# Deploying WDAC Managed Installer with Autopilot and Intune

This guide is designed to prevent students/standard users from being able to install applications on Windows devices while still allowing IT admins to deploy applications via Microsoft Intune. This leverages Windows Defender Application Control (WDAC), Managed Installer, and Applocker policies to control the environment.

## Prerequisites

Windows Enterprise or Education - If the device is starting out as Pro or Pro Education and you’re using Autopilot, a MAK can be used via an Edition Upgrade policy in Intune to upgrade the device to Enterprise or Education.

If you’re reimaging the device with Enterprise or Education, an edition upgrade key will not be necessary if you plan to include the key in your image.

Subscription activation may work, however the upgrade to Enterprise or Education happens after the user signs in. This means the WDAC/Managed Installer policies may not apply until after the user signs in and a reboot takes place. This leaves the machine open to the possibility of a user being able to install things before the policies are all applied successfully.

**To download WDAC and Managed Installer files, click** [**here**](https://github.com/rbalsleyMSFT/managedinstaller) and under the green Code button select Download ZIP.  
  
Graphical user interface, text, application, email

Description automatically generated  
  
In the ManagedInstaler-main.zip file, go into the managedinstaller-main directory and copy out/extract the WDAC\_MI\_Files directory.  
  
Graphical user interface, text, application

Description automatically generated

## Create WDAC Policy

1. Open a browser and go to **endpoint.microsoft.com**
2. Click on **Devices – Configuration Profiles**
3. Click **Create Profile**
4. In the **Platform** drop down select **Windows 10 and later**
5. In the **Profile Type** drop down select **Templates**
6. For **Template name** select **Custom** and click **Create**
7. On the Basics tab for **Name** enter **WDAC Policy** and click **Next**
8. On the Configuration settings tab for **OMA-URI Settings** click **Add**
9. On the Add Row flyout for **Name** enter **WDAC Policy**
10. For **OMA-URI** enter   
    **./Vendor/MSFT/ApplicationControl/Policies/7ff774d3-0c76-44b4-be1d-1debcee783db/Policy**  
    Note: Make sure you include the period in front of /Vendor
11. For **Data type** select **Base64 (file)**
12. Click the **folder icon** and navigate to the **WDAC\_MI\_Files\{7ff774d3-0c76-44b4-be1d-1debcee783db}.bin** file and click **Open**  
      
    Note: The WDAC\_MI\_Files folder contains a .bin and .cip file. Both files are provided depending on how you want to apply the WDAC policy. The .cip file is used when manually applying policy to a device or using provisioning packages. Intune only accepts .bin files.
13. Click **Save**
14. Back on the **Configuration settings** tab click **Next**
15. On the **Scope tags** tab click **Next** unless you know you need to assign a scope tag
16. On the **Assignments** tab select a group that you wish to deploy the policy 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.  
      
    More info on filters - [Create filters in Microsoft Intune | Microsoft Docs](https://docs.microsoft.com/en-us/mem/intune/fundamentals/filters?msclkid=fa061155cfe211eca8a1b879dce3bc7b)
17. On the **Applicability rules** tab click **Next**
18. On the **Review + create** tab click **Create**

## Create Deny Administrative Apps Applocker Policy

While WDAC can include rules to block exes, it applies at the device level. This means every user would be blocked from using those exes on the device. Often in EDU environments admins need access to command prompt, PowerShell, etc. for troubleshooting purposes while students should not have access to these. We can create a separate AppLocker policy to prevent students from accessing specific exes while admins will have the ability to use them.

1. In the Microsoft Endpoint Manager console (endpoint.microsoft.com), go to **Devices – Configuration Profiles**
2. Click **Create profile**
3. Select Platform **Windows 10 and later** and Profile type **Templates - Custom** and click **Create**
4. On the Basics page, enter a name for the custom profile and click Next
5. On the Configuration settings page, click **Add**
6. On the Add Row flyout, enter a **Name** and description (optional)
7. For the **OMA-URI**, enter  
   **./Device/Vendor/MSFT/AppLocker/ApplicationLaunchRestrictions/DenyApps/EXE/Policy**
8. For the **Data Type** select **String (XML file)**
9. Click the **folder icon** and navigate to the **WDAC\_MI\_Files**\**ApplockerDenyAdministrativeApps.xml** file and click **Open**
10. Click **Save**
11. Back on the **Configuration settings** tab click **Next**
12. On the **Scope tags** tab click **Next** unless you know you need to assign a scope tag
13. On the **Assignments** tab select a **user group** that you wish to deploy the policy to and click **Next**  
      
    Note: While you can select a device group, a user group would be preferred in this case. This will block students from using cmd, powershell, reg, regedit, powershell\_ise on any machine. If you want to allow students to use those exes on certain machines, you can create an exclude filter to exclude certain machines (maybe lab machines where some students may be using command line tools for computer science).   
      
    More info on filters - [Create filters in Microsoft Intune | Microsoft Docs](https://docs.microsoft.com/en-us/mem/intune/fundamentals/filters?msclkid=fa061155cfe211eca8a1b879dce3bc7b)
14. On the **Applicability rules** tab click **Next**
15. On the **Review + create** tab click **Create**

## 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**