

# Evaluate installed app version from devices via Graph API

Managing applications across a fleet of devices is a critical task for IT administrators. With this PowerShell script you can leverage the power of Microsoft Intune and the Microsoft Graph API to streamline application inventory management.

By using this script, you can efficiently track application installations and versions, aiding in license compliance, security updates, and software distribution planning. It empowers IT teams to make informed decisions about their application landscape, ultimately enhancing device management and security.

## PowerShell Script

Add your own access token and app name in the corresponding PowerShell variables.

```
$Global:MicrosoftEntraIDAccessToken = "<youraccesstoken>"
$AppName = "<yourappname>"
$PlatformOS = "<yourplatform>" #possible values are windows, ios, macos, android
$AllDevicesWithAppVersion = @()

#Get Intune managed devices
$uri =
"https://graph.microsoft.com/v1.0/deviceManagement/managedDevices?`$filter=startswith(operatingSystem,'$PlatformOs')"
$Results = Invoke-RestMethod -Method GET -Uri $uri -ContentType "application/json" -Headers @{Authorization = "Bearer $($Global:MicrosoftEntraIDAccessToken)"; ConsistencyLevel = "eventual" }
$IntuneDevices = $results.value
if ($results."@odata.nextLink" -ne $null) {
    $NextPageUri = $results."@odata.nextLink"
    # While there is a next page, query it and loop, append results
    While ($NextPageUri -ne $null) {
```

```

    $NextPageRequest = (Invoke-RestMethod -Headers @{Authorization = "Bearer
$(Global:MicrosoftEntraIDAccessToken)" } -Uri $NextPageURI -Method Get)

    $NxtPageData = $NextPageRequest.Value
    $NextPageUri = $NextPageRequest."@odata.nextLink"
    $IntuneDevices += $NxtPageData
}
}

# Get Apps when
foreach ($IntuneDevice in $IntuneDevices) {
    try {
        $AppsUri =
        "https://graph.microsoft.com/beta/deviceManagement/manageddevices('$($IntuneDevice.id)')/detectedApps?`$t
op=100&`$filter=contains(displayName,%20%27$($AppName)%27)&`$orderBy=displayName%20asc"
        $Apps = Invoke-RestMethod -Method GET -Uri $AppsUri -ContentType "application/json" -Headers
        @{Authorization = "Bearer $(Global:MicrosoftEntraIDAccessToken)"; ConsistencyLevel = "eventual" }
        $IntuneDevice | Add-Member -NotePropertyName "$AppName AppVersion" -NotePropertyValue
        $apps.value.version
        Write-Output "$($IntuneDevice.deviceName) -> $($apps.value.version)"
    }
    catch {
        Write-Output "Sleeping..."
        Start-Sleep 10
        $AppsUri =
        "https://graph.microsoft.com/beta/deviceManagement/manageddevices('$($IntuneDevice.id)')/detectedApps?`$t
op=100&`$filter=contains(displayName,%20%27$($AppName)%27)&`$orderBy=displayName%20asc"
        $Apps = Invoke-RestMethod -Method GET -Uri $AppsUri -ContentType "application/json" -Headers
        @{Authorization = "Bearer $(Global:MicrosoftEntraIDAccessToken)"; ConsistencyLevel = "eventual" }
        $IntuneDevice | Add-Member -NotePropertyName "$AppName AppVersion" -NotePropertyValue
        $apps.value.version
        Write-Output "$($IntuneDevice.deviceName) -> $($apps.value.version)"
    }
    $AllDevicesWithAppVersion += $IntuneDevice
}

$AllDevicesWithAppVersion | Export-CSV ".\$(Get-Date -Format yyMMdd) AllDevicesWithAppVersion.csv"

```

Revision #6

Created 19 September 2023 12:41:47

Updated 21 July 2024 15:11:16