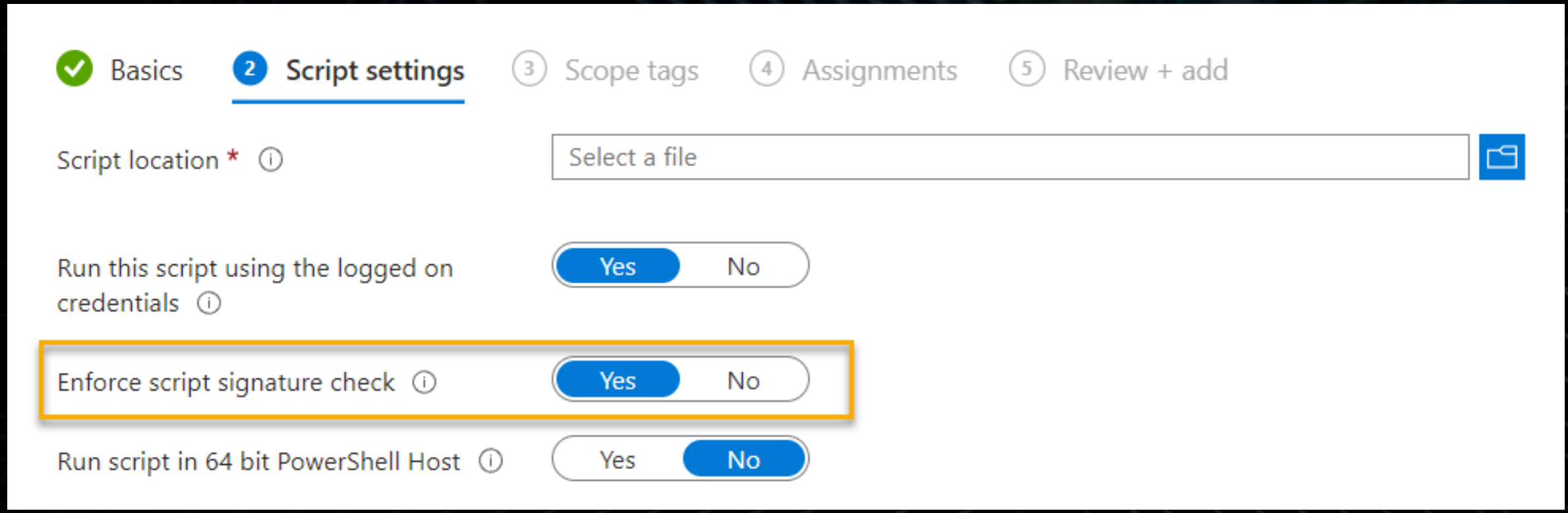



Configuration Manager Trace Log Tool - [C:\ProgramData\Microsoft\IntuneManagementExtension\Logs\IntuneManagementExtension.log]
File Tools Window Help
Log Text Component Date/Time Thread
[PowerShell] After filter, get 1 policies for user ddc3cf6e-3b3b-461c-a823-53196902d067 in session 2 IntuneManagementEx 29/08/2023 12:20 9 (0x9)
Powershell policies are detected, but device is not Mode S or E, skipping unlock. IntuneManagementEx 29/08/2023 12:20 9 (0x9)
[PowerShell] Processing policy with id = 2ee1c561-3e1e-495b-8a6d-06cca4a08bc2 for user ddc3cf6e-3b3b-461c-a823-53196902d067 IntuneManagementEx 29/08/2023 12:20 9 (0x9)
[PowerShell] Policy body = Try {$FolderContent = Get-ChildItem - Path "C:\IDontExist" -ErrorAction SilentlyContinue}Catch {#Dont tell a soul}If (-not $FolderCon... IntuneManagementEx 29/08/2023 12:20 9 (0x9)
[PowerShell] policy hash = pP/qoivrgcNmBUnDt1f7MiQoXYWkc4FBSgyw+TKtQzc= IntuneManagementEx 29/08/2023 12:20 9 (0x9)
The calculated old hash is c/jQkZVq8n9aPYeQF0b05IzqpTyeHmQOvIBCCjDDyUs=, new hash is 7IajfBeI4OvnENW0BfWpnGVEzhZr8h5UWgM3ocpYMdU= for polic... IntuneManagementEx 29/08/2023 12:20 9 (0x9)
The incoming hash is pP/qoivrgcNmBUnDt1f7MiQoXYWkc4FBSgyw+TKtQzc= IntuneManagementEx 29/08/2023 12:20 9 (0x9)
[PowerShell] Hash validation pass. IntuneManagementEx 29/08/2023 12:20 9 (0x9)
[TamperProtection] Enforcement mode = Enforcement2. PolicyType = 1. Running checks. IntuneManagementEx 29/08/2023 12:20 9 (0x9)
[TamperProtection] Blob embedded certs:DigiCert Global Root G2:DigiCert Global Root G2|08/01/2013 13:00:00|01/15/2038 12:00:00|DF3C24F9BFD666761B268... IntuneManagementEx 29/08/2023 12:20 9 (0x9)
[TamperProtection] Pass. IntuneManagementEx 29/08/2023 12:20 9 (0x9)
[PowerShell] Tamper validation pass. IntuneManagementEx 29/08/2023 12:20 9 (0x9)
Script file C:\Program Files (x86)\Microsoft Intune Management Extension\Policies\Scripts\ddc3cf6e-3b3b-461c-a823-53196902d067_2ee1c561-3e1e-495b-8a6d... IntuneManagementEx 29/08/2023 12:20 9 (0x9)
[PowerShell] The policy needs be run as System IntuneManagementEx 29/08/2023 12:20 24 (0x18)
PowerShell: Enforce signature check = True IntuneManagementEx 29/08/2023 12:20 24 (0x18)
PowerShell: Running mode = 0 IntuneManagementEx 29/08/2023 12:20 24 (0x18)
"C:\Program Files (x86)\Microsoft Intune Management Extension\agentexecutor.exe" -powershell "C:\Program Files (x86)\Microsoft Intune Management Extensi... IntuneManagementEx 29/08/2023 12:20 24 (0x18)
Launch powershell executor in machine session IntuneManagementEx 29/08/2023 12:20 24 (0x18)
Create proxy process successfully. IntuneManagementEx 29/08/2023 12:20 24 (0x18)
process id = 8936 IntuneManagementEx 29/08/2023 12:20 24 (0x18)
Execution is done, collecting result IntuneManagementEx 29/08/2023 12:20 24 (0x18)
Date/Time: 29/08/2023 12:20:45 Component: IntuneManagementExtension
Thread: 9 (0x9) Source:
[PowerShell] Policy body = Try {
$FolderContent = Get-ChildItem - Path "C:\IDontExist" -ErrorAction SilentlyContinue
}
Catch {
#Dont tell a soul
}
If (-not $FolderContent){
Try {
New-Item -Path 'C:\IDontExist' -ItemType 'Directory' -Force
Elapsed time is 0h 12m 3s 788ms (723.788 seconds)


Configuration Manager Trace Log Tool - [C:\ProgramData\Microsoft\IntuneManagementExtension\Logs\AgentExecutor.log]
File Tools Window Help
Log Text
Component
Date/Time
Thread
write output done. output = OK, error = | AgentExecutor | 29/08/2023 12:14 | 1 (0x1)
ExecutorLog AgentExecutor gets invoked | AgentExecutor | 29/08/2023 12:20 | 1 (0x1)
Creating command line parser, name delimiter is - and value separator is . | AgentExecutor | 29/08/2023 12:20 | 1 (0x1)
Getting Ordered Parameters | AgentExecutor | 29/08/2023 12:20 | 1 (0x1)
Parsing Ordered Parameters. | AgentExecutor | 29/08/2023 12:20 | 1 (0x1)
Adding argument powershell with value C:\Program Files (x86)\Microsoft Intune Management Extension\Policies\Scripts\ddc3cf6e-3b3b-461c-a823-53196902d0... | AgentExecutor | 29/08/2023 12:20 | 1 (0x1)
Powershell option gets invoked | AgentExecutor | 29/08/2023 12:20 | 1 (0x1)
C:\Program Files (x86)\Microsoft Intune Management Extension\Policies\Scripts\ddc3cf6e-3b3b-461c-a823-53196902d067_2ee1c561-3e1e-495b-8a6d-06cca4a0... | AgentExecutor | 29/08/2023 12:20 | 1 (0x1)
C:\Program Files (x86)\Microsoft Intune Management Extension\Policies\Results\ddc3cf6e-3b3b-461c-a823-53196902d067_2ee1c561-3e1e-495b-8a6d-06cca4a... | AgentExecutor | 29/08/2023 12:20 | 1 (0x1)
C:\Program Files (x86)\Microsoft Intune Management Extension\Policies\Results\ddc3cf6e-3b3b-461c-a823-53196902d067_2ee1c561-3e1e-495b-8a6d-06cca4a... | AgentExecutor | 29/08/2023 12:20 | 1 (0x1)
C:\Program Files (x86)\Microsoft Intune Management Extension\Policies\Results\ddc3cf6e-3b3b-461c-a823-53196902d067_2ee1c561-3e1e-495b-8a6d-06cca4a... | AgentExecutor | 29/08/2023 12:20 | 1 (0x1)
Prepare to run Powershell Script .. | AgentExecutor | 29/08/2023 12:20 | 1 (0x1)
scriptParams is | AgentExecutor | 29/08/2023 12:20 | 1 (0x1)
cmd line for running powershell is -NoProfile -executionPolicy allsigned -file "C:\Program Files (x86)\Microsoft Intune Management Extension\Policies\Scripts\d... | AgentExecutor | 29/08/2023 12:20 | 1 (0x1)
PowerShell path is C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe | AgentExecutor | 29/08/2023 12:20 | 1 (0x1)
[Executor] created powershell with process id 8888 | AgentExecutor | 29/08/2023 12:20 | 1 (0x1)
Powershell exit code is 1 | AgentExecutor | 29/08/2023 12:20 | 1 (0x1)
lenth of out=2 | AgentExecutor | 29/08/2023 12:20 | 1 (0x1)
lenth of error=723 | AgentExecutor | 29/08/2023 12:20 | 1 (0x1)
error from script =File C:\Program Files (x86)\Microsoft Intune Management Extension\Policies\Scripts\ddc3cf6e-3b3b-461c-a823-53196902d067_2ee1c561-3e1... | AgentExecutor | 29/08/2023 12:20 | 1 (0x1)
Powershell script is failed to execute | AgentExecutor | 29/08/2023 12:20 | 1 (0x1)
write output done. output = , error = File C:\Program Files (x86)\Microsoft Intune Management Extension\Policies\Scripts\ddc3cf6e-3b3b-461c-a823-53196902d... | AgentExecutor | 29/08/2023 12:20 | 1 (0x1)
Agent executor completed. | AgentExecutor | 29/08/2023 12:20 | 1 (0x1)
Date/Time: 29/08/2023 12:20:50
Component: AgentExecutor
Thread: 1 (0x1)
Source:
write output done. output =
, error = File C:\Program Files (x86)\Microsoft Intune Management
Extension\Policies\Scripts\ddc3cf6e-3b3b-461c-a823-53196902d067_2ee1c561-3e1e-495b-8a6d-06cca4a08bc2.ps1 cannot be
Extension\Policies\Scripts\ddc3cf6e-3b3b-461c-a823-53196902d067_2ee1c561-3e1e-495b-8a6d-06cca4a08bc2.ps1 is not
digitally signed. You cannot run this script on the current system. For more information about running scripts and
+ CategoryInfo : SecurityError: (:) [], ParentContainsErrorRecordException
+ FullyQualifiedErrorId : UnauthorizedAccess
Elapsed time is 16h 35m 39s 906ms (59739.906 seconds)

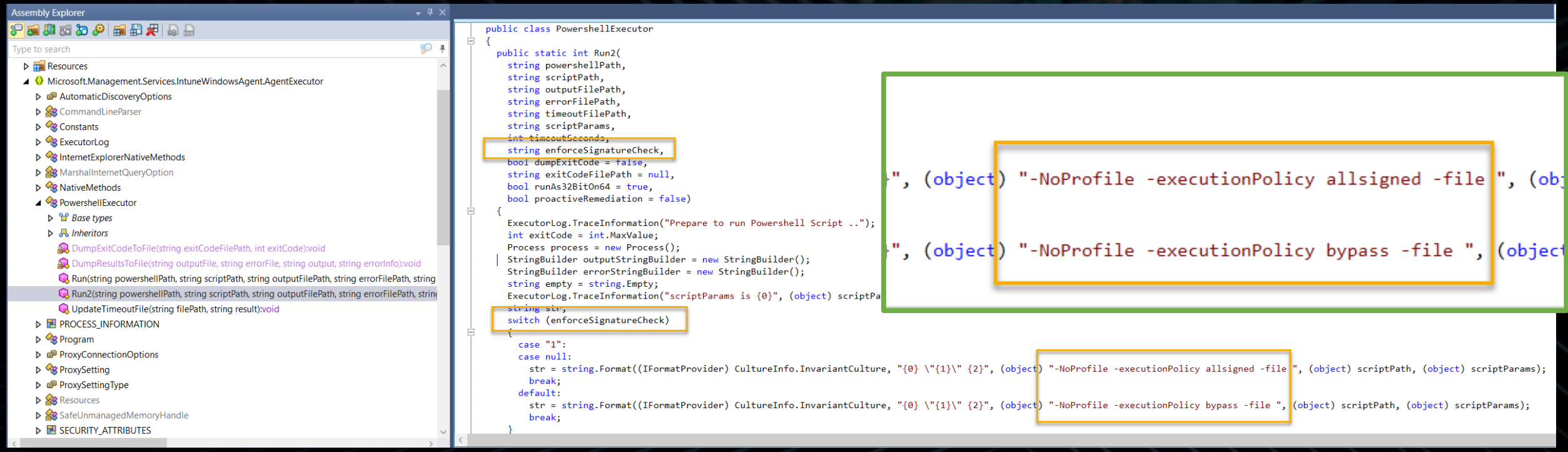
How does the IME process scripts?

Enforce script signature is now enabled by default



How does the IME process scripts?

Agent Executor invokes PowerShell to run platform scripts and sets the PowerShell policy to allsigned or bypass according to the policy received



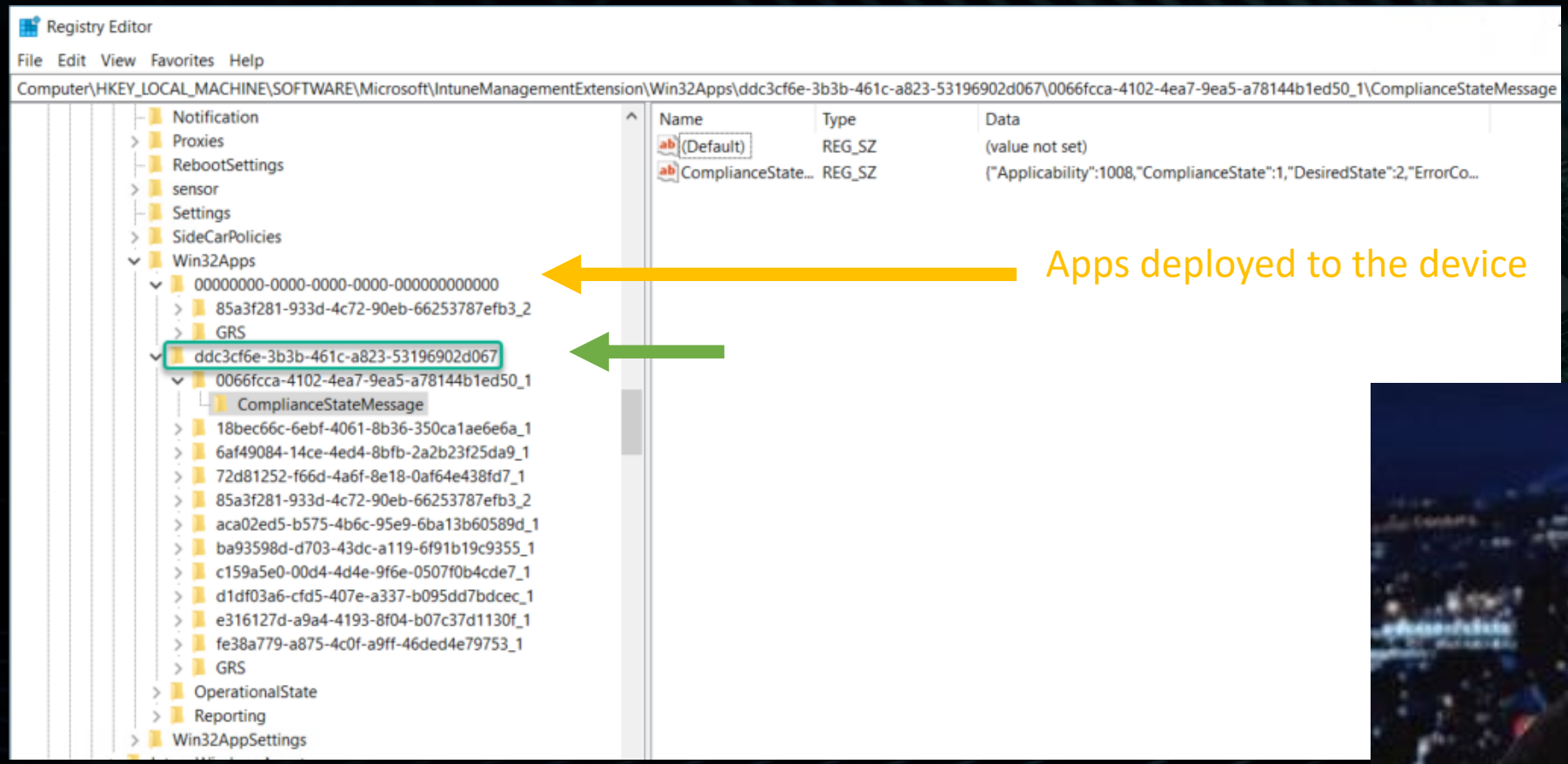
How does the IME process scripts?

- Platform scripts are executed before Win32 apps
- Platform scripts will try 3 times. Will not retry unless script is modified
- IME deployed even if Client Apps workload is not set to Intune
- Script size must be less than 200kb
- Platform scripts time out after 30 minutes

Tip: Use PSEXEC to simulate a script running as SYSTEM

Digging into win32 app policy event state messages

Win32 app policy events are stored in the registry



Apps deployed to the device

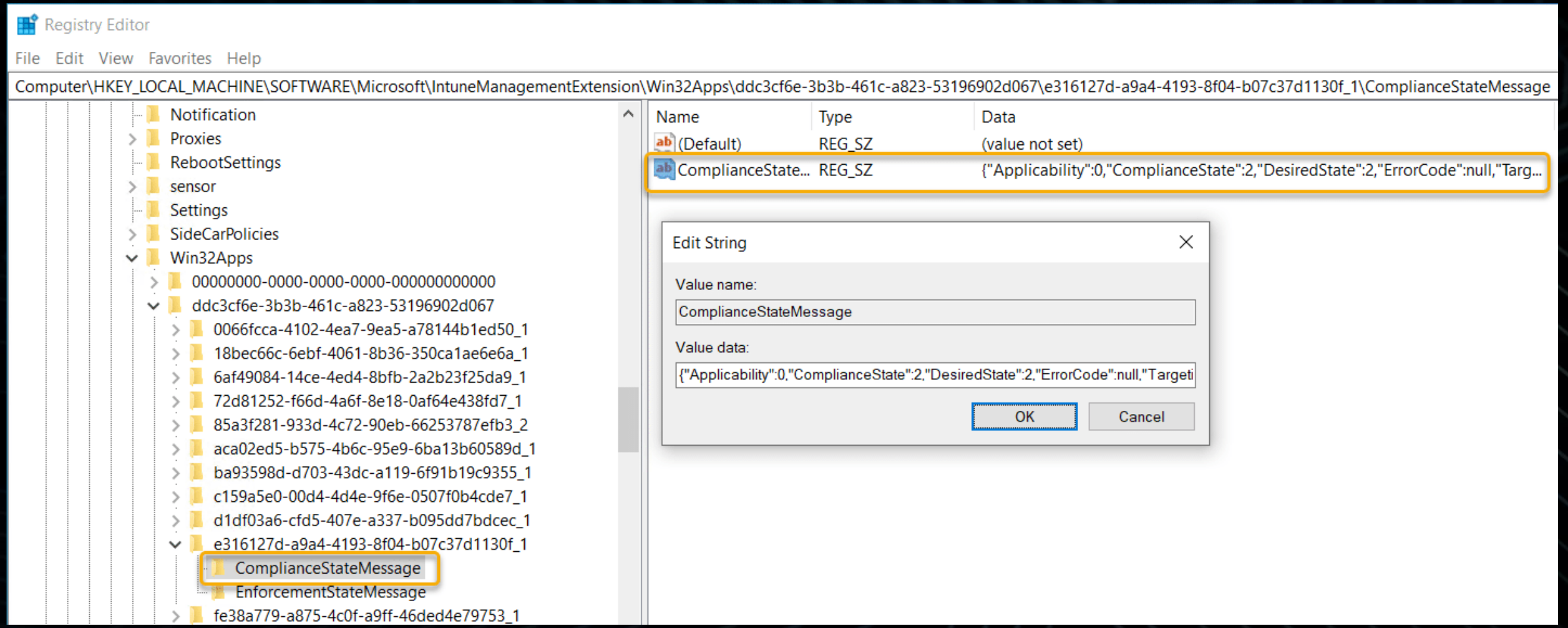
HLKM:SOFTWARE\Microsoft\IntuneManagementExtension\Win32Apps

TUNEDIN MEETUPS

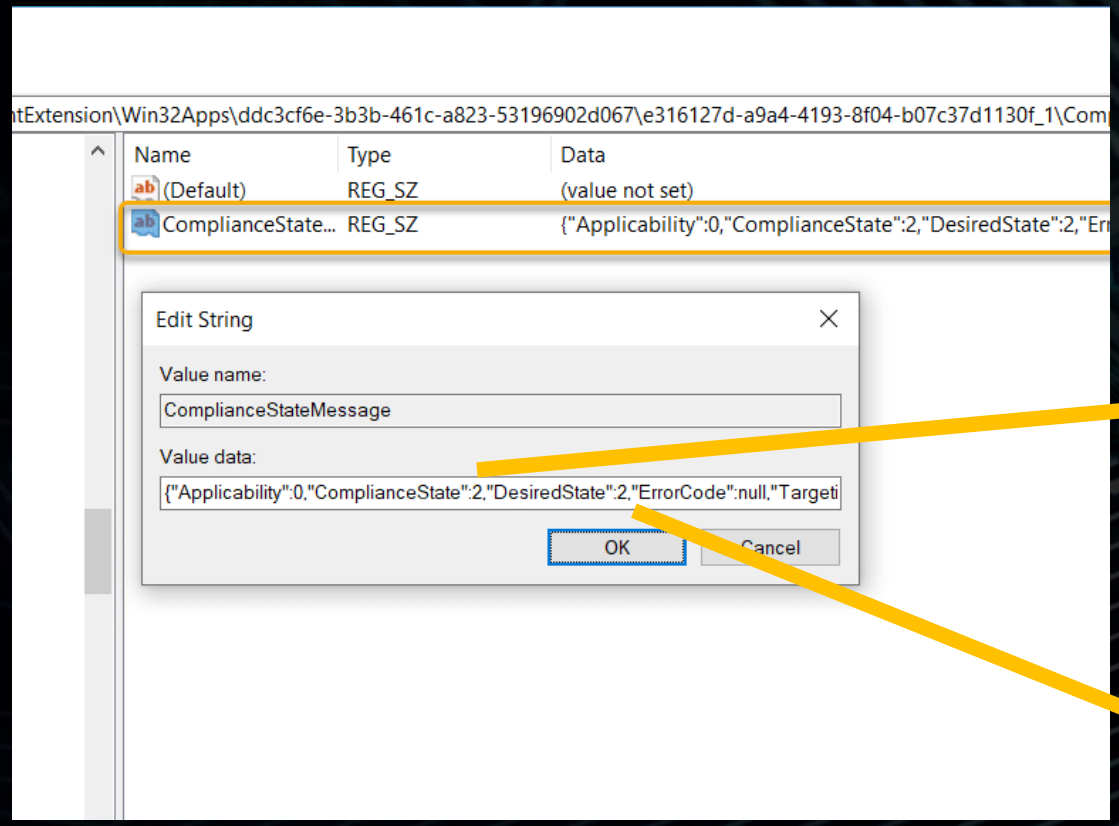
TunedIn:London

Digging into win32 app policy event state messages

Win32 app policy events are stored in the registry



Digging into win32 app policy event state messages



App deployment Type (Intent):

Values	Description
1	Available
3	Required
4	Uninstall

Compliance State:

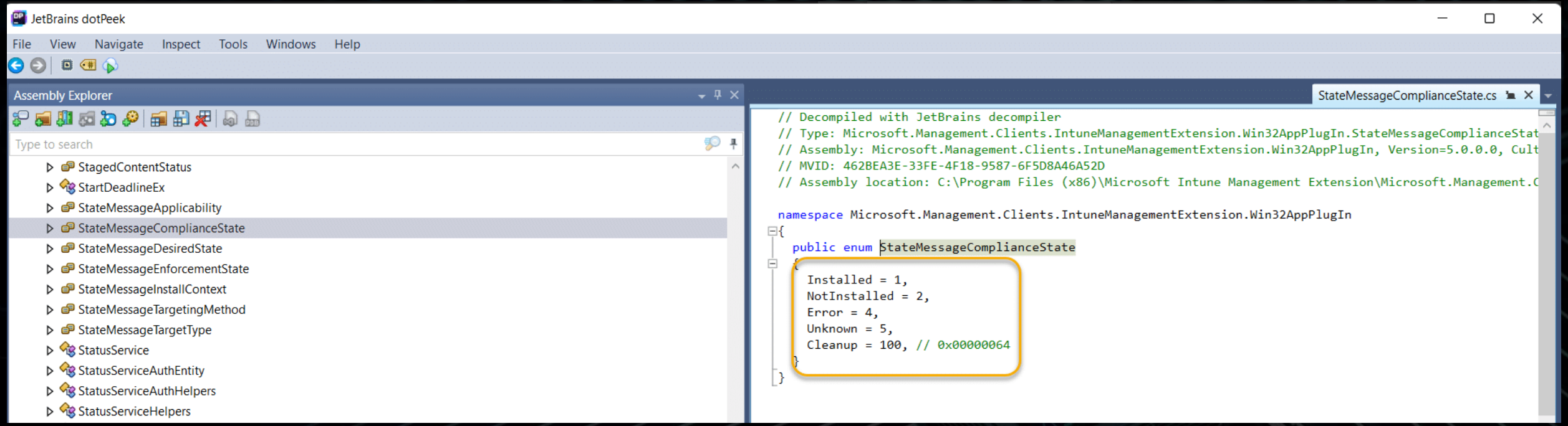
Values	Description
0	Unknown
1	Compliant
2	Not compliant
3	Conflict (Not applicable for app deployment)
4	Error

Desired State:

Values	Description
0	None
1	NotPresent
2	Present
3	Unknown
4	Available

Digging into win32 app policy event state messages

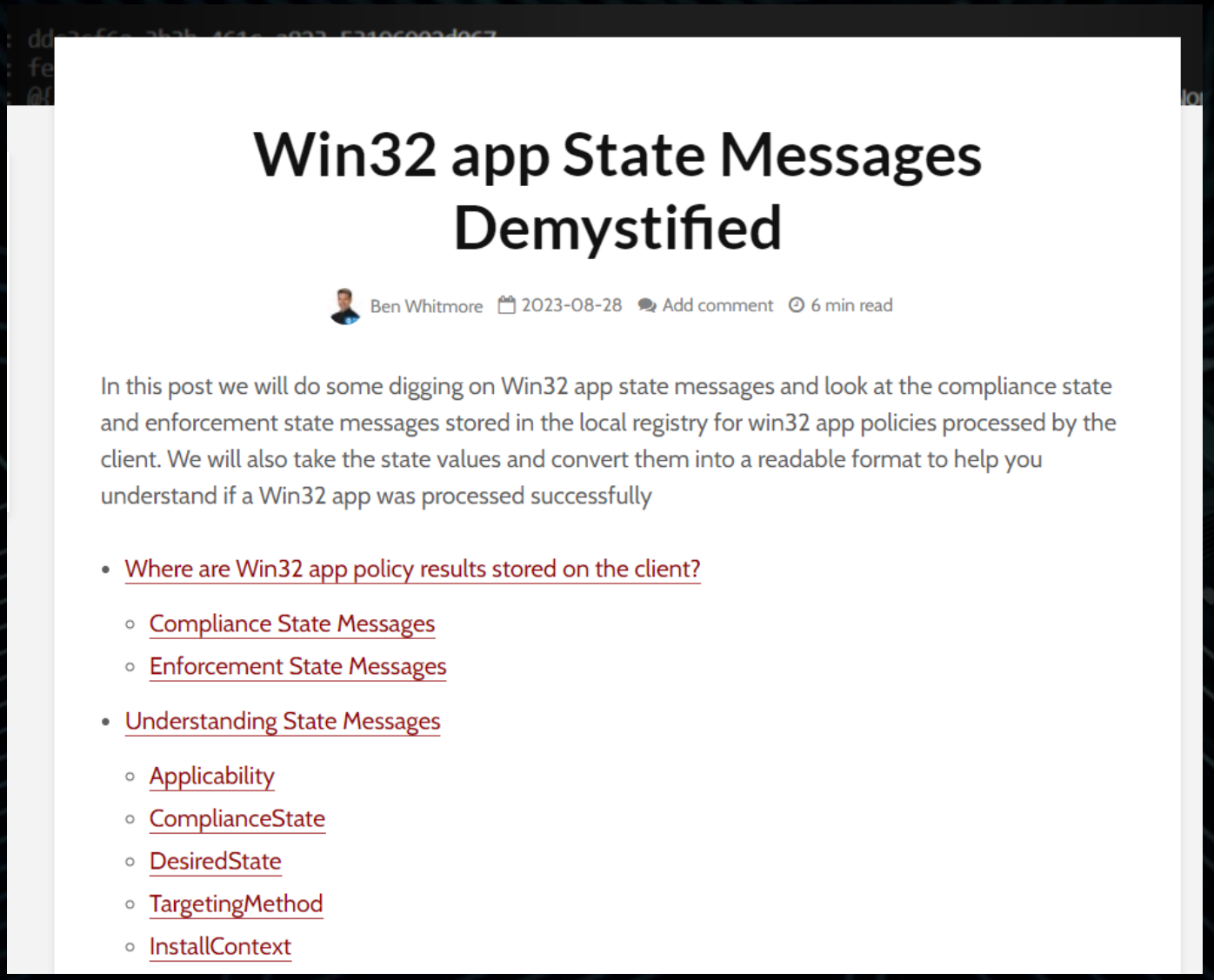
State Message Magic




```
$stateMessageComplianceState = @{
    1   = "Installed"
    2   = "NotInstalled"
    4   = "Error"
    5   = "Unknown"
    100 = "Cleanup"
}
```

Digging into win32 app policy event state messages

State Message Magic



Win32 app State Messages Demystified

Ben Whitmore 2023-08-28 Add comment 6 min read

In this post we will do some digging on Win32 app state messages and look at the compliance state and enforcement state messages stored in the local registry for win32 app policies processed by the client. We will also take the state values and convert them into a readable format to help you understand if a Win32 app was processed successfully

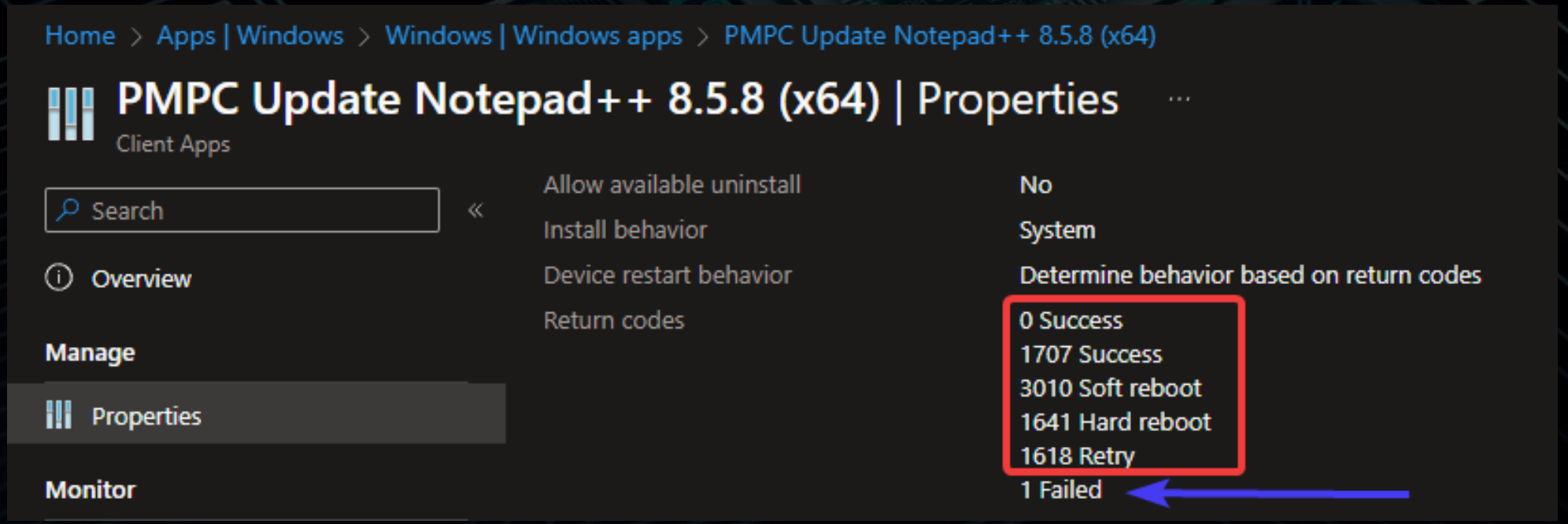
- Where are Win32 app policy results stored on the client?
  - Compliance State Messages
  - Enforcement State Messages
- Understanding State Messages
  - Applicability
  - ComplianceState
  - DesiredState
  - TargetingMethod
  - InstallContext

Digging into win32 app policy event state messages

How the IME handles failures and retries

Failed apps retry 3 times every 5 minutes and then every 24 hours if they are required and the installer exits with a known retry code

Failed apps retry every 24 hours if they are required, and the installer exits with a failure or unknown exit code



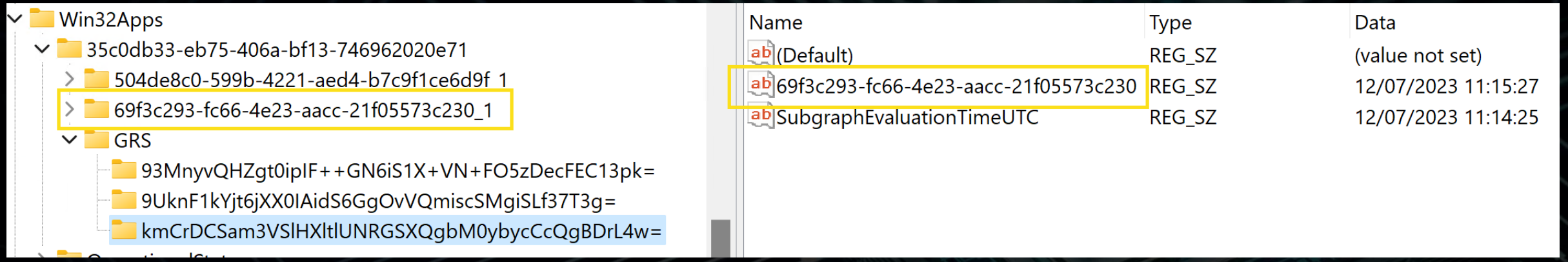
How the IME handles failures and retries

- When a new app is assigned, it is evaluated
- Time Column is local | Time in the lower row is UTC
- Two components:
  - Reevaluation Schedule Manager (subgraph) – expires and is evaluated every 8 hours
  - Global Retry Schedule (GRS) - expires every 24 hours; it controls when a failed app install is retried

ⓘ	IntuneManagementEx…	Get policies = [{"Id":"5cfd8e28-25a4-44f1-86d0-a0d107dd0a5d","Name" "Update for Notepad++ 8.5.8 (x64)","Version":1,"Intent":3,"TargetType":2,"AppApplicabilityStateDueToAssgi	10/20/2023 3:47:51.916
ⓘ	IntuneManagementEx…	[Win32App][V3Processor] Processing 1 subgraphs.	10/20/2023 3:47:51.947
⚠	IntuneManagementEx…	[Win32App][ReevaluationScheduleManager] Did not find any previous reevaluation check-in time.	10/20/2023 3:47:52.009
⚠	IntuneManagementEx…	[Win32App][ReevaluationScheduleManager] Did not find any previous subgraph reevaluation time at key RaBK4ZlzalZCl9uOMmgh0I+TgBdDX3vPGIcPpFfitQI=.	10/20/2023 3:47:52.009
ⓘ	IntuneManagementEx…	[Win32App][ReportingManager] App with id: 5cfd8e28-25a4-44f1-86d0-a0d107dd0a5d and prior AppAuthority: V2 could not be loaded from store. Reporting state initialized with in:	10/20/2023 3:47:57.916
ⓘ	IntuneManagementEx…	[Win32App][ReportingManager] Not sending status update for user with id: 807efa31-c262-4f9c-9372-c22189e33964 and app: 5cfd8e28-25a4-44f1-86d0-a0d107dd0a5d because there is	10/20/2023 3:47:57.931
ⓘ	IntuneManagementEx…	[Win32App][ReportingManager] Real time status is not reportable for user: 807efa31-c262-4f9c-9372-c22189e33964 and app: 5cfd8e28-25a4-44f1-86d0-a0d107dd0a5d after switch to '	10/20/2023 3:47:57.931
ⓘ	IntuneManagementEx…	[Win32App][ReportingCacheManager] No full sync time found for user: 807efa31-c262-4f9c-9372-c22189e33964, which is treated as expired. All targeted apps will be evaluated an	10/20/2023 3:47:57.947
ⓘ	IntuneManagementEx…	[Win32App][V3Processor] Processing subgraph with app ids: 5cfd8e28-25a4-44f1-86d0-a0d107dd0a5d	10/20/2023 3:47:57.947
ⓘ	IntuneManagementEx…	[Win32App][GRSManager] Reading GRS values from storage path: 807efa31-c262-4f9c-9372-c22189e33964\GRS\RaBK4ZlzalZCl9uOMmgh0I+TgBdDX3vPGIcPpFfitQI=\.	10/20/2023 3:47:57.963
ⓘ	IntuneManagementEx…	[Win32App][GRSManager] App with id: 5cfd8e28-25a4-44f1-86d0-a0d107dd0a5d has no recorded GRS value which will be treated as expired.Hash = RaBK4ZlzalZCl9uOMmgh0I+TgBdDX3vPGI	10/20/2023 3:47:57.963
ⓘ	IntuneManagementEx…	[Win32App][ReevaluationScheduleManager] Subgraph has no recorded reevaluation info which will be treated as expired.Hash = RaBK4ZlzalZCl9uOMmgh0I+TgBdDX3vPGIcPpFfitQI=	10/20/2023 3:47:57.963
ⓘ	IntuneManagementEx…	[Win32App][ReevaluationScheduleManager] Setting subgraph reevaluation time with value: 10/20/2023 9:47:57 AM for subgraph with hash RaBK4ZlzalZCl9uOMmgh0I+TgBdDX3vPGIcPpFfit	10/20/2023 3:47:57.963

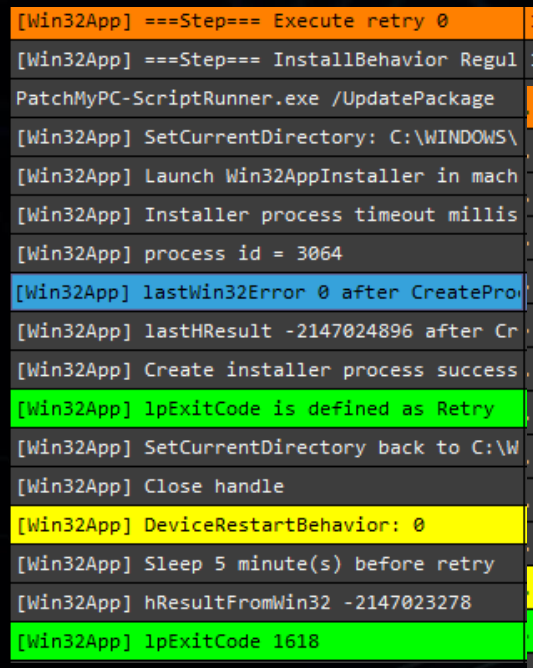
How the IME handles failures and retries

Policy is processed and GRS key is set

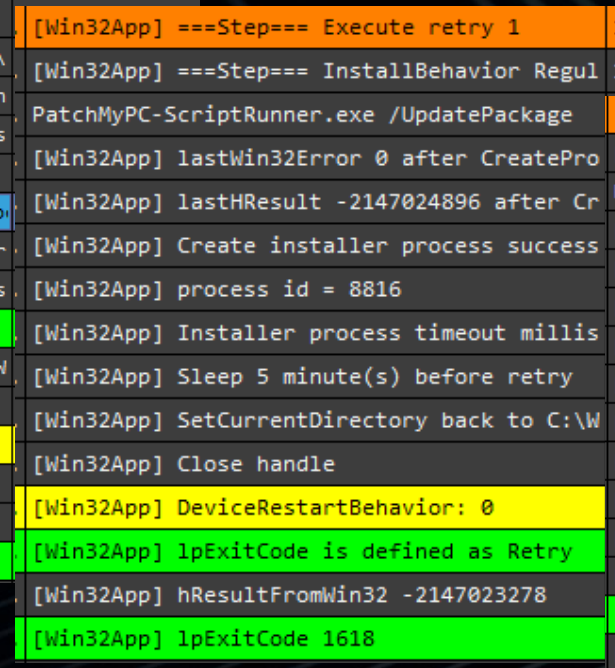


How the IME handles failures and retries

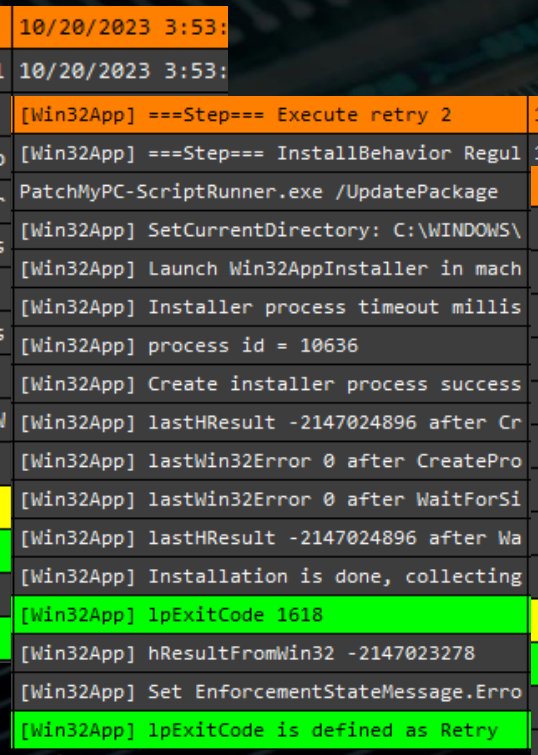
In this example, the app installer terminated with a known retry code and tried 3 more times to attempt the installation (5 minute intervals)



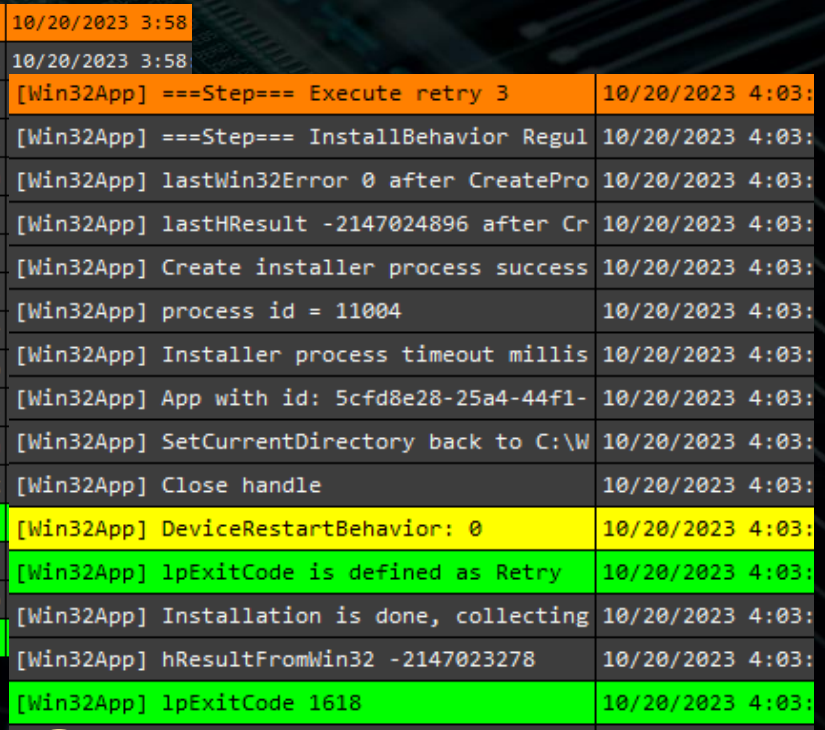
Initial Install



1



2



3

How the IME handles failures and retries

After 3 failed retries, the policy will be tried again in 24 hours time when the GRS value expires

Get policies = [{"Id":"5cfd8e28-25a4-44f1-86d0-a0d107dd0a5d","Name":"Update for Notepad++ 8.5.8 (x64)","Version":1,"Intent":3,"TargetType":2,"AppApplicabilityStateDueToAssgi	10/21/2023 4:57:35.
[Win32App] Got 1 Win32App(s) for user 807efa31-c262-4f9c-9372-c22189e33964 in session 2	10/21/2023 4:57:35.
[Win32App] Got result with session id 60f6aa80-8671-428d-8bc4-bde8f8a3c186	10/21/2023 4:57:35.
[Win32App] espSupportForSupersedenceEnabled : False, v2AppPrococesorDisabled : True	10/21/2023 4:57:35.
[Win32App][DetectionActionHandler] Detection running for policy with id: 5cfd8e28-25a4-44f1-86d0-a0d107dd0a5d.	10/21/2023 4:57:35.
"C:\Program Files (x86)\Microsoft Intune Management Extension\agentexecutor.exe" -powershellDetection "C:\Program Files (x86)\Microsoft Intune Management Extension\Content	10/21/2023 4:57:35.
[Win32App] Detection script file C:\Program Files (x86)\Microsoft Intune Management Extension\Content\DetectionScripts\5cfd8e28-25a4-44f1-86d0-a0d107dd0a5d_1.ps1 is saved.	10/21/2023 4:57:35.
[Win32App] SideCarScriptDetectionManager create files for outputs	10/21/2023 4:57:35.
[Win32App] Start detectionManager SideCarScriptDetectionManager	10/21/2023 4:57:35.
[Win32App] DetectionType 3	10/21/2023 4:57:35.
[Win32App] ProcessDetectionRules Parsing InstallEx...	10/21/2023 4:57:35.
[Win32App] ProcessDetectionRules starts	10/21/2023 4:57:35.
[Win32App][DetectionActionHandler] Handler invoked for 1 app detections.	10/21/2023 4:57:35.
[Win32App][V3Processor] Processing subgraph with app ids: 5cfd8e28-25a4-44f1-86d0-a0d107dd0a5d	10/21/2023 4:57:35.
[Win32App][ReevaluationScheduleManager] Setting subgraph reevaluation time with value: 10/21/2023 10:57:35 AM for subgraph with hash RaBK4ZlzalZCl9uOMmgh0I+TgBdDX3vPGIcPpFfi	10/21/2023 4:57:35.
[Win32App][ReevaluationScheduleManager] Subgraph reevaluation interval is expired.Has = RaBK4ZlzalZCl9uOMmgh0I+TgBdDX3vPGIcPpFfitQI=	10/21/2023 4:57:35.
[Win32App][GRSManager] App with id: 5cfd8e28-25a4-44f1-86d0-a0d107dd0a5d is expired Hash = RaBK4ZlzalZCl9uOMmgh0I+TgBdDX3vPGIcPpFfitQI=GRSTimeUTC = 10/20/2023 9:48:16 AM	10/21/2023 4:57:35.
[Win32App][GRSManager] Found GRS value: 10/20/2023 09:48:16 at key 807efa31-c262-4f9c-9372-c22189e33964\GRS\RaBK4ZlzalZCl9uOMmgh0I+TgBdDX3vPGIcPpFfitQI=\5cfd8e28-25a4-44f1-8	10/21/2023 4:57:35.
[Win32App][GRSManager] Reading GRS values from storage path: 807efa31-c262-4f9c-9372-c22189e33964\GRS\RaBK4ZlzalZCl9uOMmgh0I+TgBdDX3vPGIcPpFfitQI=\.	10/21/2023 4:57:35.

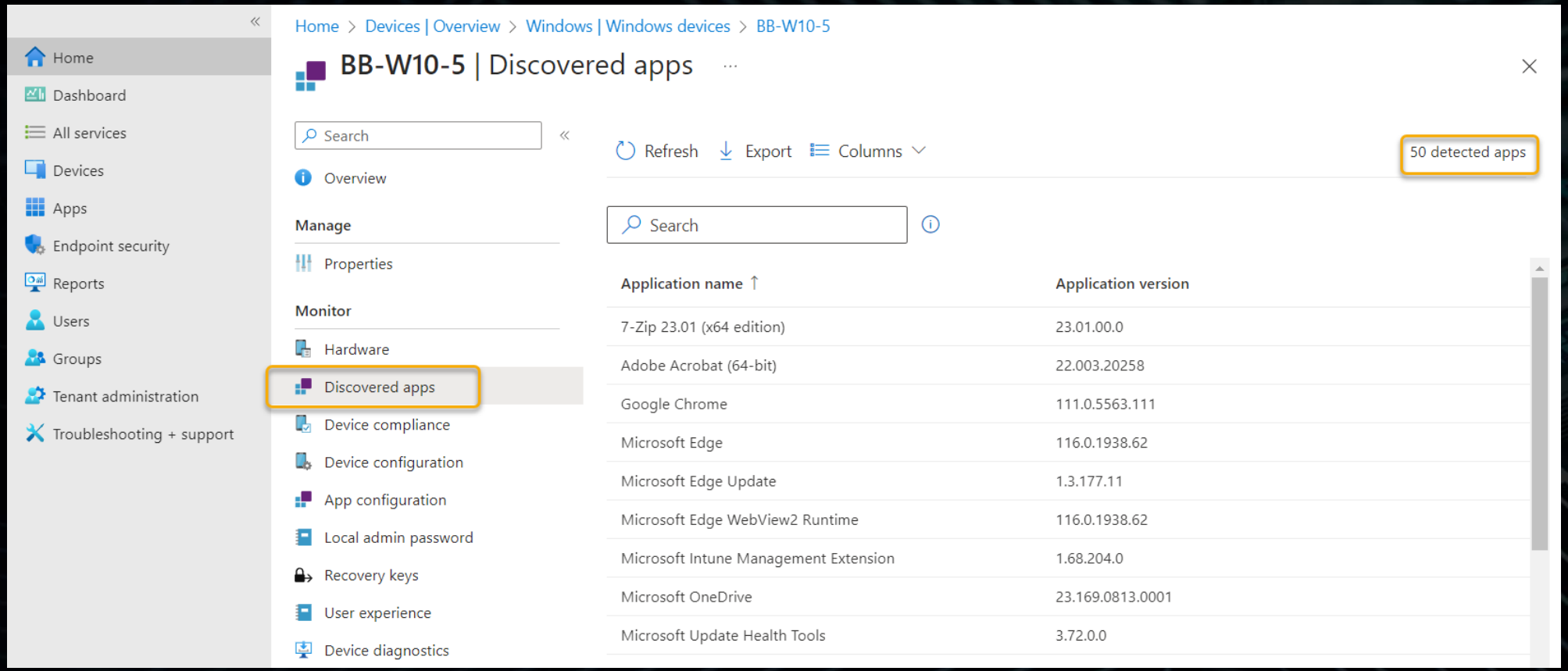
How the IME handles failures and retries

Summary

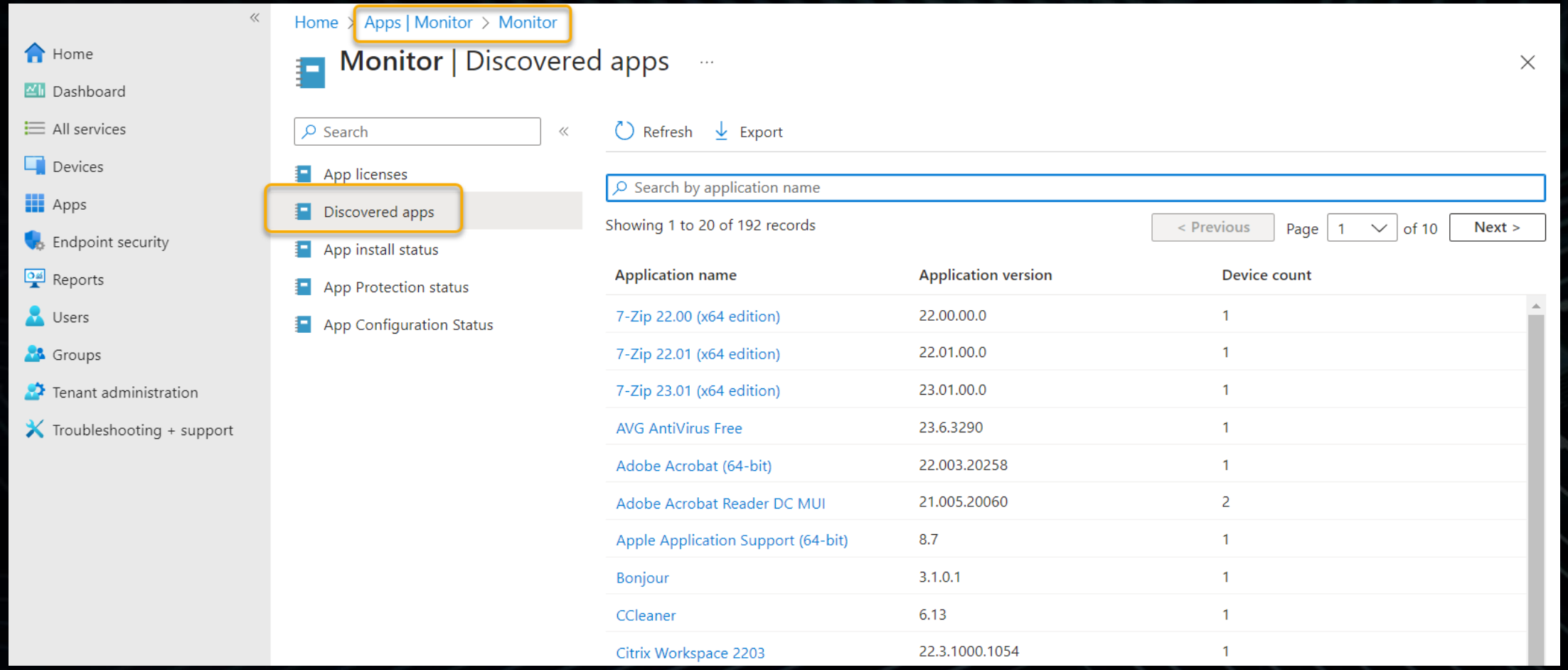
1. If the install fails, does the exit code indicate “Retry”? If so, retry 3 more times every 5 minutes
2. If the installation is failed (still), add the app to the GRS
3. Evaluate a sub graph every 8 hours to check when 24 hours have passed since the app was added to GRS
4. After 24 hours, retry the installation. If failed, update GRS check-in time value
5. Repeat forever until successful

Inventory

The IME will inventory installed applications.

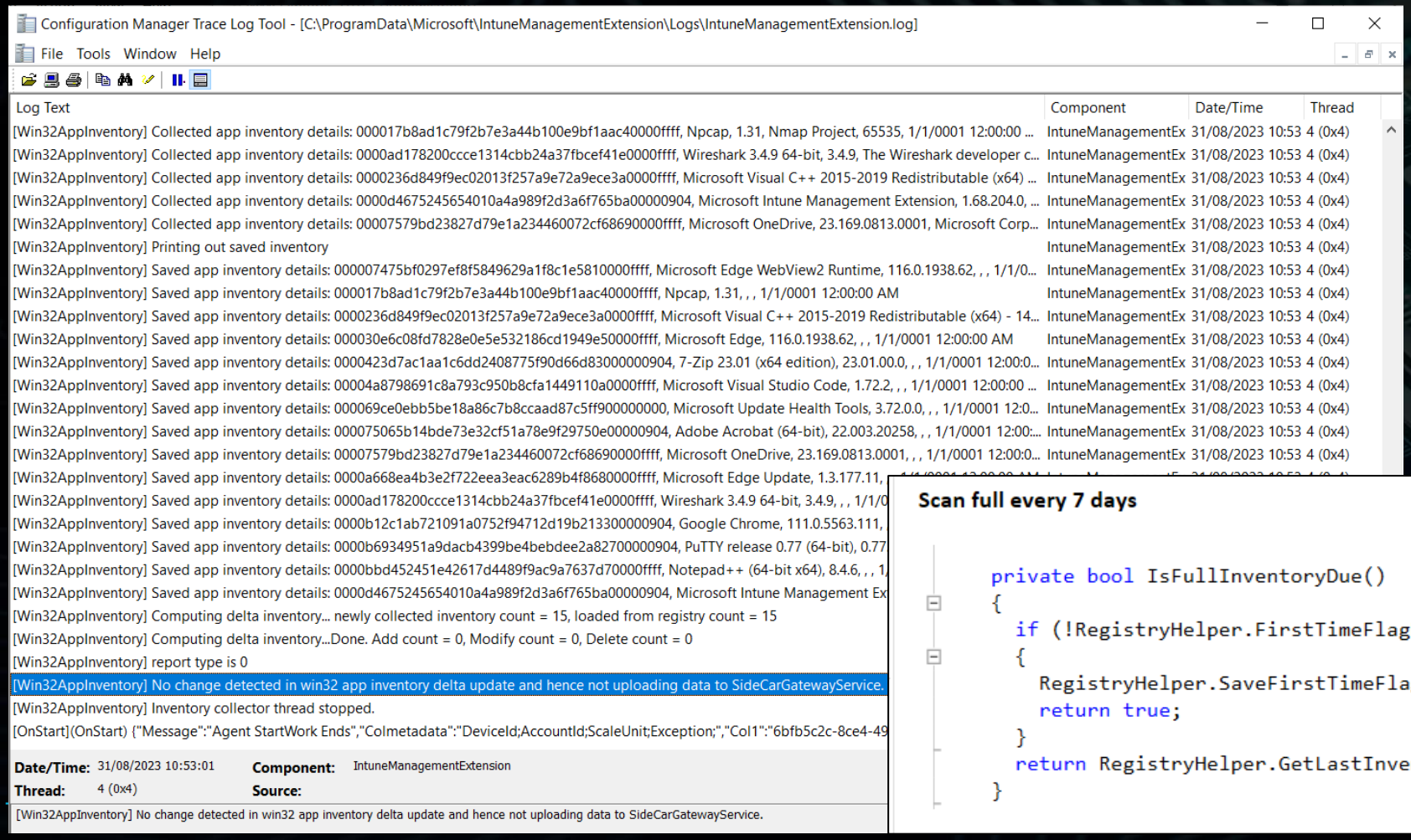


Inventory


Home
Dashboard
All services
Devices
Apps
Endpoint security
Reports
Users
Groups
Tenant administration
Troubleshooting + support
Home > Apps | Monitor > Monitor
Monitor | Discovered apps
Search
App licenses
Discovered apps
App install status
App Protection status
App Configuration Status
Refresh
Export
Search by application name
Showing 1 to 20 of 192 records
< Previous
Page 1 of 10
Next >
Application name
Application version
Device count
7-Zip 22.00 (x64 edition) 22.00.00.0 1
7-Zip 22.01 (x64 edition) 22.01.00.0 1
7-Zip 23.01 (x64 edition) 23.01.00.0 1
AVG AntiVirus Free 23.6.3290 1
Adobe Acrobat (64-bit) 22.003.20258 1
Adobe Acrobat Reader DC MUI 21.005.20060 2
Apple Application Support (64-bit) 8.7 1
Bonjour 3.1.0.1 1
CCleaner 6.13 1
Citrix Workspace 2203 22.3.1000.1054 1

Inventory

The IME performs a delta inventory every 24hrs and/or when the IME service starts. A full inventory is taken every 7 days and the first time the IME is installed



Remember .NET disassemblers!

Scan full every 7 days

```
private bool IsFullInventoryDue()
{
  if (!RegistryHelper.FirstTimeFlagExists())
  {
    RegistryHelper.SaveFirstTimeFlag();
    return true;
  }
  return RegistryHelper.GetLastInventoryFullSyncTimeUtc() < DateTime.UtcNow.AddDays(-7.0);
}
```

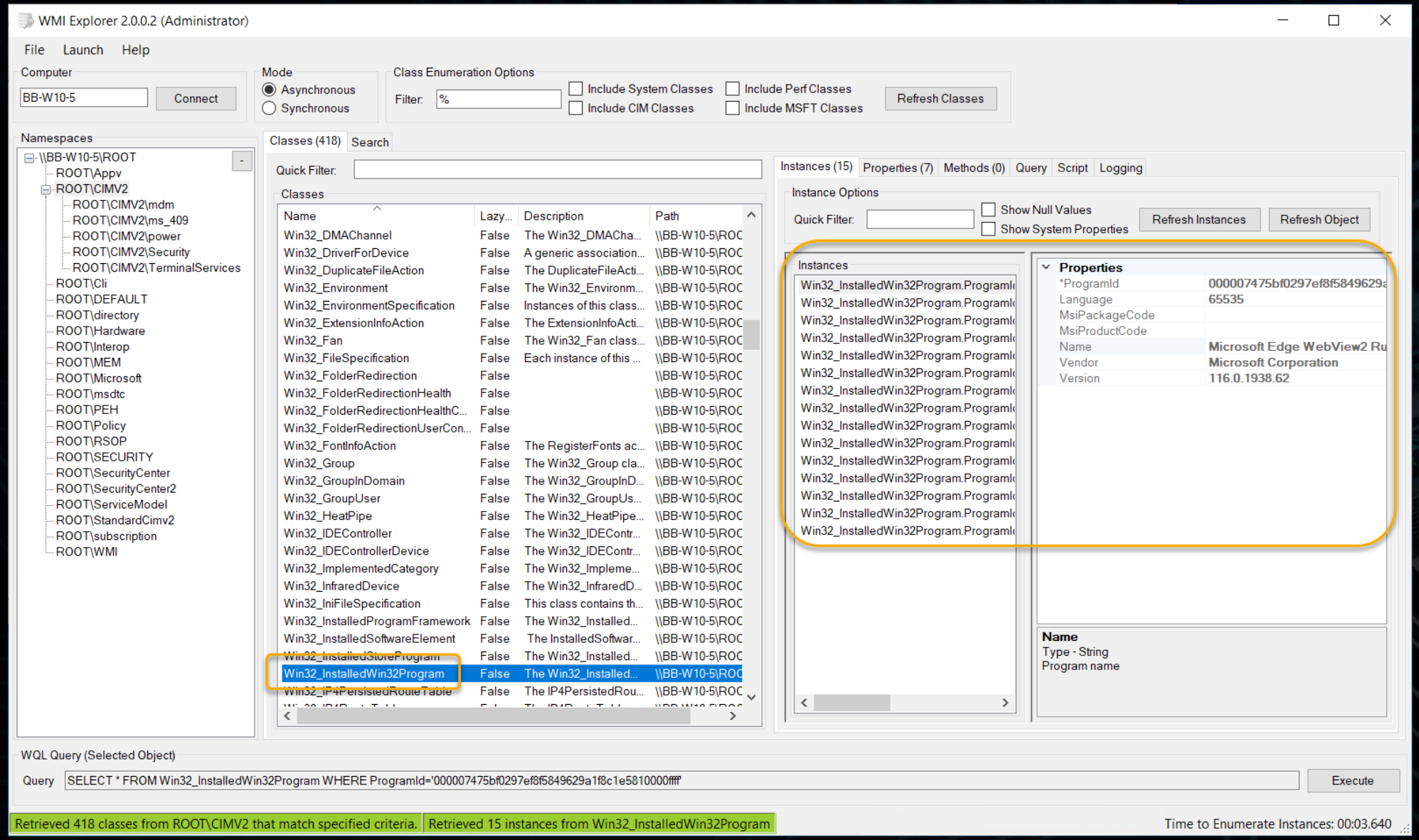
Inventory

The IME uses a specific WMI class to query installed software

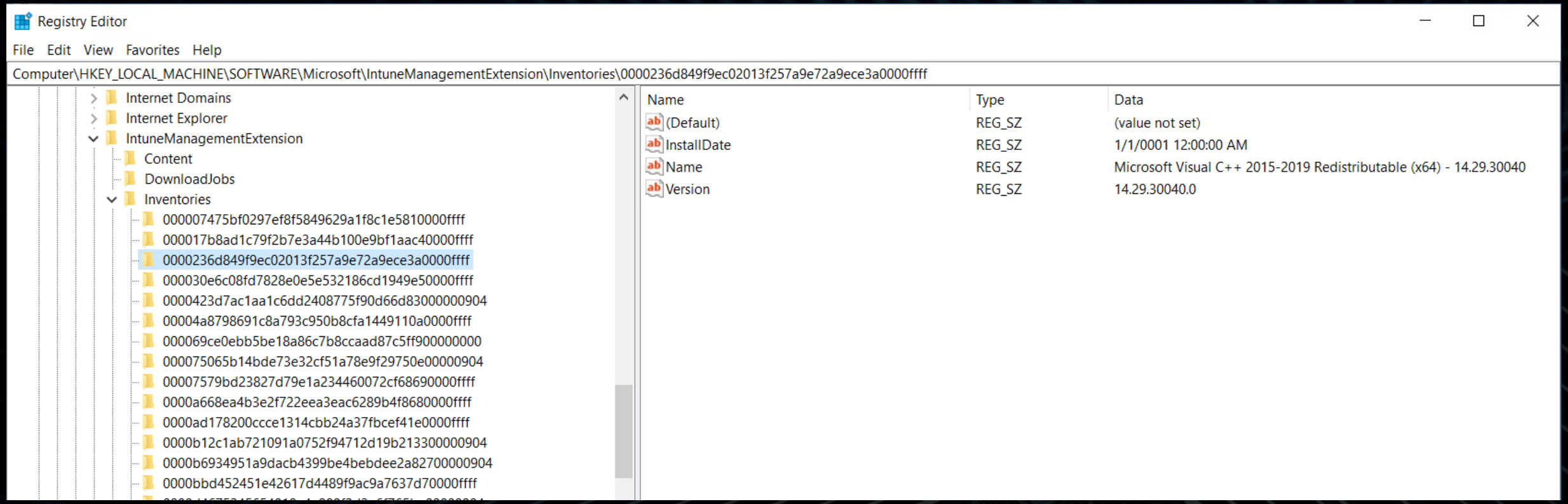
```
    {
      List<WindowsApplication> savedInventory = RegistryHelper.LoadApplicationInventoryFromRegistr
      reportToSend.ApplicationInventory = this.ComputeDeltaInventoryList(windowsApplicationList, s
      reportToSend.InventoryReportType = 0;
    }
    Win32AppInventoryLog.TraceInformation(string.Format("[Win32AppInventory] report type is {0}",
    if (reportToSend.InventoryReportType == 1 || reportToSend.InventoryReportType == 0 && ((IEnume
      this.uploader.UploadInventory(reportToSend, sessionId);
    else
      Win32AppInventoryLog.TraceInformation("[Win32AppInventory] No change detected in win32 app i
  }

  public IEnumerable<WindowsApplication> GetWin32AppInventoryExpanded()
  {
    string queryString = "select * from Win32_InstalledWin32Program";
    List<WindowsApplication> inventoryExpanded = new List<WindowsApplication>();
    Dictionary<string, WindowsApplication> dictionary = new Dictionary<string, WindowsApplication>
    try
    {
      using (ManagementObjectSearcher managementObjectSearcher = new ManagementObjectSearcher(quer
      {
        foreach (ManagementObject managementObject in managementObjectSearcher.Get())
        {
          WindowsApplication windowsApplication = new WindowsApplication()
          {
            ApplicationId = (string) managementObject["ProgramID"],
            ApplicationName = (string) managementObject["Name"],
            ApplicationPublisher = (string) managementObject["Vendor"],
            ApplicationVersion = (string) managementObject["Version"],
            ApplicationLanguage = (string) managementObject["Language"]
```

Inventory


WMI Explorer 2.0.0.2 (Administrator)
File
Launch
Help
Computer
BB-W10-5
Connect
Mode
Asynchronous
Synchronous
Class Enumeration Options
Filter:
%
Include System Classes
Include CIM Classes
Include Perf Classes
Include MSFT Classes
Refresh Classes
Namespaces
\\BB-W10-5\ROOT
ROOT\Appv
ROOT\CIMV2
ROOT\CIMV2\mdm
ROOT\CIMV2\ms_409
ROOT\CIMV2\power
ROOT\CIMV2\Security
ROOT\CIMV2\TerminalServices
ROOT\Cli
ROOT\DEFAULT
ROOT\directory
ROOT\Hardware
ROOT\Interop
ROOT\MEM
ROOT\Microsoft
ROOT\msdtc
ROOT\PEH
ROOT\Policy
ROOT\RSOP
ROOT\SECURITY
ROOT\SecurityCenter
ROOT\SecurityCenter2
ROOT\ServiceModel
ROOT\StandardCimv2
ROOT\subscription
ROOT\WMI
Classes (418)
Search
Quick Filter:
Classes
Name
Lazy...
Description
Path
Win32_DMAChannel
Win32_DriverForDevice
Win32_DuplicateFileAction
Win32_Environment
Win32_EnvironmentSpecification
Win32_ExtensionInfoAction
Win32_Fan
Win32_FileSpecification
Win32_FolderRedirection
Win32_FolderRedirectionHealth
Win32_FolderRedirectionHealthC...
Win32_FolderRedirectionUserCon...
Win32_FontInfoAction
Win32_Group
Win32_GroupInDomain
Win32_GroupUser
Win32_HeatPipe
Win32_IDEController
Win32_IDEControllerDevice
Win32_ImplementedCategory
Win32_InfraredDevice
Win32_IniFileSpecification
Win32_InstalledProgramFramework
Win32_InstalledSoftwareElement
Win32_InstalledWin32Program
Instances (15)
Properties (7)
Methods (0)
Query
Script
Logging
Instance Options
Quick Filter:
Show Null Values
Show System Properties
Refresh Instances
Refresh Object
Instances
Properties
*ProgramId
Language
MsiPackageCode
MsiProductCode
Name
Vendor
Version
Microsoft Edge WebView2 Ru
Microsoft Corporation
116.0.1938.62
Name
Type - String
Program name
WQL Query (Selected Object)
Query
SELECT * FROM Win32_InstalledWin32Program WHERE ProgramId='000007475bf0297ef8f5849629a1f8c1e5810000ffff'
Execute
Retrieved 418 classes from ROOT\CIMV2 that match specified criteria.
Retrieved 15 instances from Win32_InstalledWin32Program
Time to Enumerate Instances: 00:03.640

Inventory



HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\IntuneManagementExtension\Inventories

Inventory

1. New apps appear in Intune reports fairly quickly
2. It can take up to 7 days for delta changes (removes) to be reflected in Intune reports
3. Inventory is run every 24 hours
4. A full inventory is run the first time and every 7 days, a delta inventory occurs subsequent runs
5. Delete FirstTimeRun key to force a full inventory

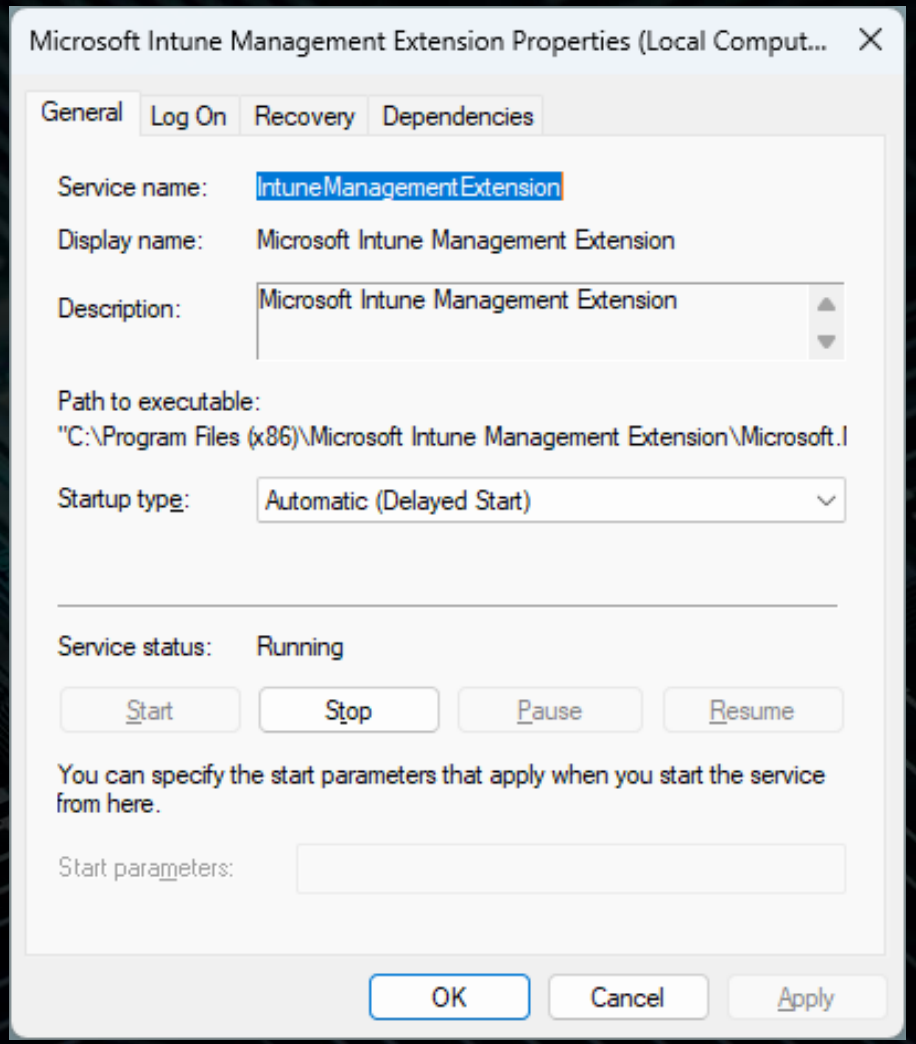
Invoking the IME

When does the IME process policy?

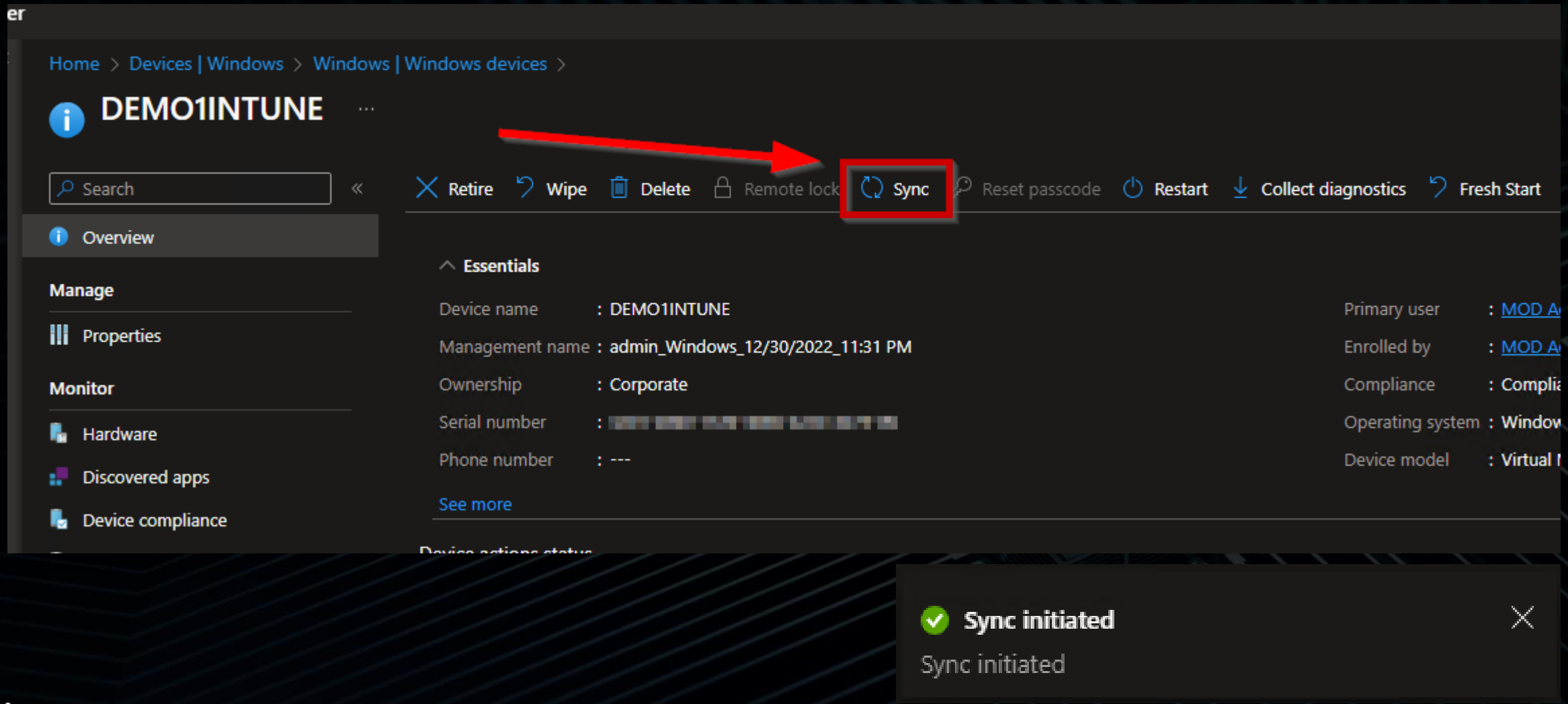
- When the service starts
- Every 60 minutes

Can you invoke the policy on-demand?

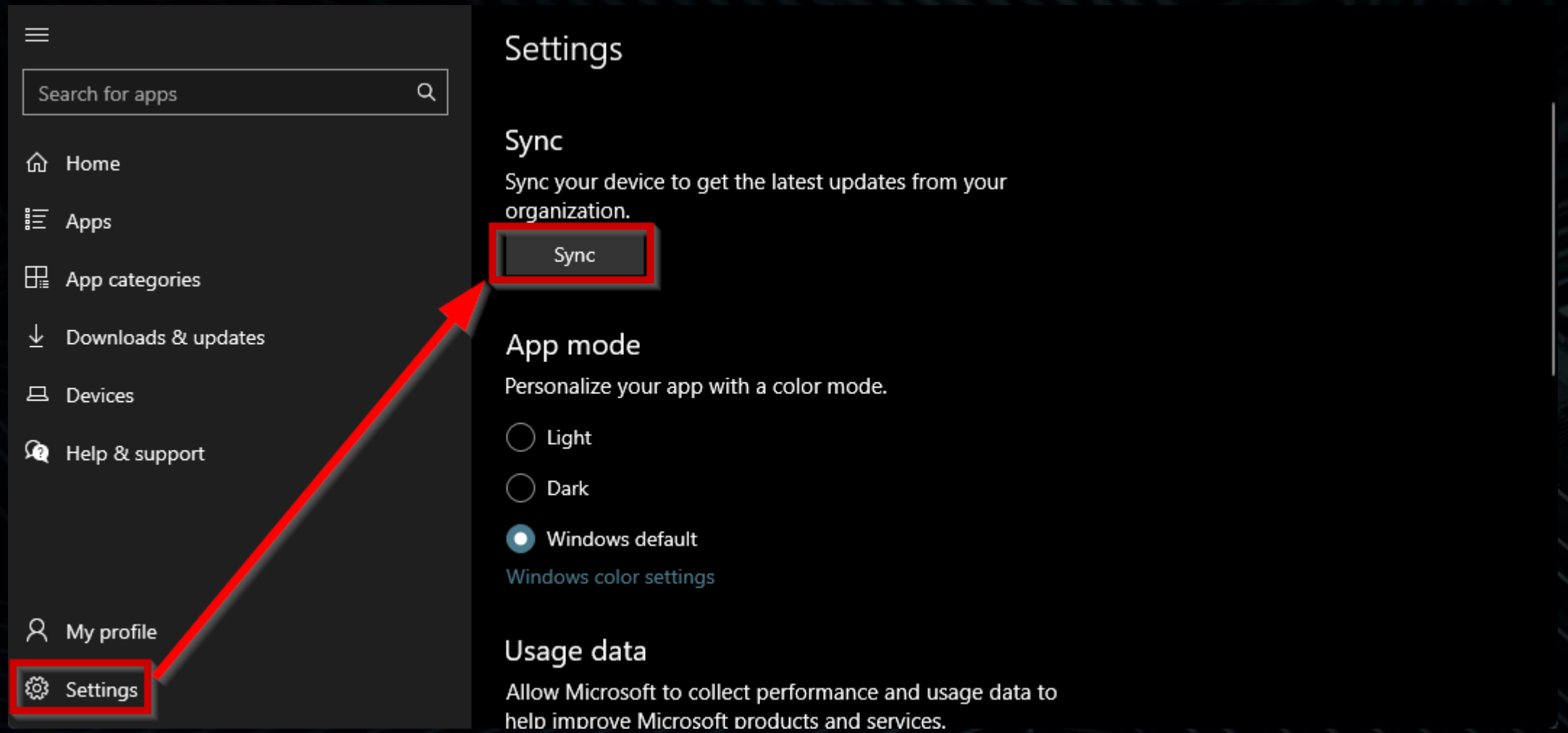
- Can emu's fly?



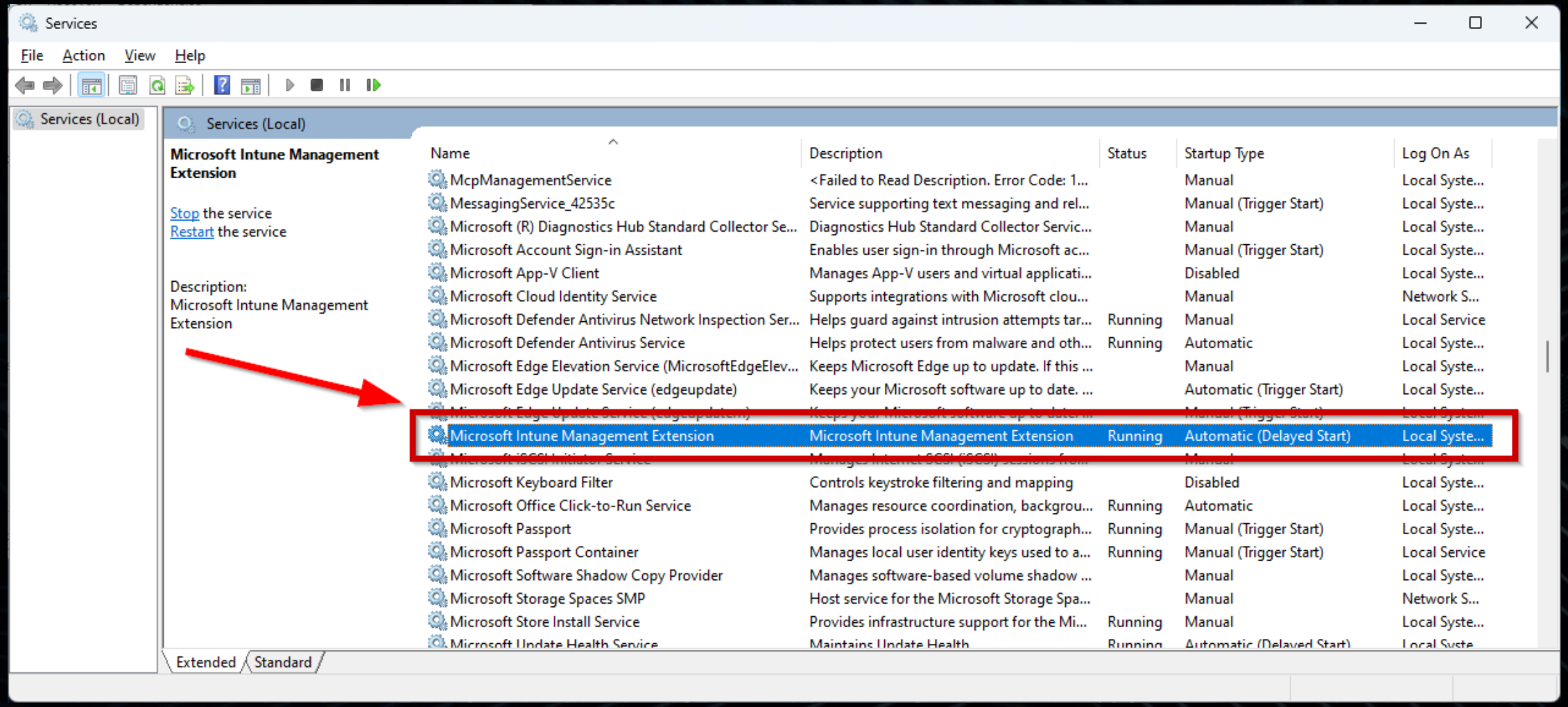
Invoking the IME



Invoking the IME



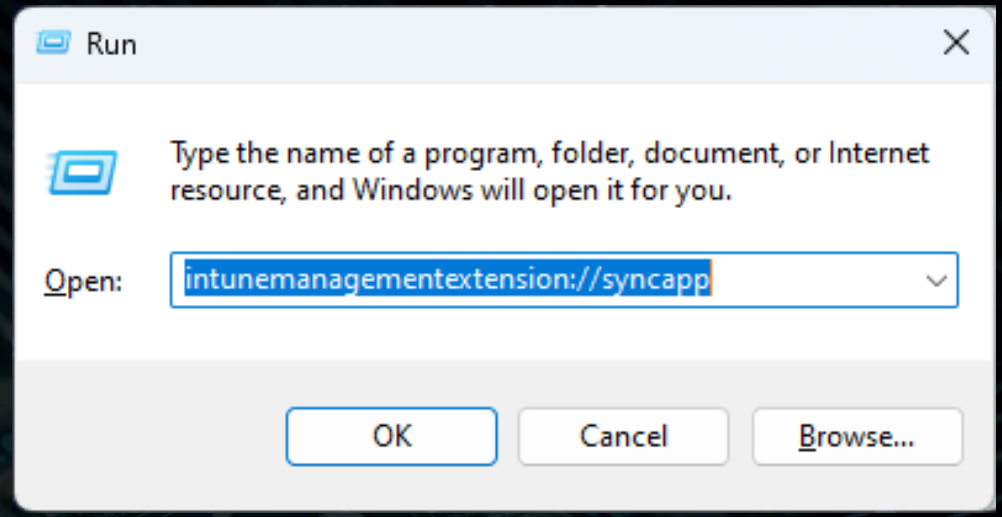
Invoking the IME

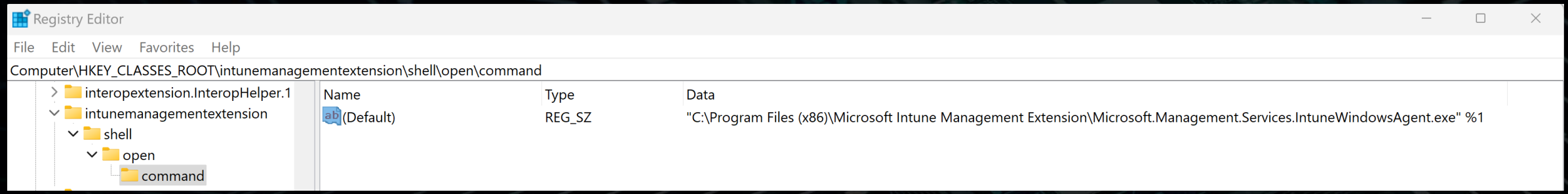


Invoking the IME

URL Moniker FTW

Thanks to Oliver Kieselbach (MVP)

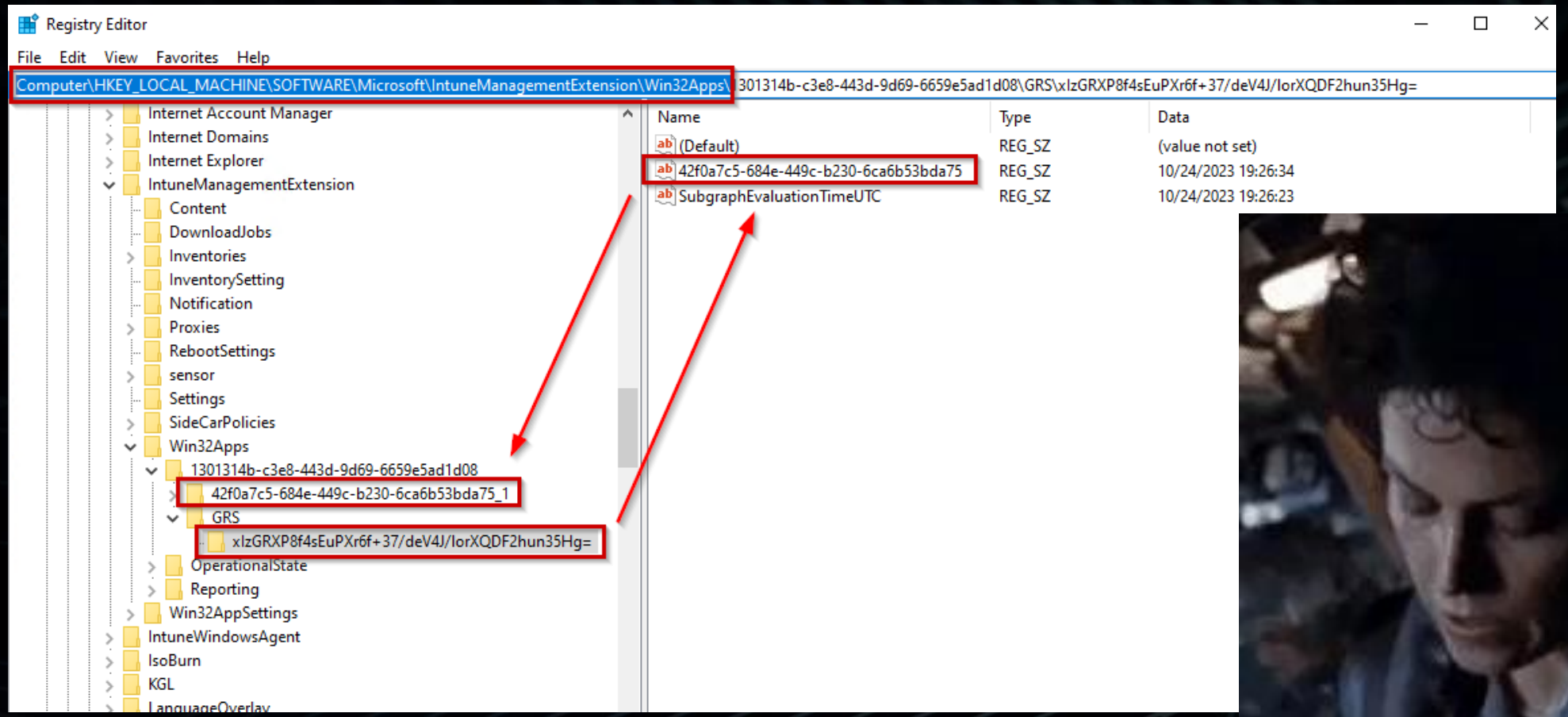





```
$newShell = New-Object -ComObject Shell.Application
$newShell.open("intunemanagementextension://syncapp")
```

https://oliverkieselbach.com/2020/11/03/triggering-intune-management-extension-ime-sync/

Invoking the IME



Q&A

Thank you again to our Sponsors

Microsoft

TUNEDIN
MEETUPS

TunedIn: London

8th December 2023