## Applying Managed Installer via Win32 App for Windows 11 SE

This is a work in progress. Some things to test

1. Do we still need a reboot? We used to, but with the recent changes to allow MI to be installed on Pro devices, that may or may not be necessary. The TAP program doesn’t seem to indicate that a reboot is needed, but it’s possible they aren’t aware that one is required and may be why we see so many issues.

## Create Win32 App to Configure Managed Installer

Managed Installer is a function of WDAC, however it’s configured similar to Applocker. There’s a XML configuration that specifies which exes are allowed to be managed installers. In this guide, the Intune Management extension, the OMADMClient, and Google Chrome are configured as Managed Installers. Chrome is added as a managed installer since it updates itself with a different executable (GoogleUpdate.exe). This exe needs to be configured as a managed installer for Chrome to update itself. If you don’t want Chrome as a managed installer, the ApplyManagedInstaller.ps1 file can be modified to remove GoogleUpdate.exe as a managed installer. If you have other apps you wish to allow to update themselves, you can modify the ApplyManagedInstaller.ps1 file to include the exes. You’ll need to re-package the ps1 file into a new intunewin file if you wish to modify the ApplyManagedInstaller.ps1.

The ApplyManagedInstaller.intunewin file contains a the ApplyManagedInstaller.ps1 script that is responsible for configuring the Managed Installer. This can only be done via script as there aren’t any MDM Configuration Service Providers (CSPs) for managed installer functionality.

Whatever executable file(s) are designated as managed installers must also be restarted after the managed installer is configured. This poses a challenge when trying to configure during Autopilot as it requires the Intune Management Extension to be restarted, which can impact how the Enrollment Status Page tracks application installation status. Because of this requirement, the Apply Managed Installer PowerShell script requires a reboot and the Win32 app must be configured to kick off that reboot.

1. In the **Microsoft Endpoint Manager console** (endpoint.microsoft.com) select **Apps – All Apps**
2. Click **Add**
3. Under **App type** select **Windows app (Win32)** and click **Select**
4. On the **App information** page click **Select app package file**
5. On the **App package file** flyout page, select the **folder icon** for the App package file and select **applymanagedinstaller.intunewin** and click **Open** then click **OK**
6. Back on the **App information** tab, change the **name** of the application to **Apply Managed Installer**
7. Click the **Edit Description link** and in the **Edit description** flyout enter an appropriate description and click **OK**
8. For the **Publisher** enter **Microsoft** and click **Next**
9. On the **Program** tab, for the **Install command** enter   
     
   **cmd /c Powershell -executionpolicy Bypass -command "& {. .\ApplyManagedInstaller.ps1}"**
10. For the **Uninstall command**, enter   
      
    **cmd /c**  
      
    Note: Something must be entered here, but this command won’t remove the script
11. For **Device restart behavior** select **Intune will force a mandatory device restart** and click **Next**
12. On the **Requirements** tab, for **Operating system architecture** select **64 bit**
13. For **Minimum operating system** select **Windows 10 1703** and click **Next**
14. On the **Detection rules** tab under **Rules format** select **Manually configure detection rules** and click **Add**
15. On the **Detection rule** flyout under **Rule type** select **File**
16. For **Path** enter   
      
    **c:\windows\system32\applocker**
17. For File or folder enter  
      
    **ManagedInstaller.Applocker**
18. For **Detection method** select **File or folder exists** and click **OK** then click **Next**
19. On the **Dependencies** tab click **Next**
20. On the **Supersedence** tab click **Next**
21. On the **Scope tags** tab click **Next**
22. On the **Assignments** tab select a group that you wish to deploy the application to and click **Next**  
      
    Note: Using All Devices and assigning an Include Filter is the recommended approach as it reduces the amount of time for the assignment to reach a device, especially in new device provisioning scenarios.
23. On the **Review + create** tab click **Create**

### Optional - Modify Enrollment Status Page to Require Managed Installer Win32 App

This step is optional only if you are not using the Enrollment Status Page with Autopilot or Provisioning. If you are using the Enrollment Status Page, then you must include the Managed Installer Win32 app as required. If you don’t, then the Managed Installer Win32 app will reboot the machine after the user has signed in. It’s best to have this reboot happen during enrollment.

1. In the **Microsoft Endpoint Manager console** (endpoint.microsoft.com) select **Devices**
2. On the **Devices** page select **Enroll devices**
3. On the **Enroll devices** page select **Enrollment Status Page**
4. On the **Enrollment Status Page** for the **Default** page select **All users and all devices**  
     
   Note: You can also select another page if you have created your own custom page instead of using Default
5. On the **All users and all devices** page select **Properties**
6. On the **Properties** page next to **Settings** click **Edit**
7. On the **Edit profile** page scroll to the bottom and under **Block device use until these required apps are installed if they are assigned to the user/device**, if Selected is selected (meaning you are only requiring specific applications to be installed), click **Select Apps** and select **Apply Managed Installer**  
   It will look something like this  
   Text

   Description automatically generated
8. Click **Review + save**
9. Click **Save**