Windows 11 Readiness for MECM

# Overview

In order to help you prepare for your journey to Windows 11, we’ve developed a process that adds devices to a device collection based on the returned results of a PowerShell script deployed to clients. The PowerShell script uses .NET API to retrieve details about hardware on the device and whether they meet the hardware requirements in place for Windows 11.

This will allow you to get a good overall picture of your environment and what actions you can take now and what actions you can take in the future if device hardware is upgraded. For instance, you can get a collection of all devices that don’t meet drive space requirements, create a plan to upgrade those device’s drives or free up space on the drives

# Importing the Device Collection

We have provided a .MOF file that will create the seven device collections in your Configuration Manager. These seven device collections are named based on the results from the PowerShell script that we will deploy later. The results from the script are stored on the CM database, where the query rule retrieves the results and modifies the membership as needed.

To import these collections, navigate to Assets and Compliance, right click on Device Collections, and select “Import Collections”. In the Import collections Wizard, select “Next”, and Browse for the path where the MOF is saved. Select “Next” until Completion, then close the Wizard.

There should now be seven new collections in your Device Collections. If you would like to create a folder and move the newly created collections to that folder, you may. The limiting collection for the created collections is the default “All Systems” collection. The limiting collection can be changed through the collection’s properties, or the MOF file can be modified in any text editor to add the desired limiting collection’s ID.

Each collection will have its own block in the MOF. These blocks contain the basic data needed to create the collections. To change the limiting collection for the created collections, add the line LimitToCollectionID = “{Collection ID}”; just under the RefreshType = 2; line on each block. For example, if my limiting collection’s ID is PRI0001C, my first block would look like Figure 1.

Once the device collections have been created, the PowerShell script can be imported and ran on the client devices to start populating the collections.

Beginning of the MOF file with LimitToCollectionID = "PRI0001C"; after the refresh type line. 


Figure

# Importing and Running the Script

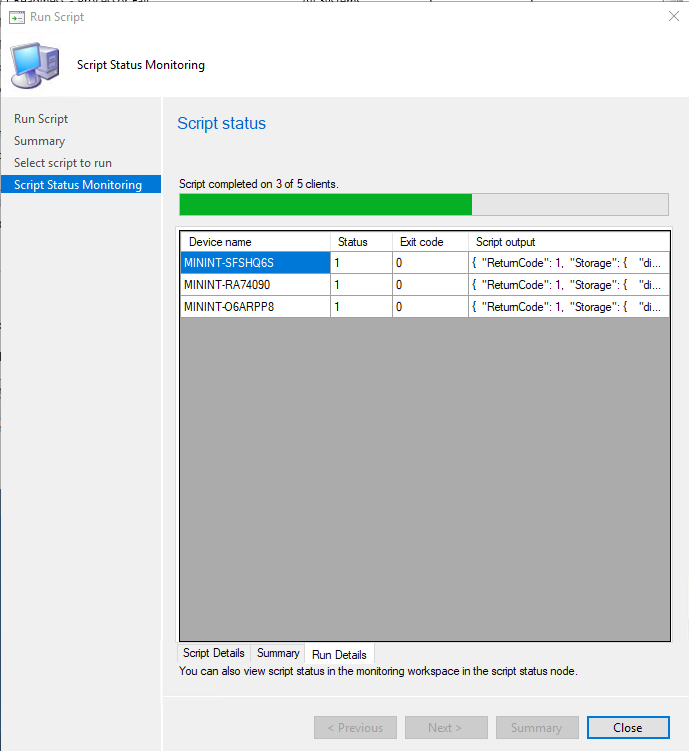
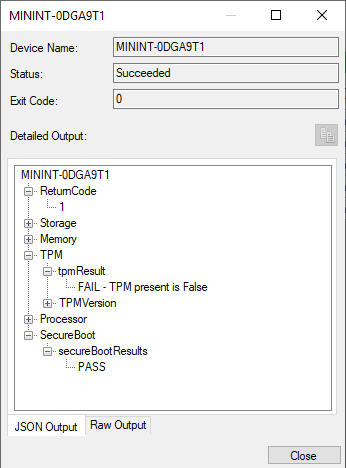
The script that we will be importing is a slightly modified version of the HardwareReadiness.PS1 script shared on the  [Microsoft Endpoint Manager Blog](https://techcommunity.microsoft.com/t5/microsoft-endpoint-manager-blog/understanding-readiness-for-windows-11-with-microsoft-endpoint/ba-p/2770866) (<https://aka.ms/HWReadinessScript>). To export the results locally on the device, and to make the results easier to read within MECM, and have a different entry per status reason, there is a custom PS object created using the original script’s logic, and the object is used for the results and exported document.

To import the script, navigate to Software Library, right click on Scripts, and select “Create Script”. Once in the Create Script window, give the script a name, select “Import”, browse to the script and Open it, and select “Next” until completion, then close the window.

Once the script has been imported, it will need to be Approved. To do this, right click on the script, and select “Approve/Deny”. In the Approve or Deny Script window, select “Next” until completion and Close the window.

Now that the script is approved, it can be run on client devices and the results stored. This can be done on any device collection that you’d like to collect details on.

Navigate to Assets and Compliance, right click on the device collection that you would like to run the script on and select “Run Script”. In the Run Script window, select the imported script, click “Next” until the Script Status Monitoring page. In this page, we can monitor the output of the scripts as it runs (Figure 2). We can also get more details about devices and drill down into the reasons why they may have failed (Figure 3).

****

Figure

Figure

Now that results are being returned to MECM, the groups will start populating with matching devices.

Note: The script must be approved by an admin other than the user that uploaded it, unless this has been explicitly disabled in your environment. Devices’ client settings must have the PowerShell execution policy set to Bypass.

# Conclusion

We believe that Windows 11 will help keep your organization and your users safe and secure while providing a more modern, yet familiar, user experience. We hope that this helps you get greater insights into your environment as you plan your organization’s journey to Windows 11. Please reach out to your account manager if you’d like assistance getting this setup in your environment or if you’d like guidance from Microsoft on your journey to Windows 11. We look forward to working with you.