# Deploy WDAC USING SCCM

## Contents



Windows Defender Application Control is designed to protect devices against malware and other untrusted software. It prevents malicious code from running by ensuring that only approved code, that you know, can be run.

Application Control is a software-based security layer that enforces an explicit list of software that is allowed to run on a PC. On its own, Application Control doesn't have any hardware or firmware prerequisites. Application Control policies deployed with Configuration Manager enable a policy on devices in targeted collections that meet the minimum Windows version and SKU requirements outlined in this article. Optionally, hypervisor-based protection of Application Control policies deployed through Configuration Manager can be enabled through group policy on capable hardware.

You can use Configuration Manager to deploy an Application Control policy. This policy lets you configure the mode in which Application Control runs on devices in a collection.

You can configure one of the following modes:

- **Enforcement enabled** - Only trusted executables are allowed to run.
- **Audit only** - Allow all executables to run, but log untrusted executables that run in the local client event log.

## Supported WDAC Policies in SCCM

Deploying WDAC policy using SCCM only supports the following rule types:

- **Managed Installer** – When a policy is created using SCCM, SCCM will be registered as a "Managed Installer". This means any application deployed using SCCM will be allowed to run
- **File** – Any executable selected during the policy creation will be allowed to run
- **Folder** – Any executable running in the selected folder will be allowed to run

Since managed installer is a heuristic-based mechanism, it doesn't provide the same security guarantees that explicit allow or deny rules do. The managed installer is best suited for use where each user operates as a standard non-administrative user and where all software is deployed and installed by a software distribution solution, such as SCCM.

Users with administrator privileges, or malware running as an administrator user on the system, may be able to circumvent the intent of Windows Defender Application Control when the managed installer option is allowed.

If a managed installer process runs in the context of a user with standard privileges, then it's possible that standard users or malware running as standard user may be able to circumvent the intent of Windows Defender Application Control.

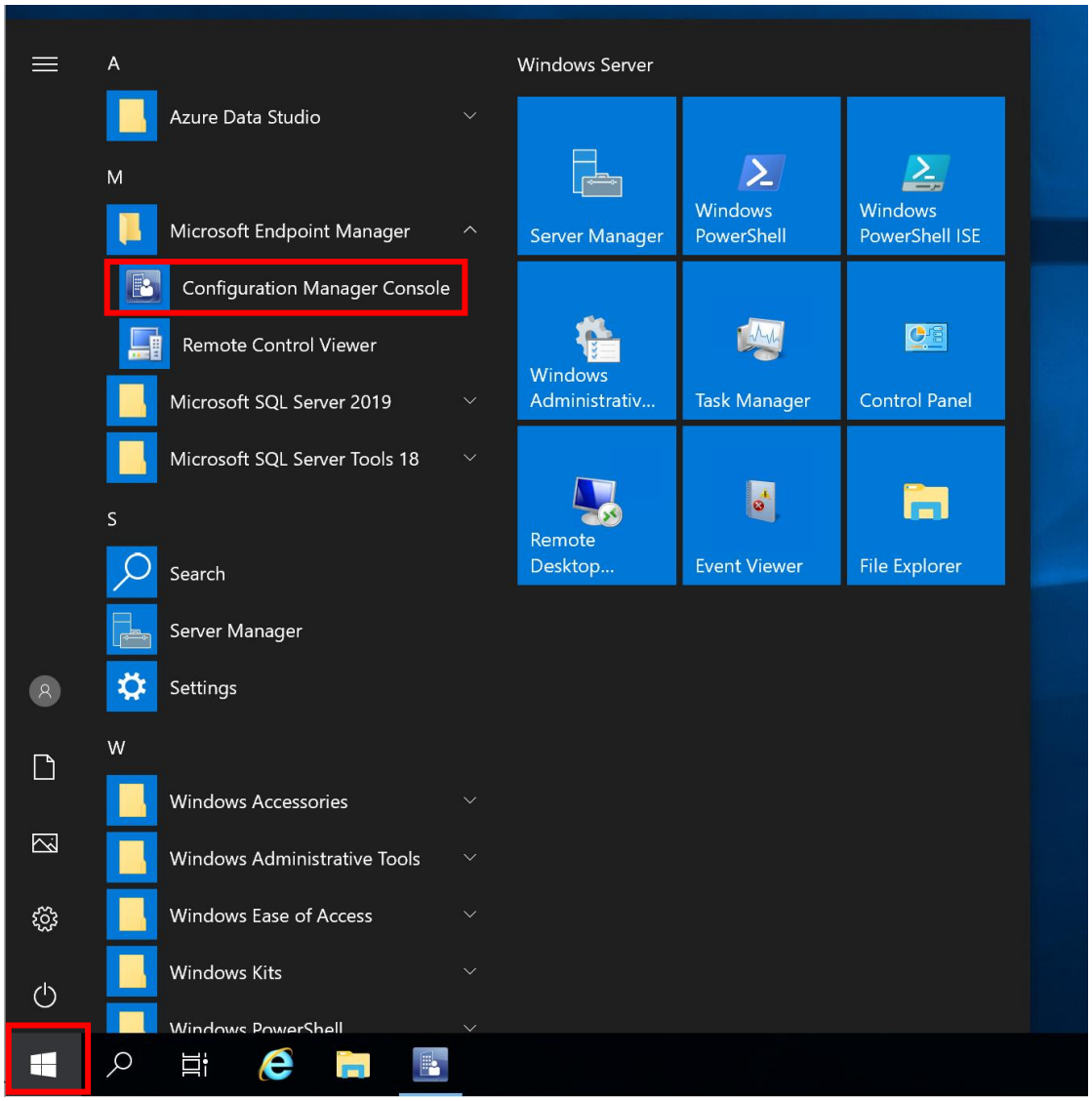
### WDAC limitations with managed installer

- WDAC policies, based on managed installer, doesn't support applications that self-update. An application that self-updates is an application that updates its self outside of the managed installer process such as an update pushed through SCCM. If an application that was deployed by a managed installer later updates itself, the updated application files won't include the origin information from the managed installer, and they might not be able to run. When you rely on managed installers, you must deploy and install all application updates by using a managed installer, or update the WDAC policy rules to authorize the newly updated app to run.
- Packaged apps (MSIX) deployed through a managed installer aren't tracked by the managed installer and will need to be separately authorized in your WDAC policy.
- Some applications or installers may extract, download, or generate binaries and immediately attempt to run them. Files run by such a process may not be allowed by the managed installer. In some cases, it may be possible to also designate an application binary that performs such an operation as a managed installer.
- The managed installer heuristic doesn't authorize kernel drivers. The WDAC policy must be updated to have rules that allow the necessary drivers to run or a supplemental policy must be created

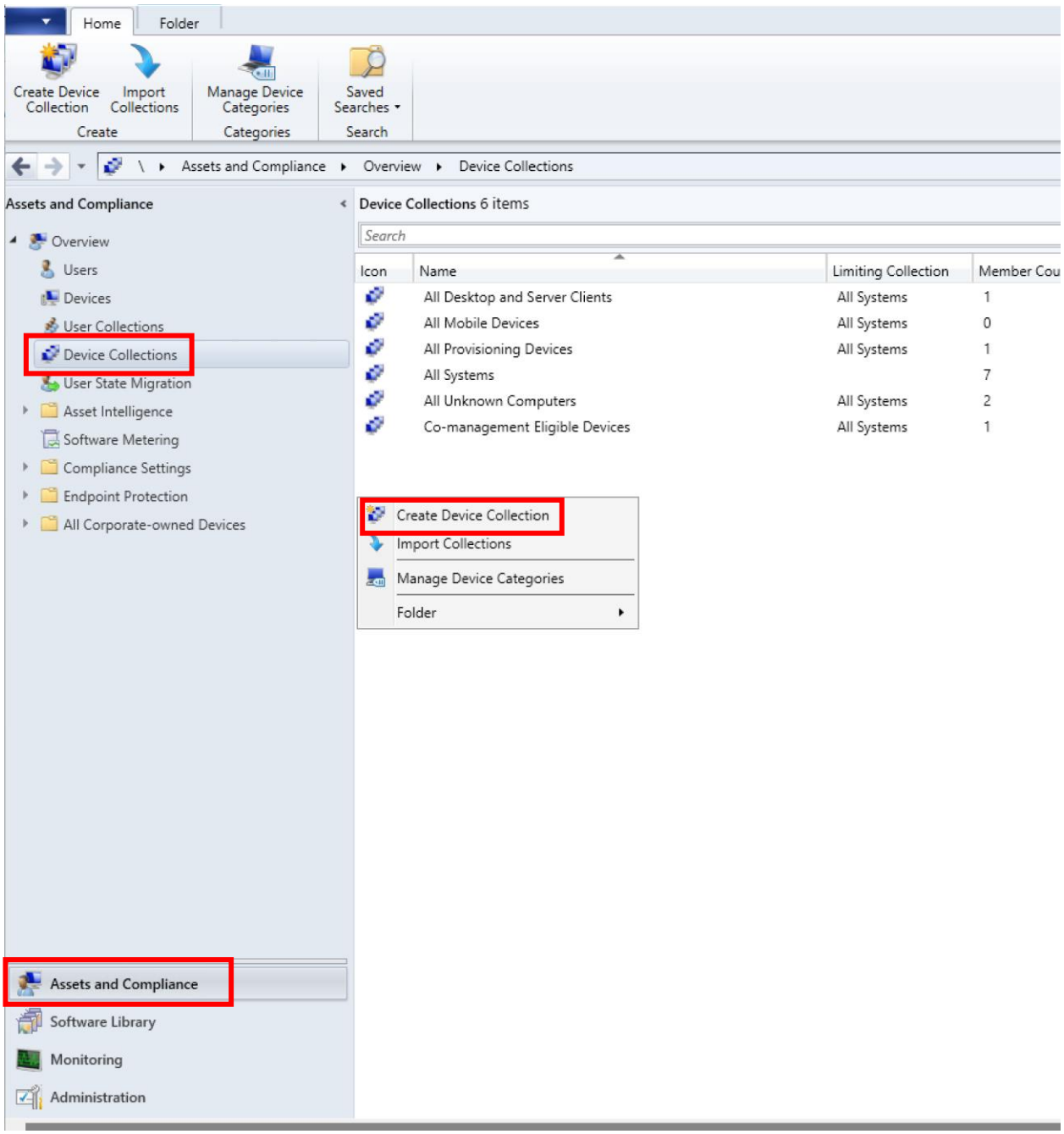
# Deploy WDAC policy using SCCM

## Create Device Collection

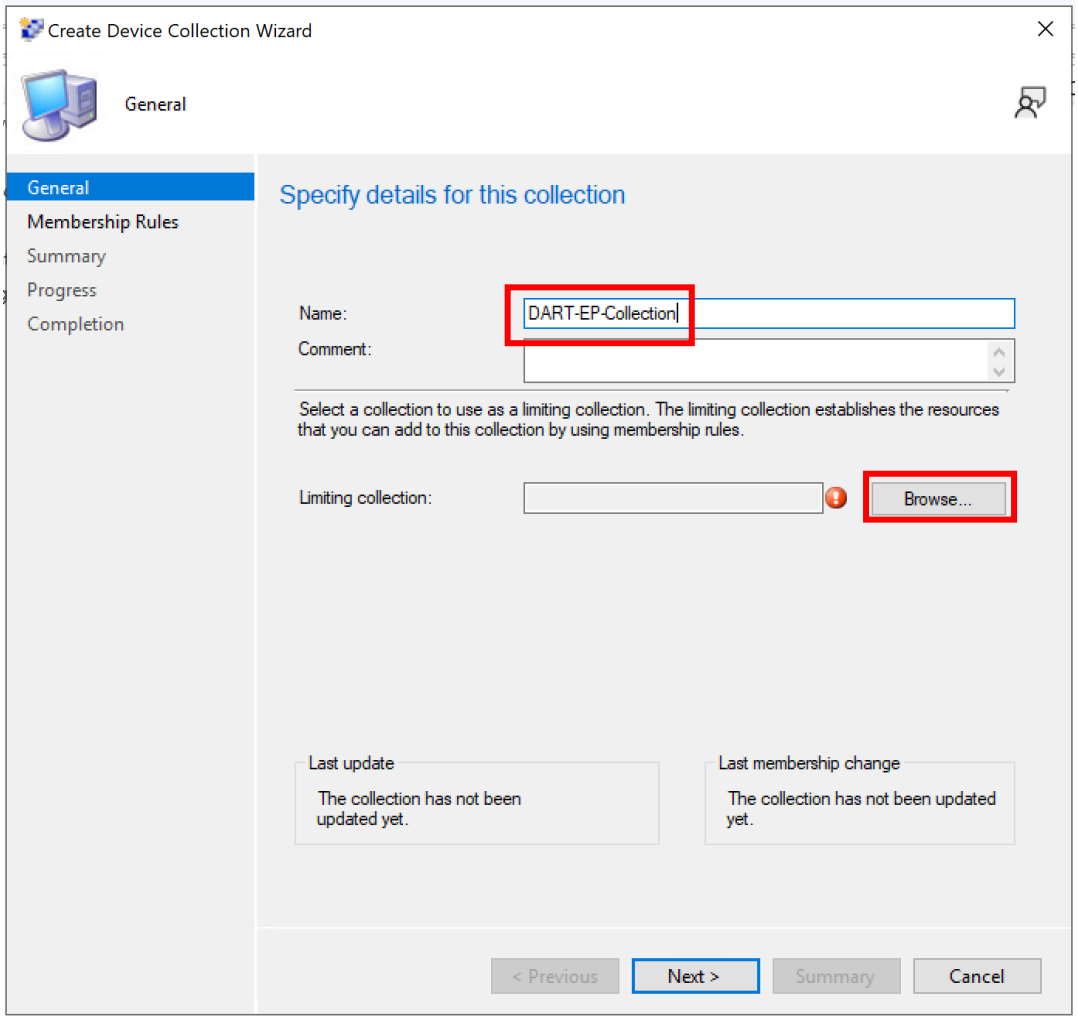
1. Log into the Primary Site system
2. Open the



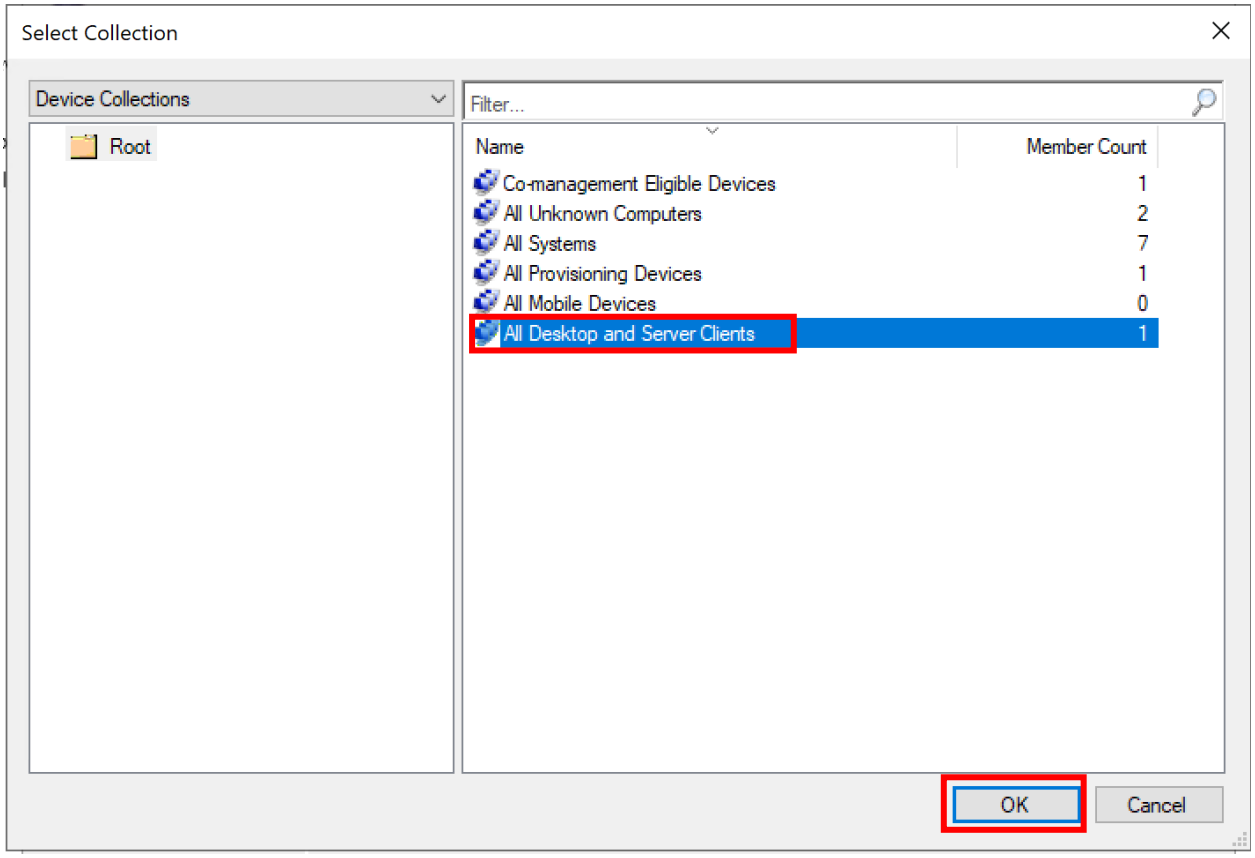

3. In the bottom left select **Assets and Compliance → Device Collections → Right click → Create Device Collection**



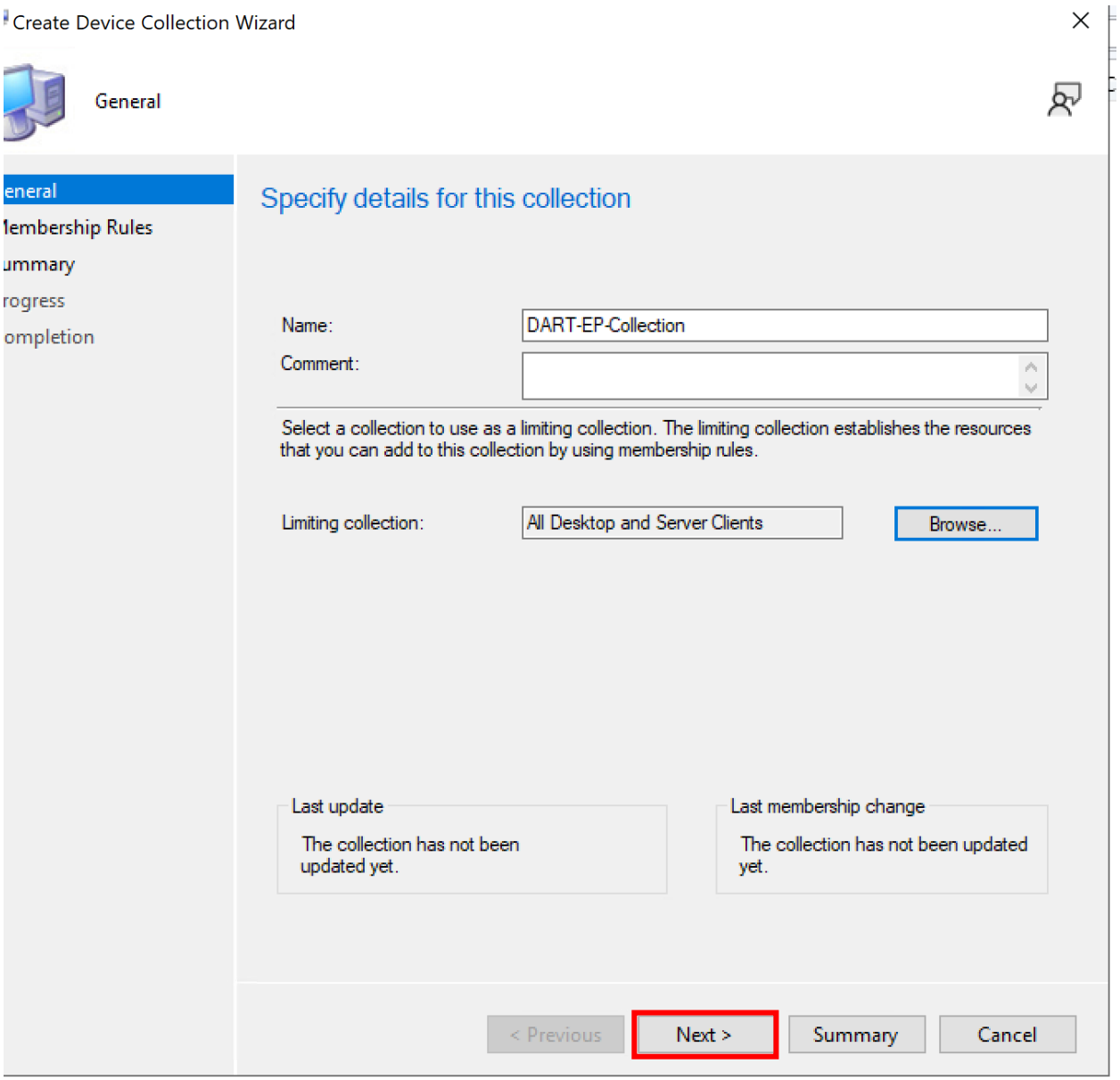

4. Enter a **Name → Browse**



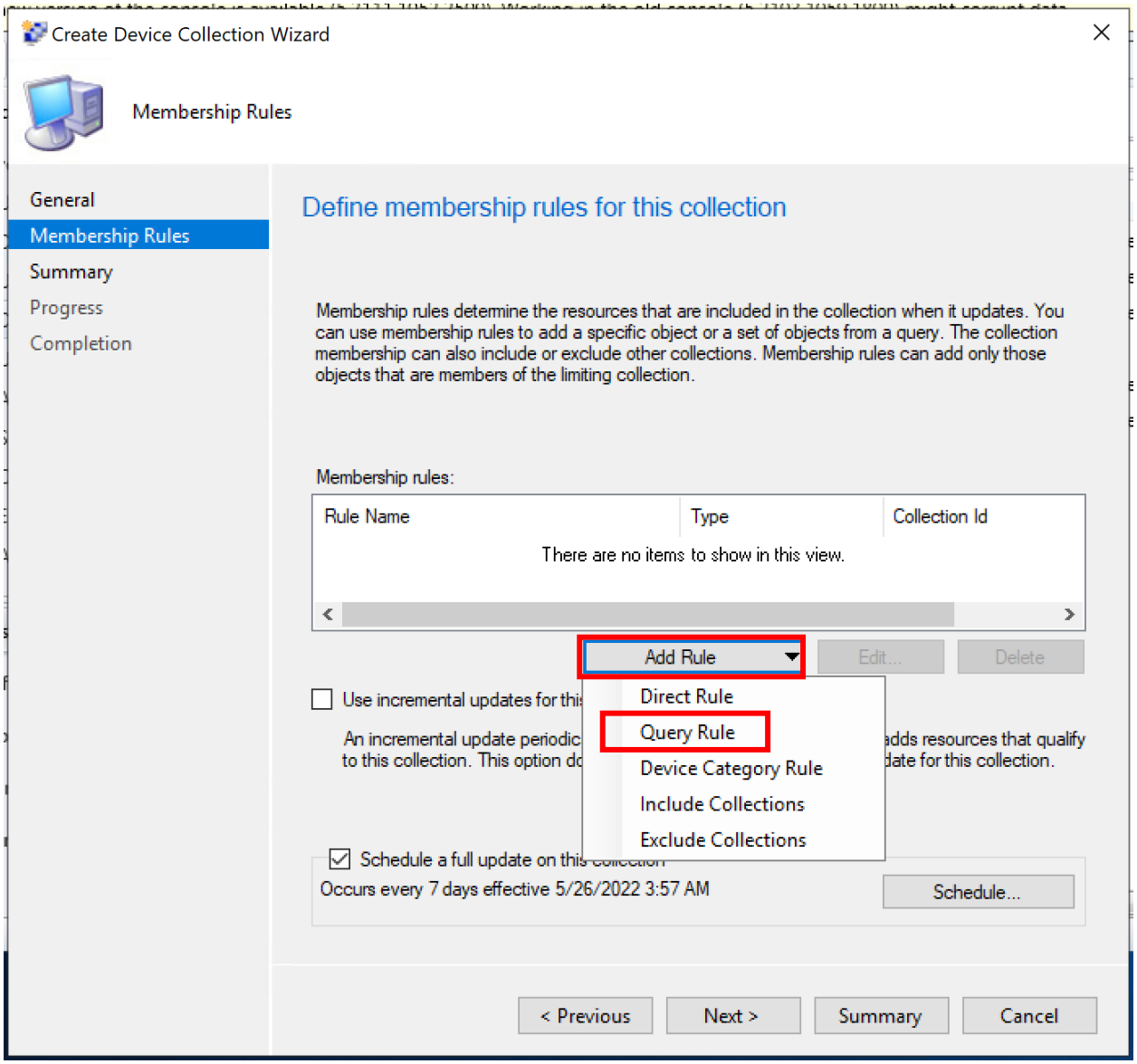

5. Select **All Desktop and Server Clients →**



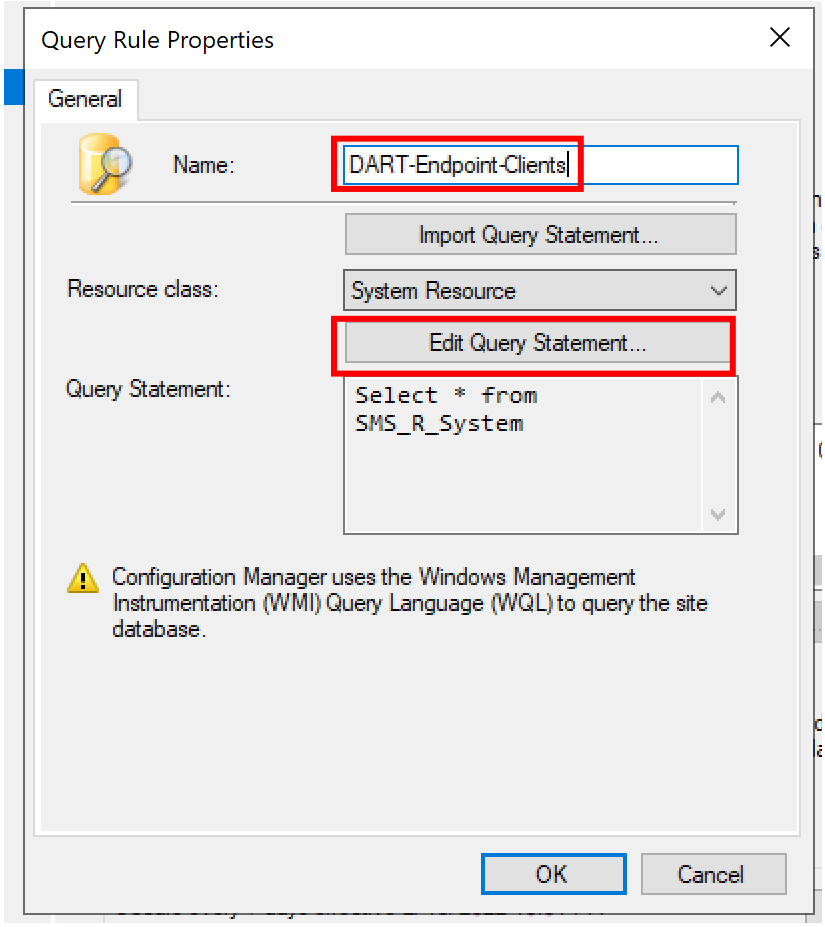

6. Select **Next**



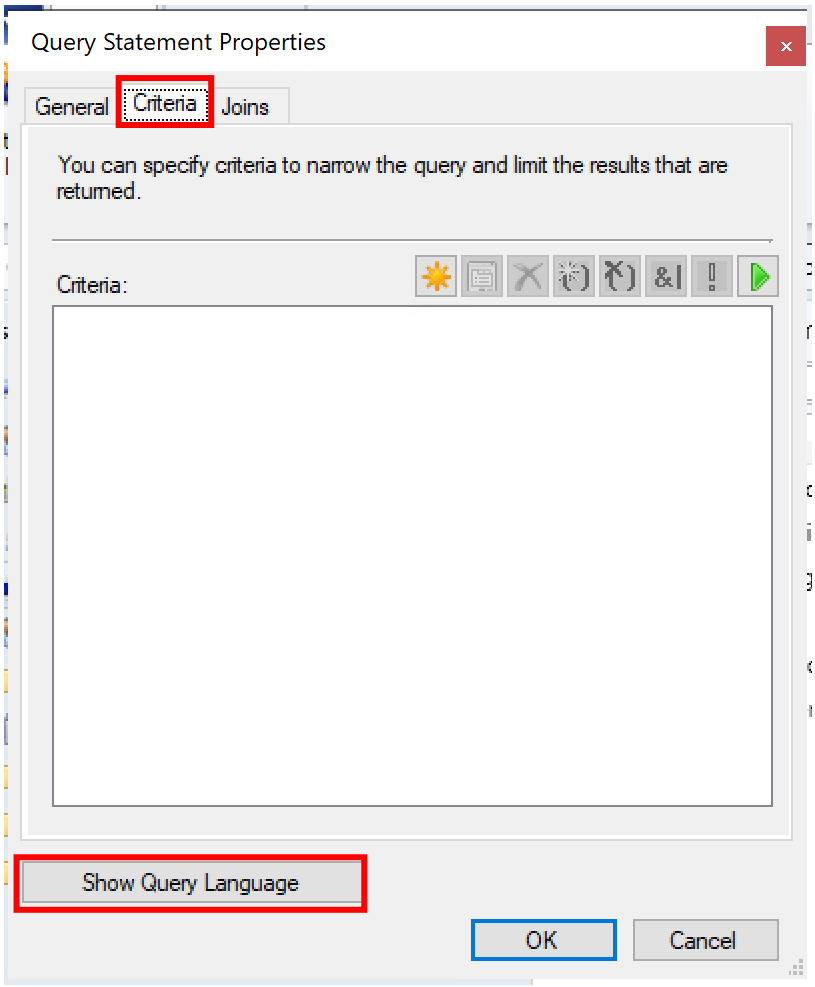

7. Select **Add Rule → Query Rule**




8. Enter a query name
9. Select **Edit Query Statement**

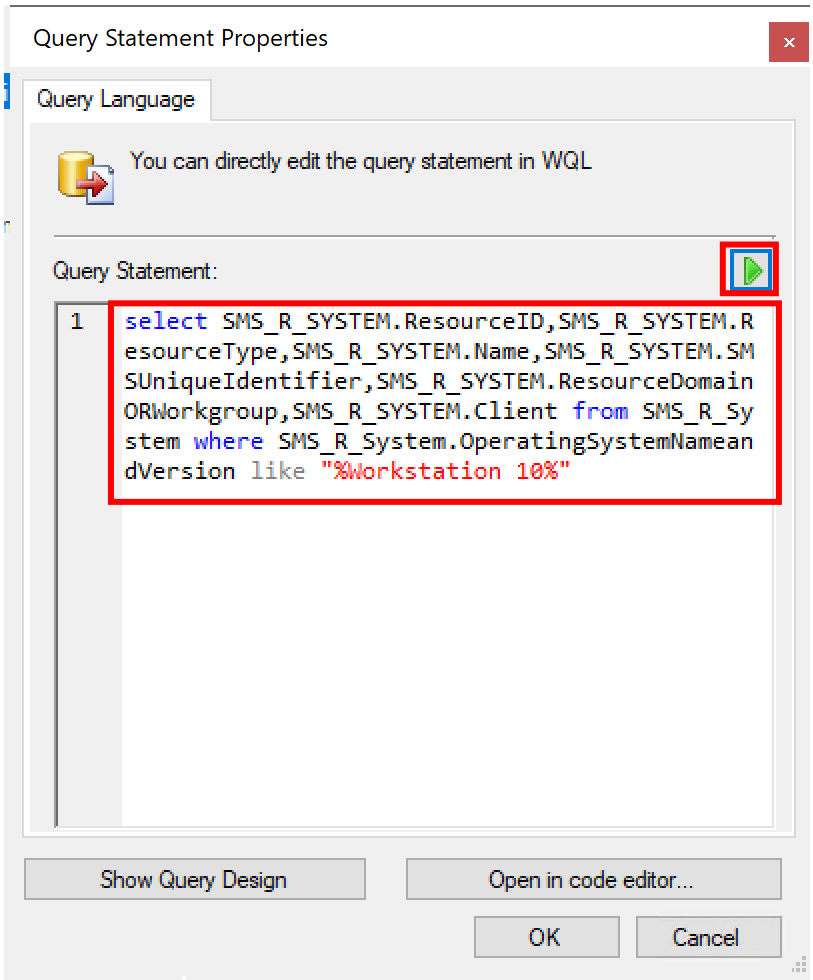



10. Select **Criteria → Show Query Language**




11. Cut and paste the following queries depending on the Operating system using the examples below. Use the ledger below to determine which query is best for you.

12. Select the **Green** triangle. Clicking the **Green** triangle will cause the query to run and return results




**Example query 1 - Windows 11 ONLY**

select SMS_R_SYSTEM.ResourceID,SMS_R_SYSTEM.ResourceType,SMS_R_SYSTEM.Name,SMS_R_SYSTEM.SMSUniqueIdentifier,SMS_R_SYSTEM.ResourceDomainORWorkgroup,SMS_R_SYSTEM.Client from SMS_R_System inner join SMS_G_System_OPERATING_SYSTEM on SMS_G_System_OPERATING_SYSTEM.ResourceID = SMS_R_System.ResourceId where SMS_G_System_OPERATING_SYSTEM.Name like "%Microsoft Windows 11 Enterprise%"

**Example query 2 - Windows 10 ONLY**

select SMS_R_SYSTEM.ResourceID,SMS_R_SYSTEM.ResourceType,SMS_R_SYSTEM.Name,SMS_R_SYSTEM.SMSUniqueIdentifier,SMS_R_SYSTEM.ResourceDomainORWorkgroup,SMS_R_SYSTEM.Client from SMS_R_System where SMS_R_System.OperatingSystemNameandVersion like "%Workstation 10%"

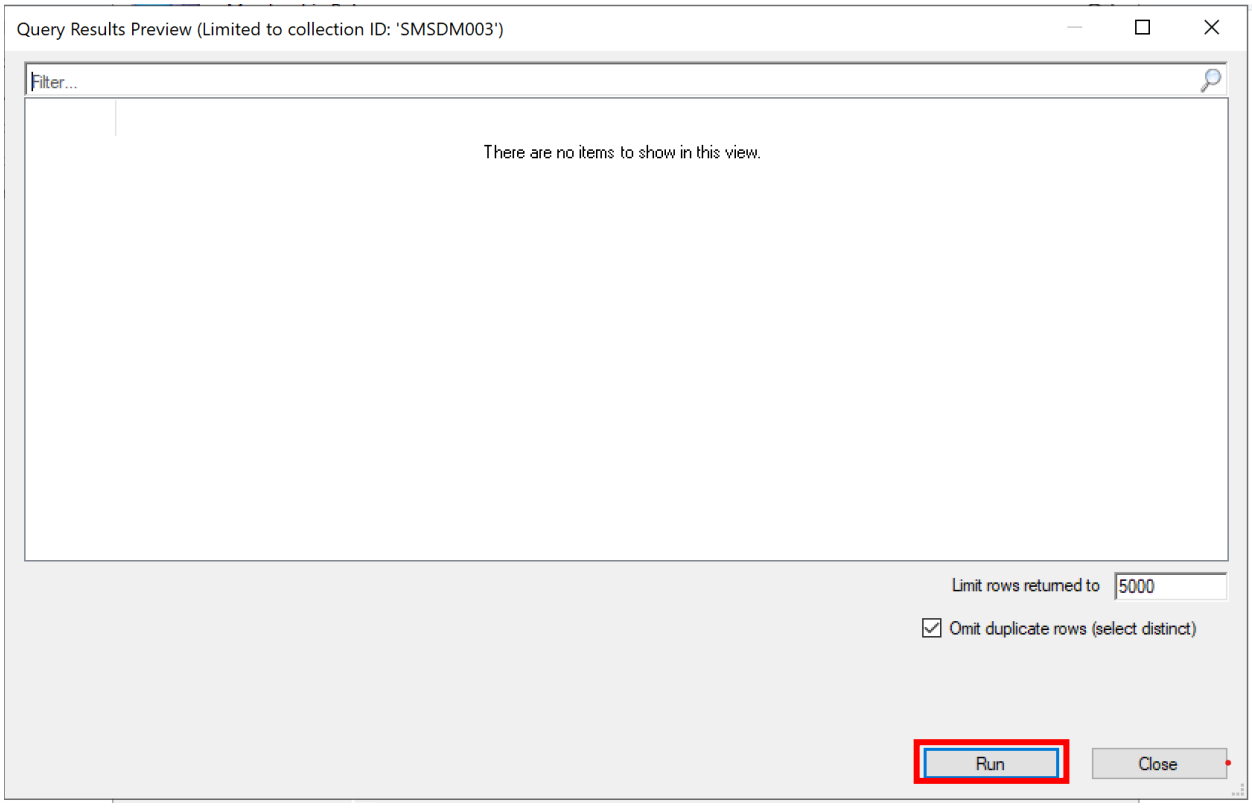
**Example query 3 - All Windows Server (2012, 2016, 2019 and 2022)**

select SMS_R_SYSTEM.ResourceID,SMS_R_SYSTEM.ResourceType,SMS_R_SYSTEM.Name,SMS_R_SYSTEM.SMSUniqueIdentifier,SMS_R_SYSTEM.ResourceDomainORWorkgroup,SMS_R_SYSTEM.Client from SMS_R_System where SMS_R_System.OperatingSystemNameandVersion like "%Server%"

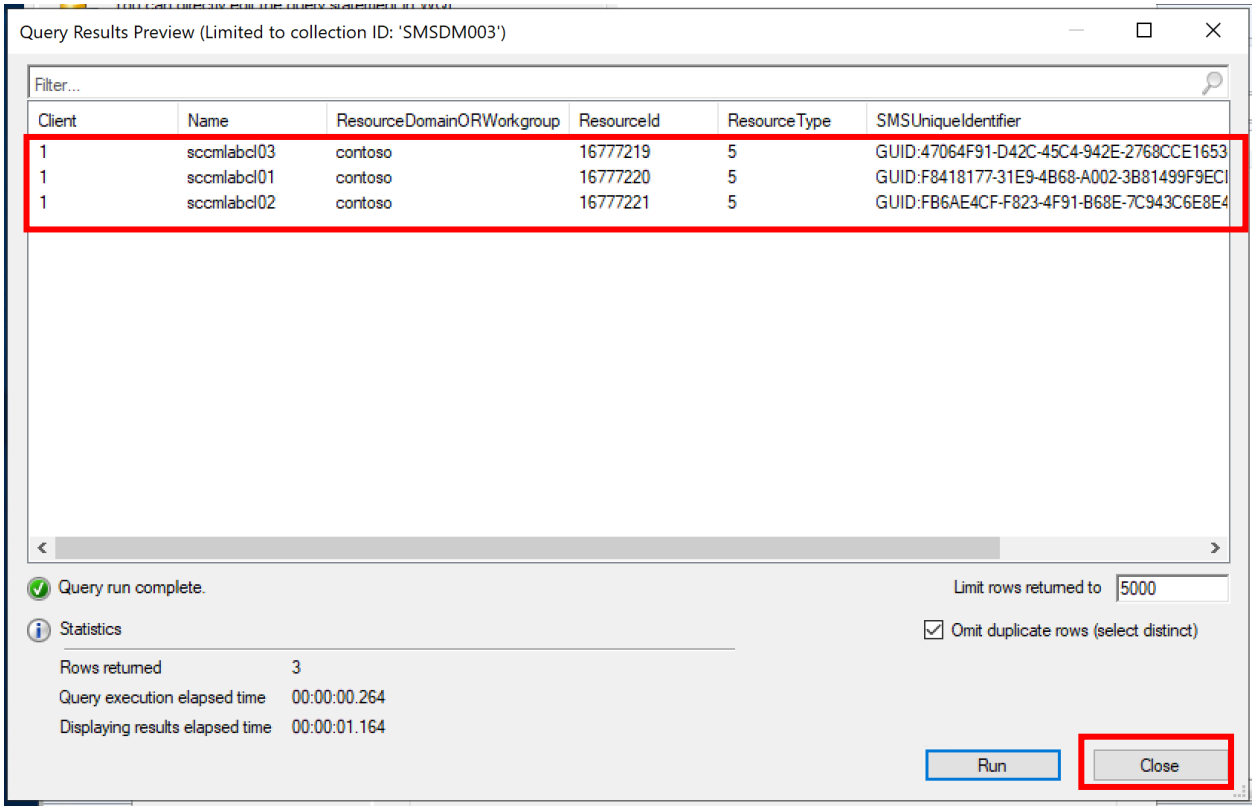
**Example query 4 - All Desktops (Windows 10, Windows 11)**

select SMS_R_SYSTEM.ResourceID,SMS_R_SYSTEM.ResourceType,SMS_R_SYSTEM.Name,SMS_R_SYSTEM.SMSUniqueIdentifier,SMS_R_SYSTEM.ResourceDomainORWorkgroup,SMS_R_SYSTEM.Client from SMS_R_System where SMS_R_System.OperatingSystemNameandVersion like "%Workstation%"

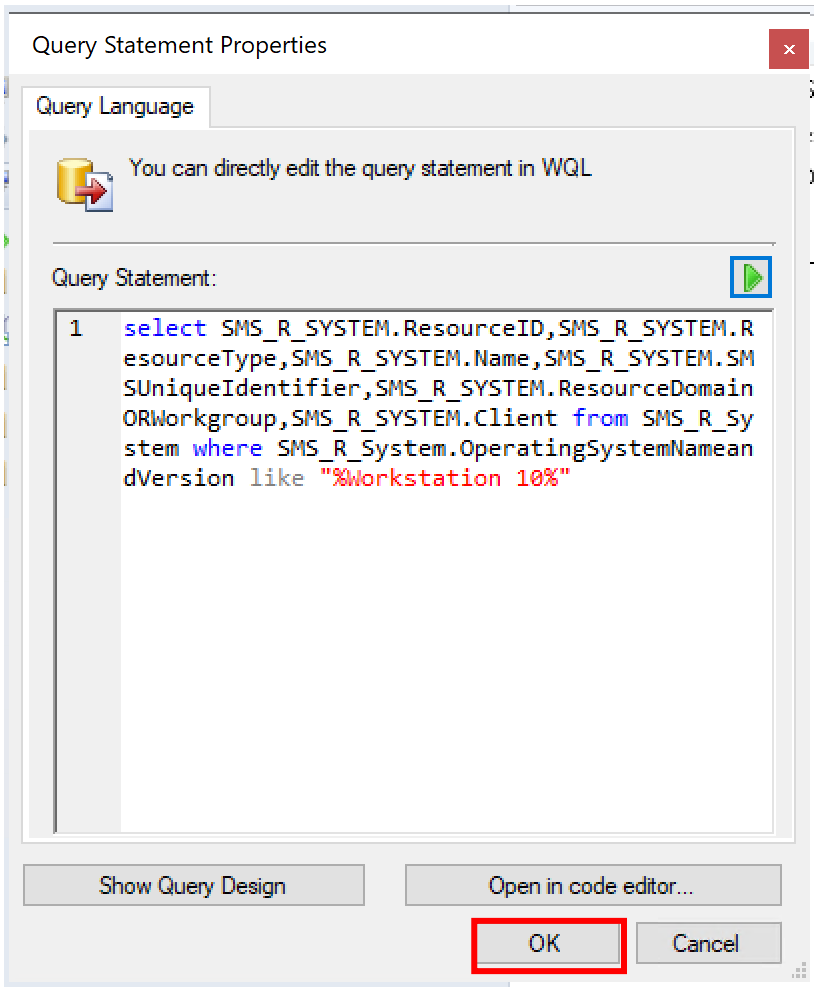
13. Select **Run**



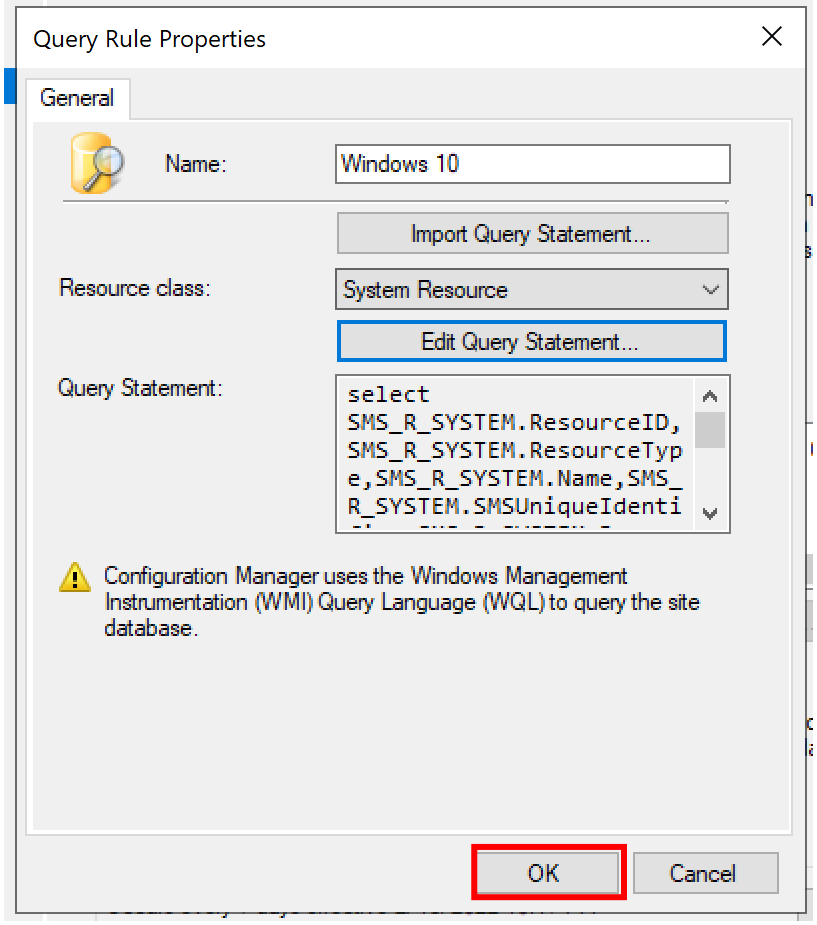
14. Confirm all the correct systems are present for the Operating system you want to onboard → **Close**



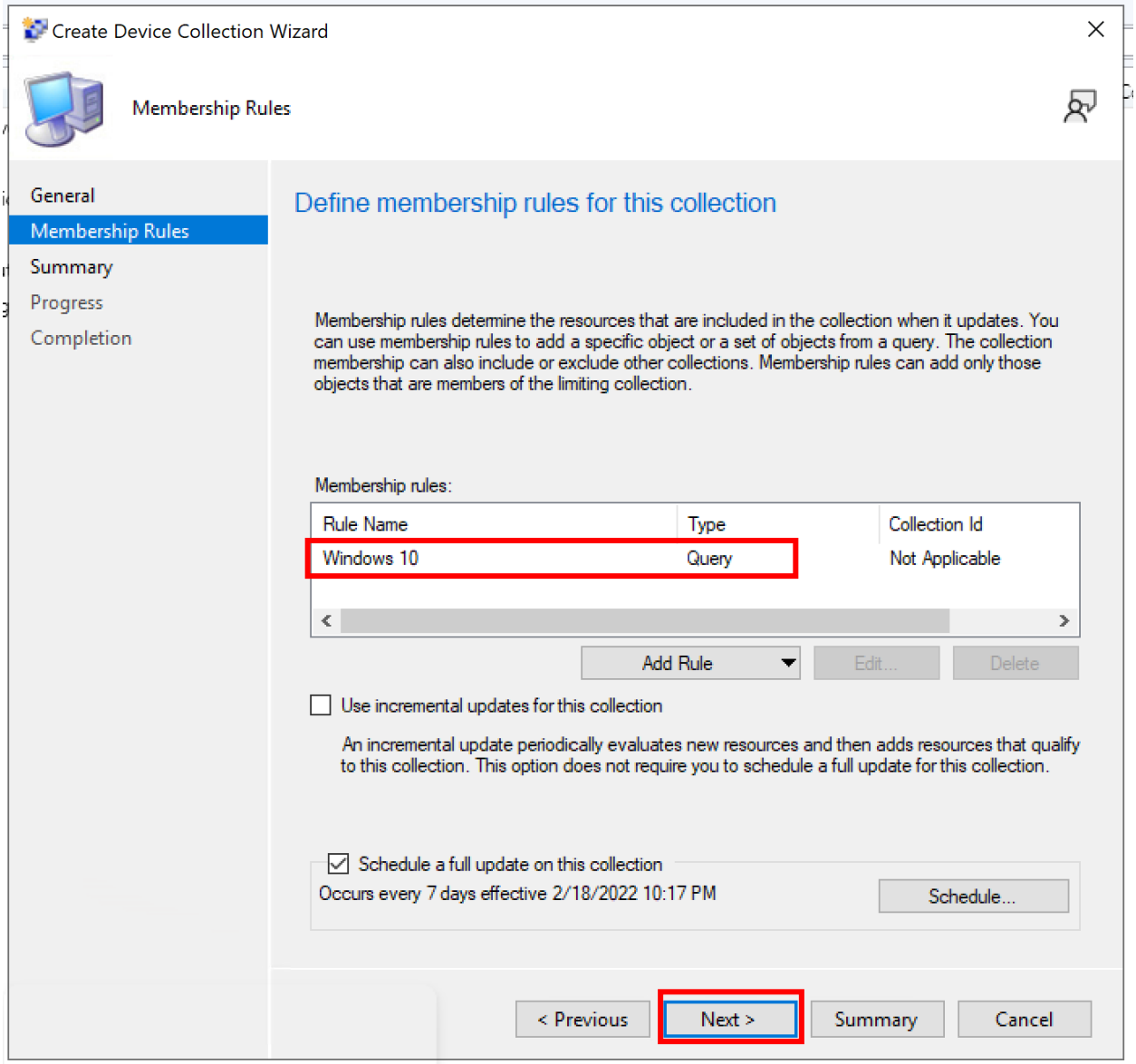

15. Select **OK**


Query Statement Properties
Query Language
You can directly edit the query statement in WQL
Query Statement:
1 select SMS_R_SYSTEM.ResourceID,SMS_R_SYSTEM.Resourcetype,SMS_R_SYSTEM.Name,SMS_R_SYSTEM.SMSUniqueIdentifier,SMS_R_SYSTEM.ResourceDomainORWorkgroup,SMS_R_SYSTEM.Client from SMS_R_System where SMS_R_System.OperatingSystemNameandVersion like "%Workstation 10%"
Show Query Design
Open in code editor...
OK
Cancel

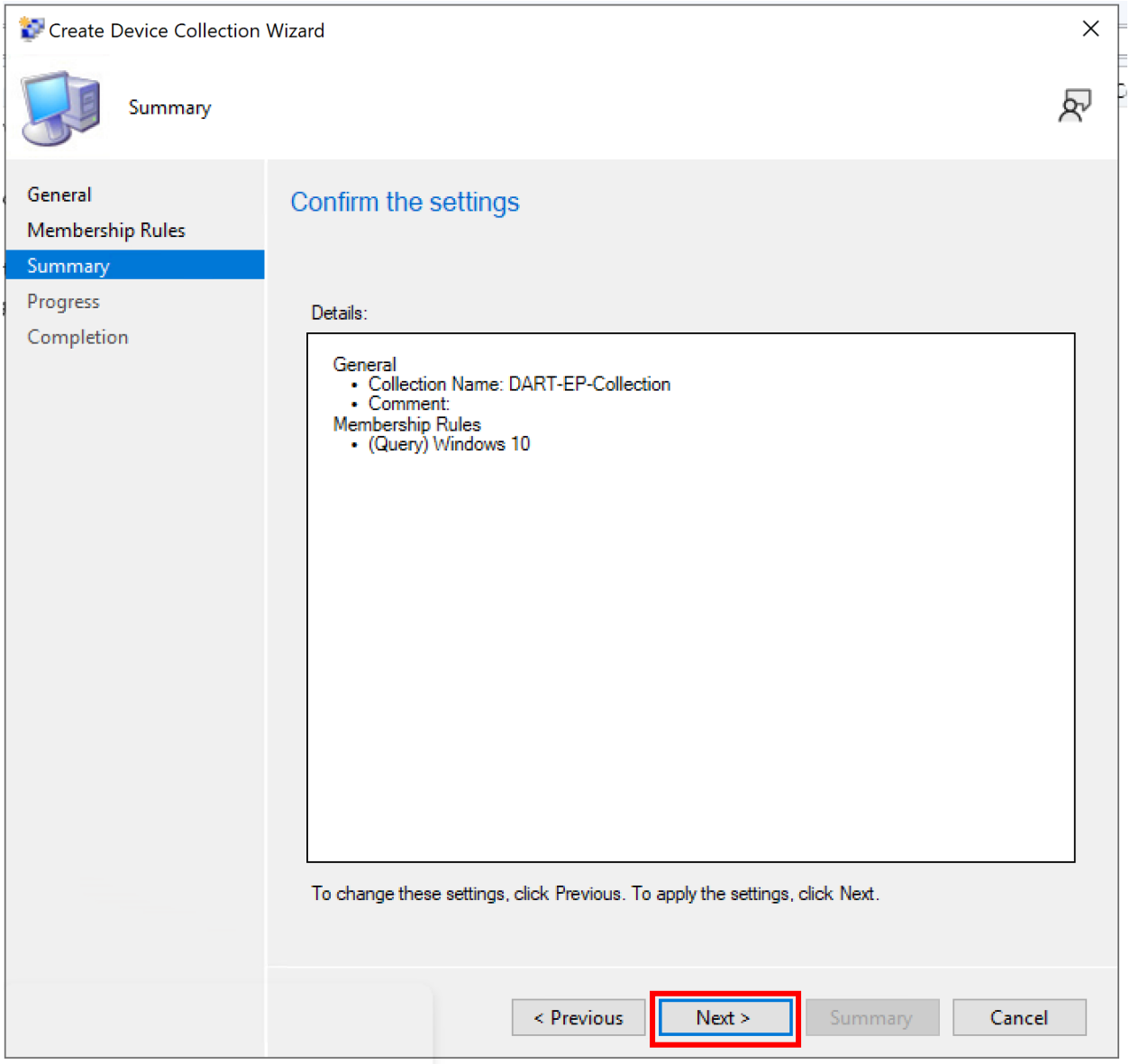
16. Select **OK**



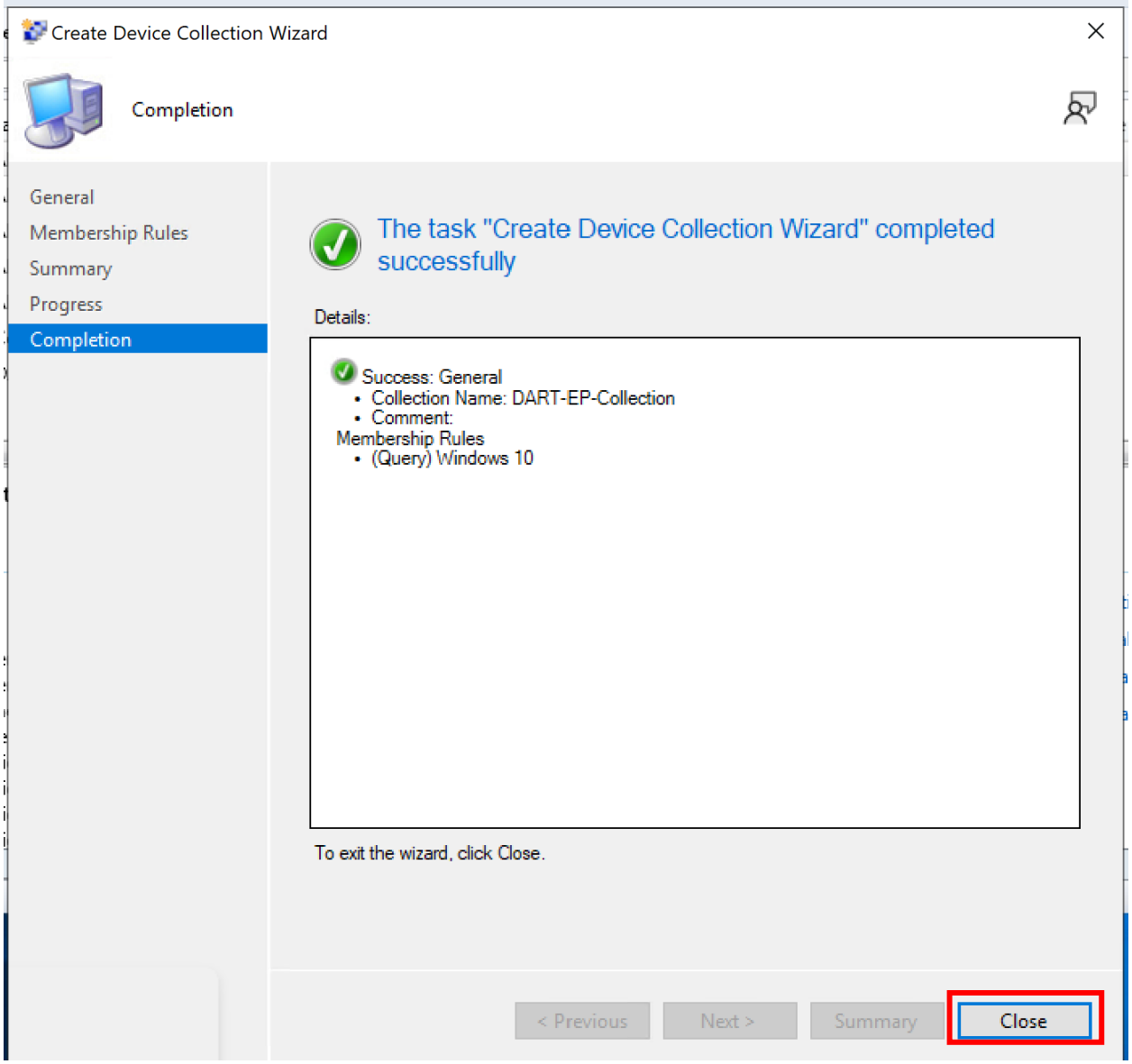

17. Select **Next**



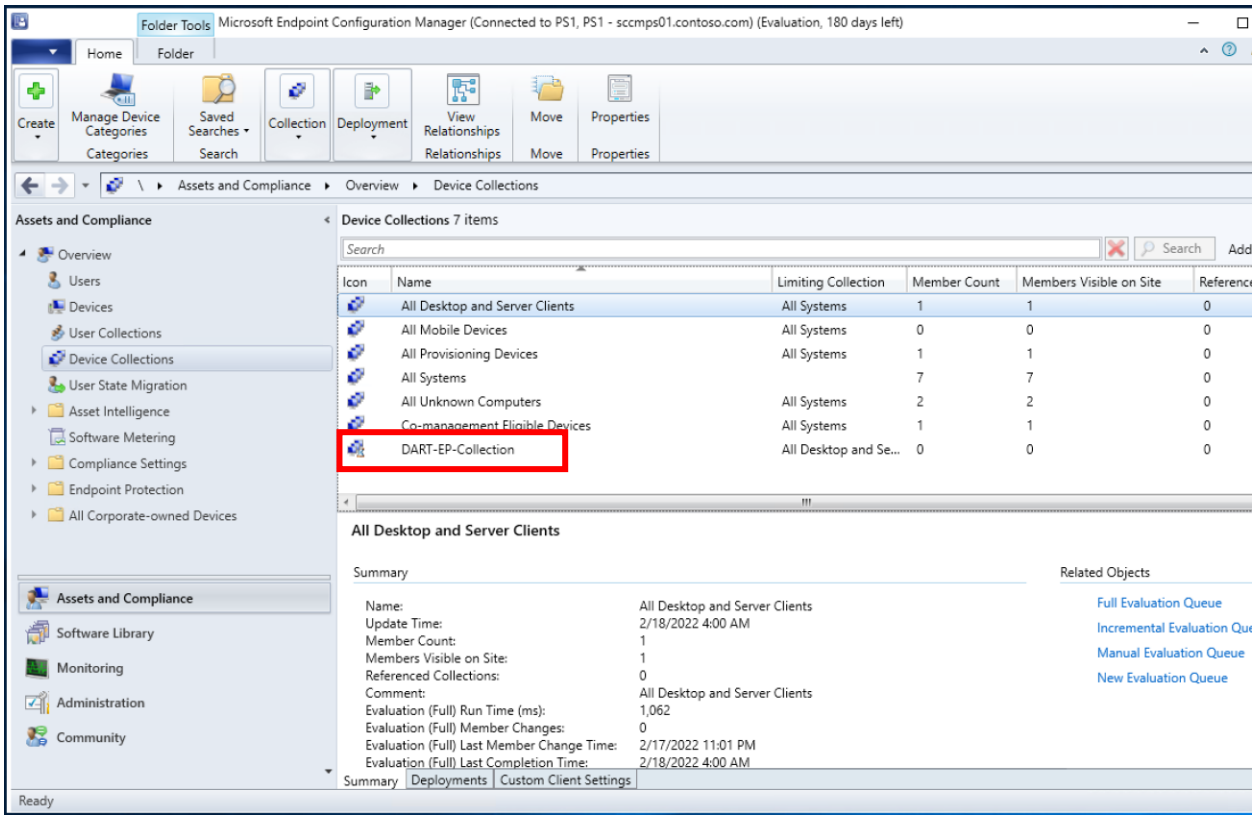

18. Select **Next**



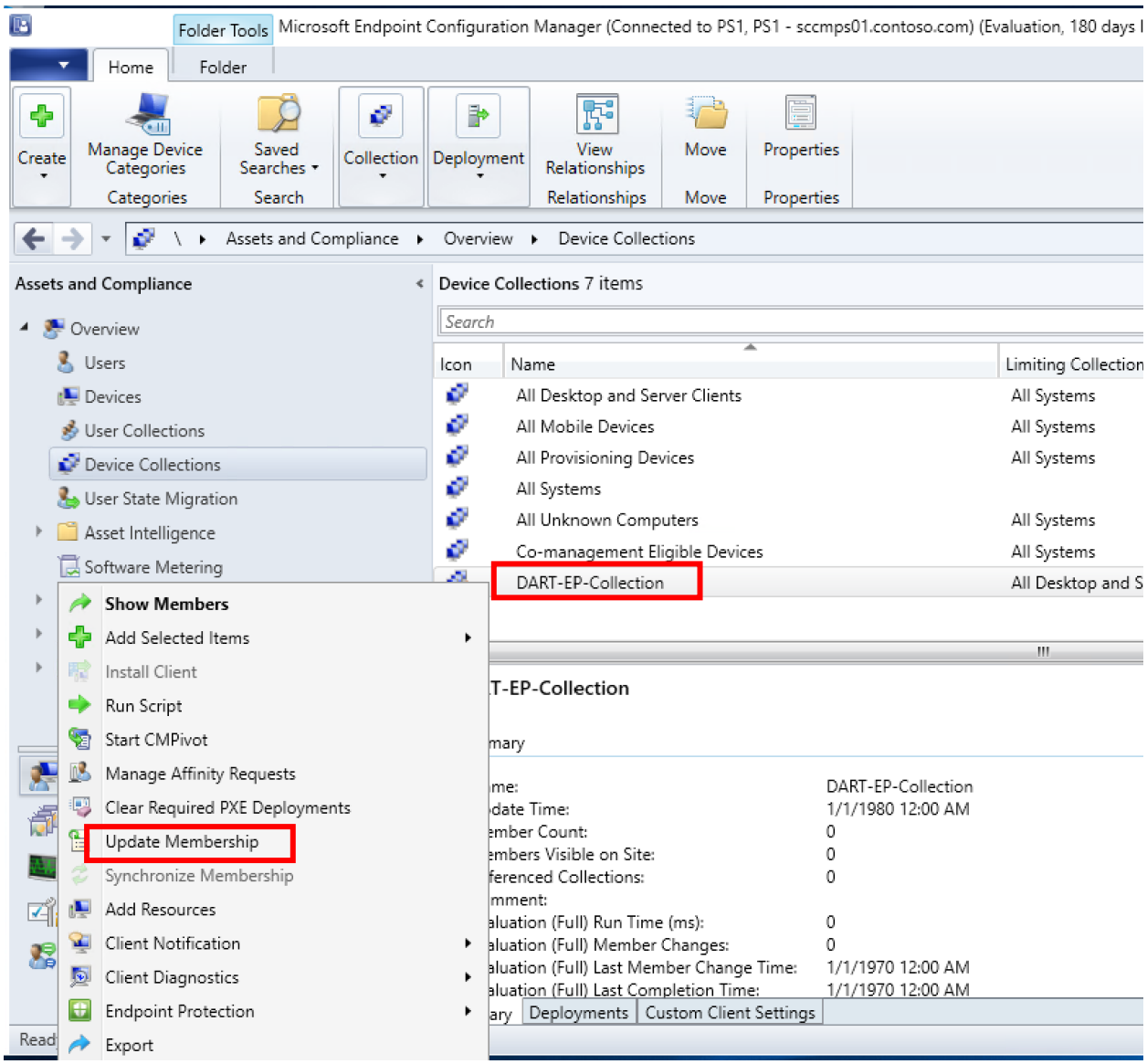

19. Select **Close**



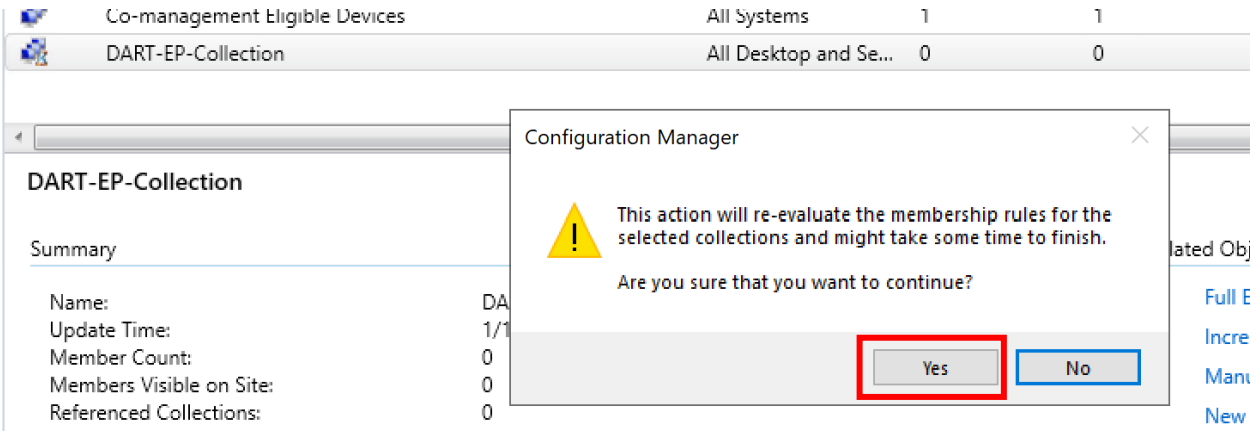

20. Now you should have the newly created SCCM collection




21. Right click Collection → **Update membership**

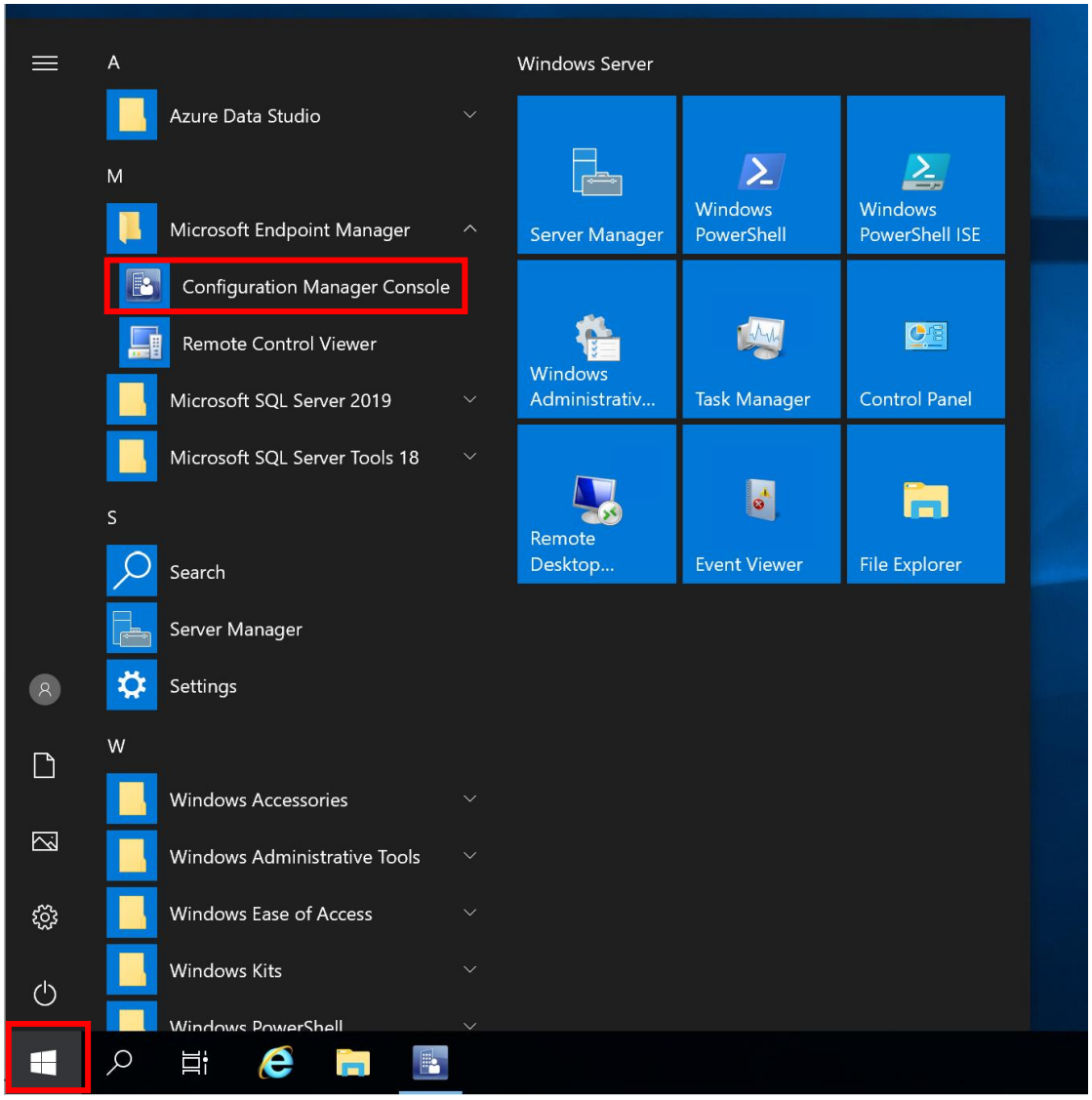



22. Select **Yes**

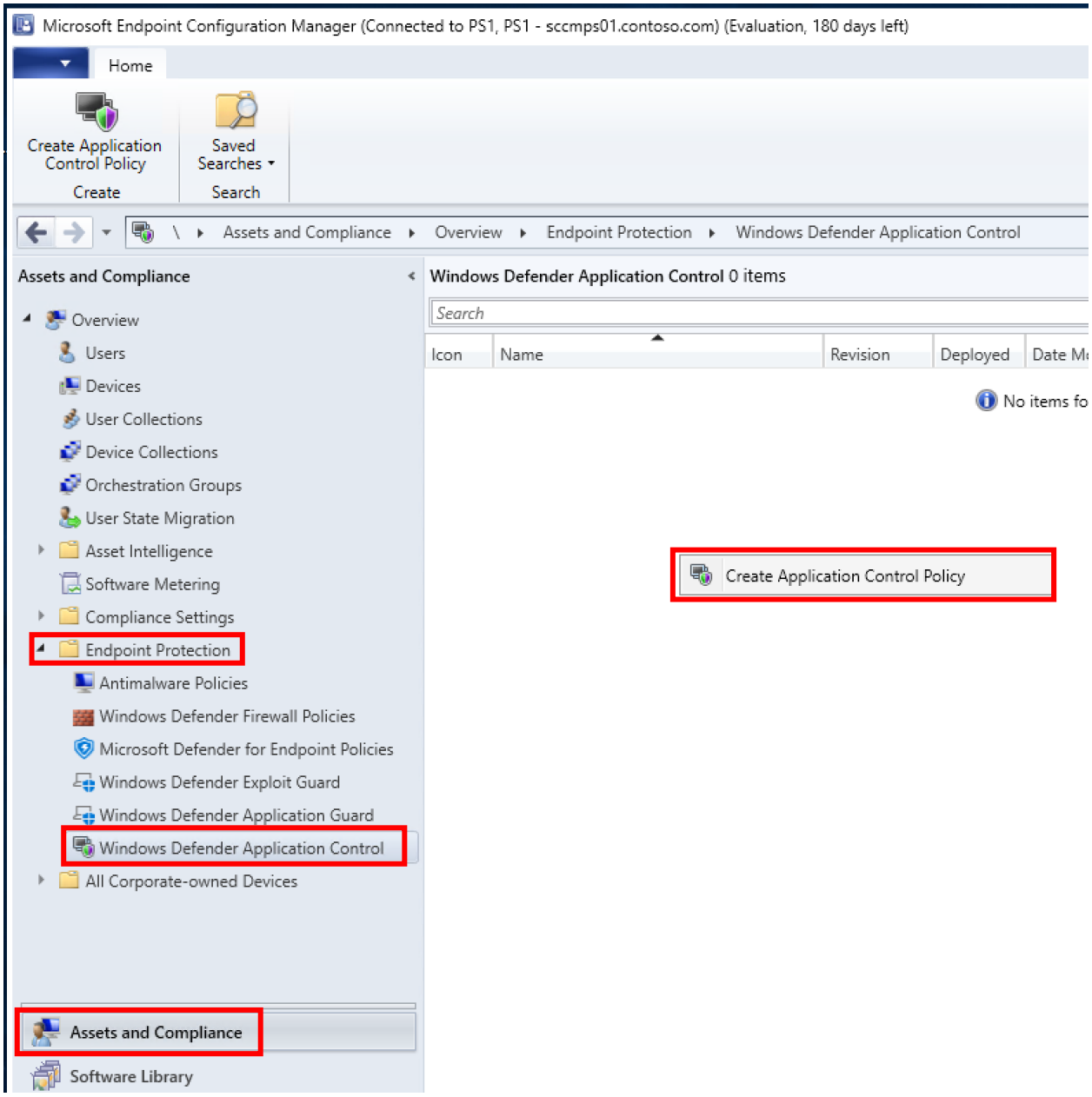


## Create WDAC Policy

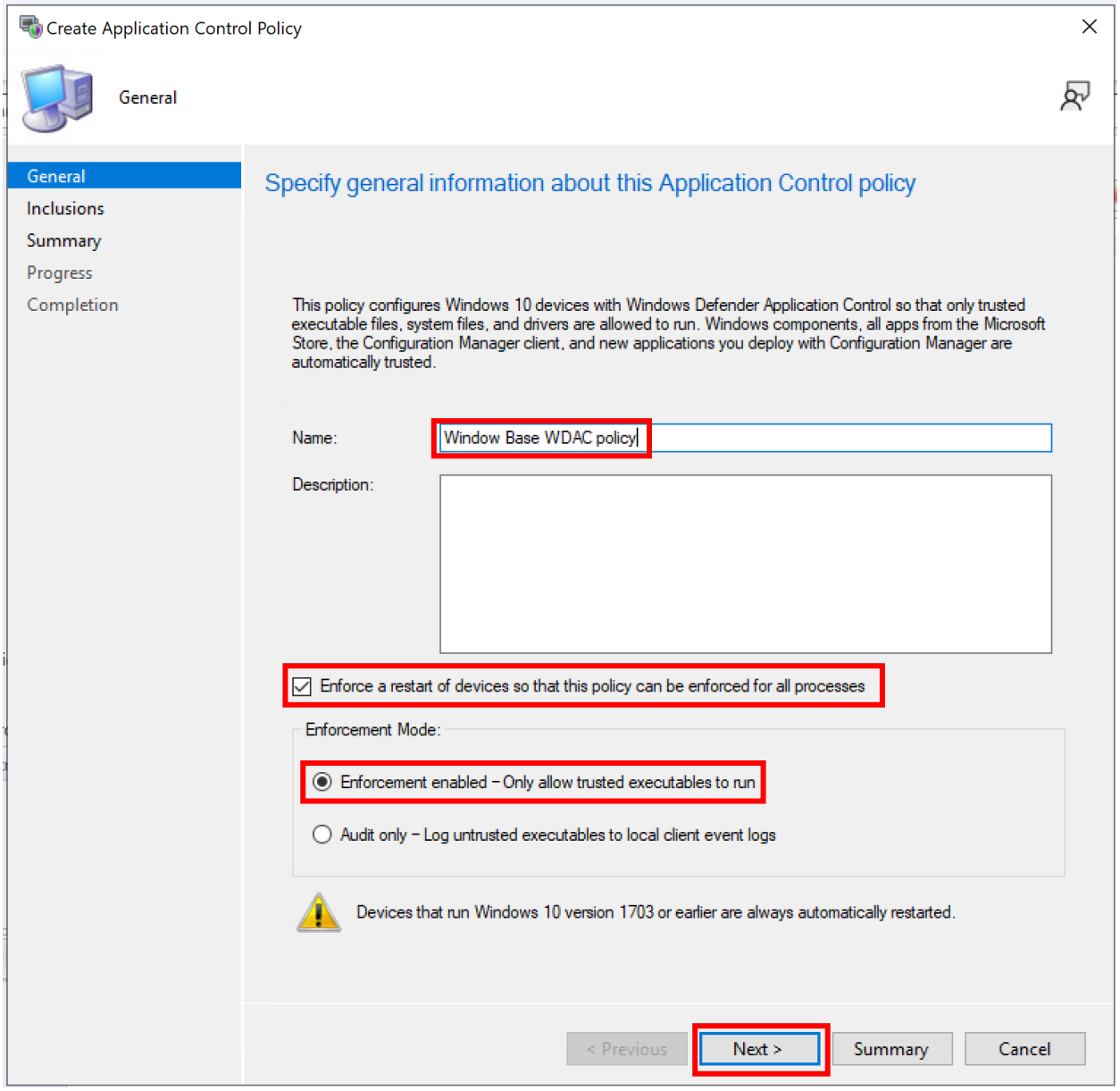
23. Log into the Primary Site system
24. Open the



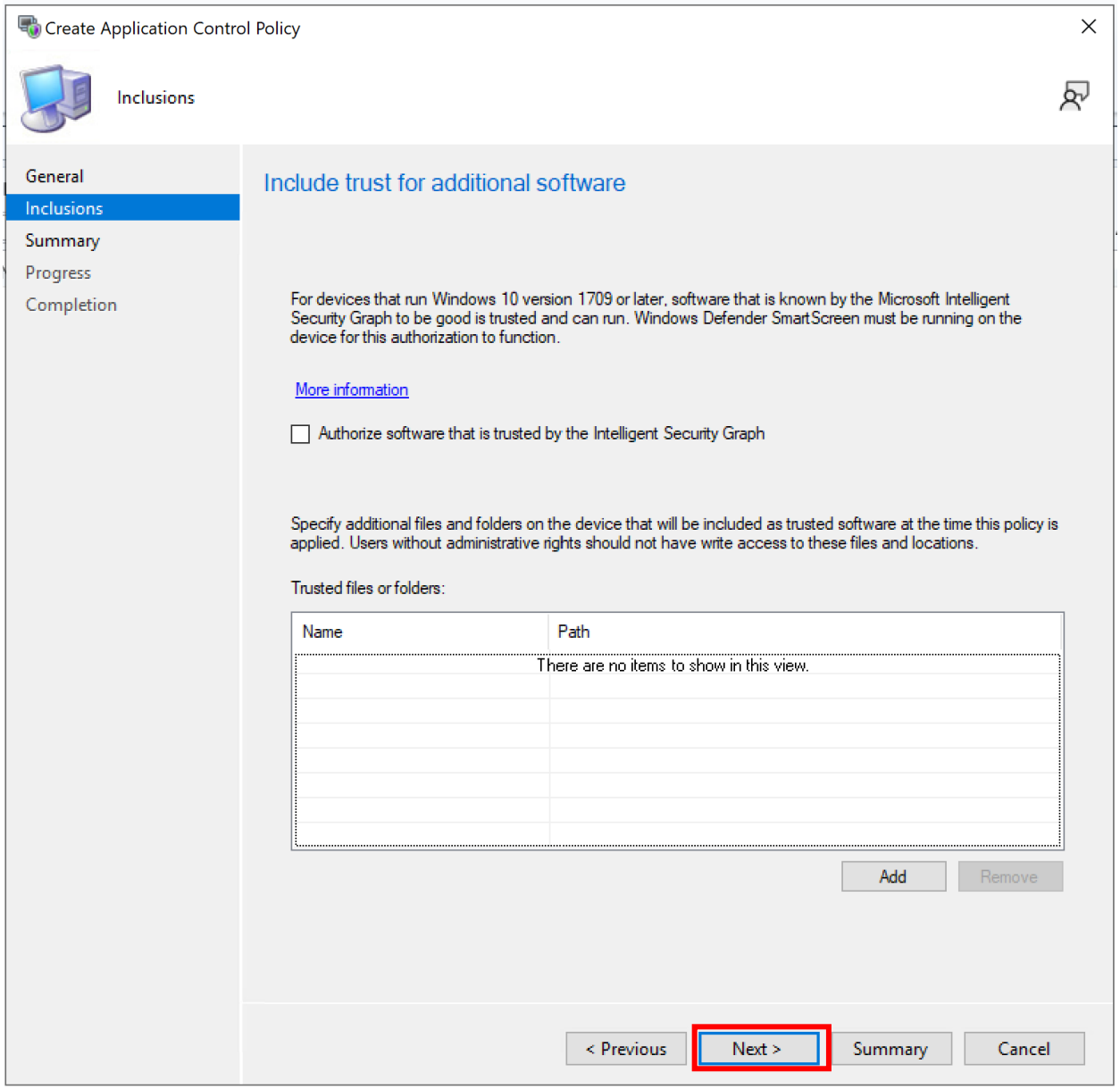

25. Select **Asset and Compliance → Endpoint Protection → Windows Defender Application Control → Create Application Control Policy**



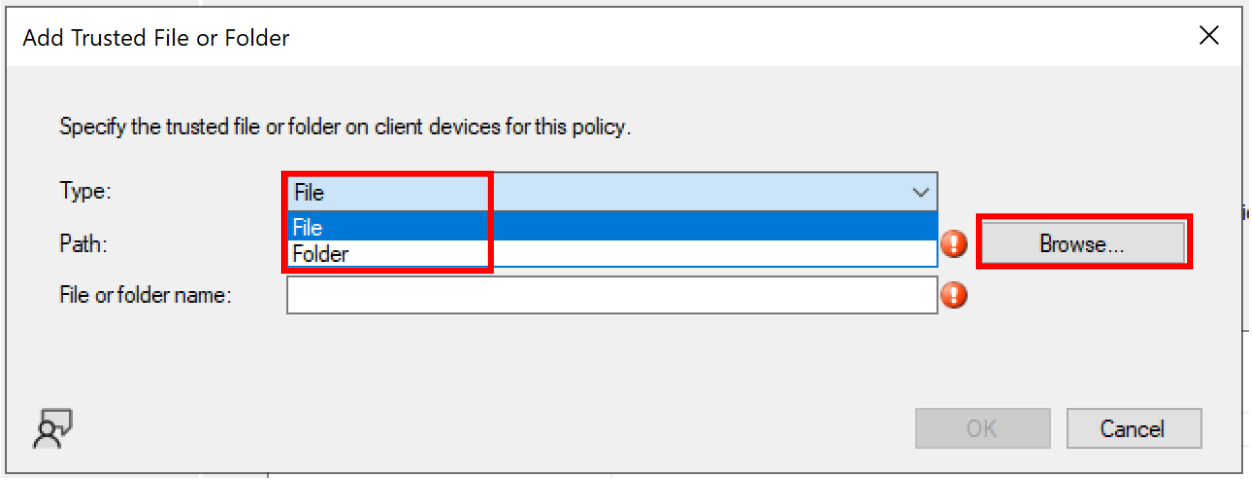

26. Enter the name of the policy → **Next**
27. Enable **Enforce a restart of devices so that this policy can be enforced for all processes**
28. Select the mode which you want the policy to run (Enforce enabled / Audit Only)
29. Click **Next**


Create Application Control Policy
General
General
Inclusions
Summary
Progress
Completion
Specify general information about this Application Control policy
This policy configures Windows 10 devices with Windows Defender Application Control so that only trusted executable files, system files, and drivers are allowed to run. Windows components, all apps from the Microsoft Store, the Configuration Manager client, and new applications you deploy with Configuration Manager are automatically trusted.
Name:
Window Base WDAC policy
Description:
Enforce a restart of devices so that this policy can be enforced for all processes
Enforcement Mode:
Enforcement enabled – Only allow trusted executables to run
Audit only – Log untrusted executables to local client event logs
Devices that run Windows 10 version 1703 or earlier are always automatically restarted.
< Previous
Next >
Summary
Cancel

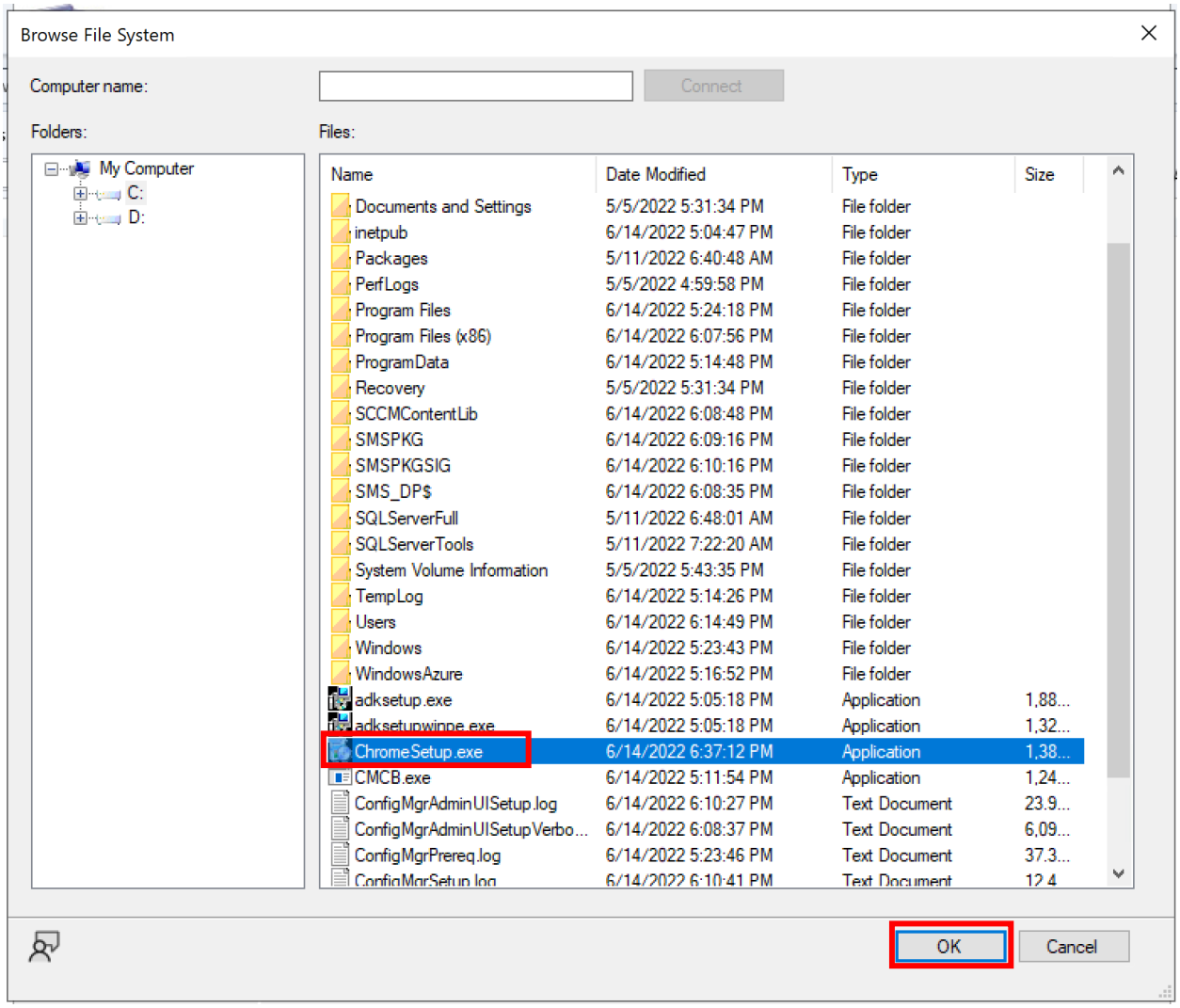
30. Click **Add**



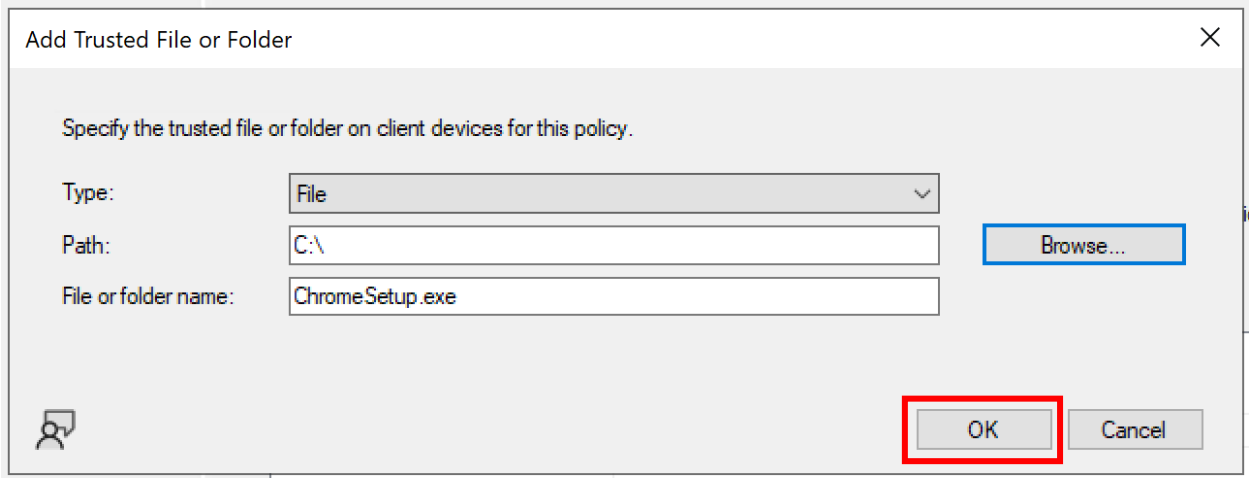

31. Select **File** or **Folder** → **Browse**



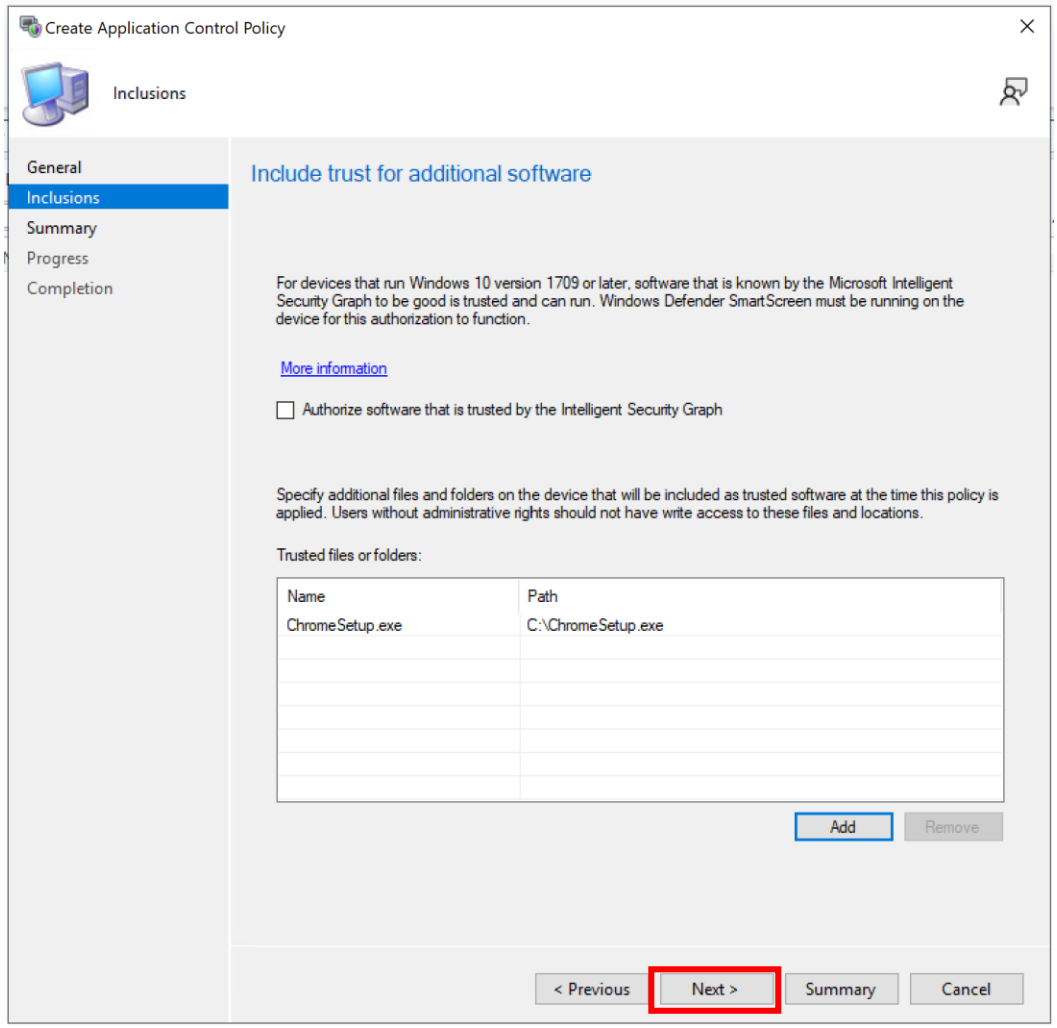

32. Select the executable for your policy → **OK**



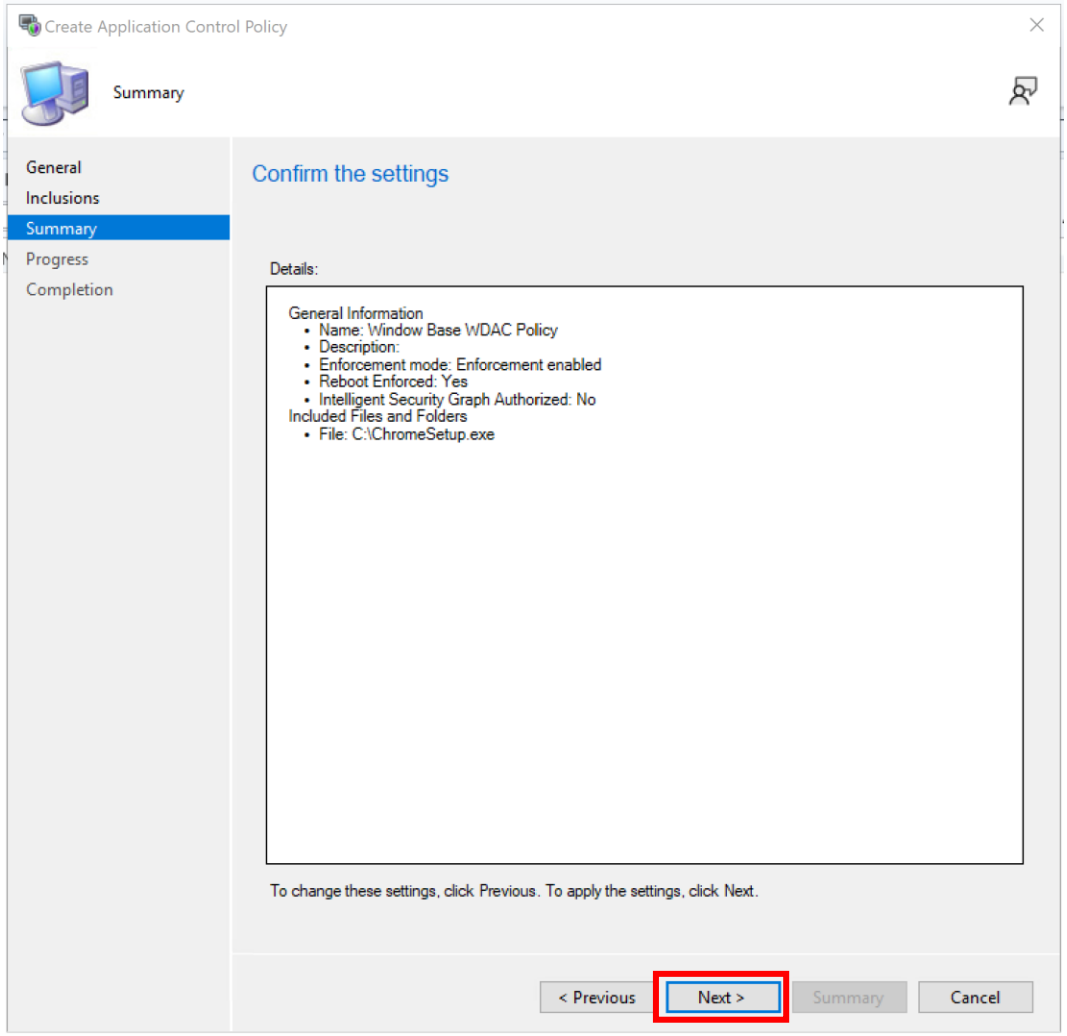
33. Select **OK**



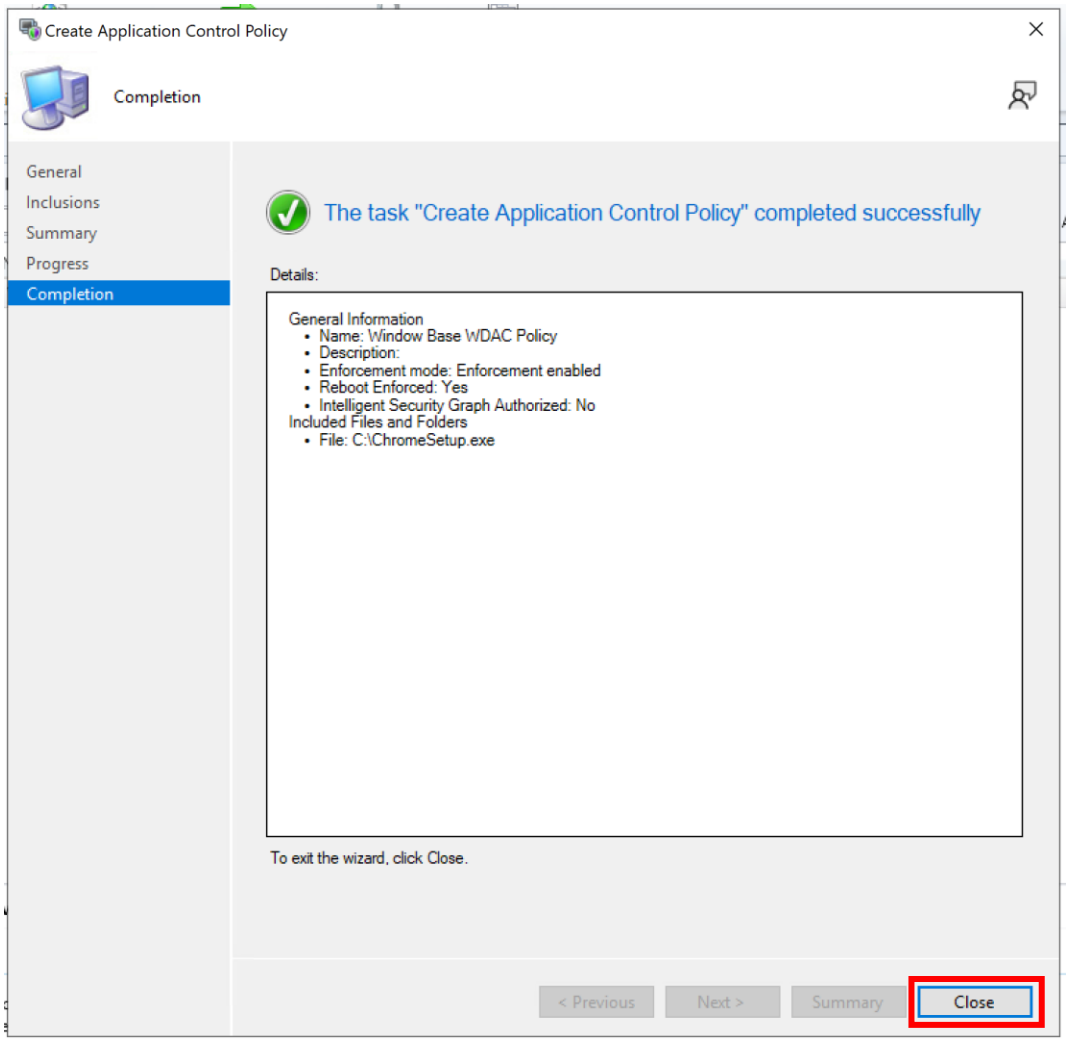

34. Select **Next**



35. Select **Next**

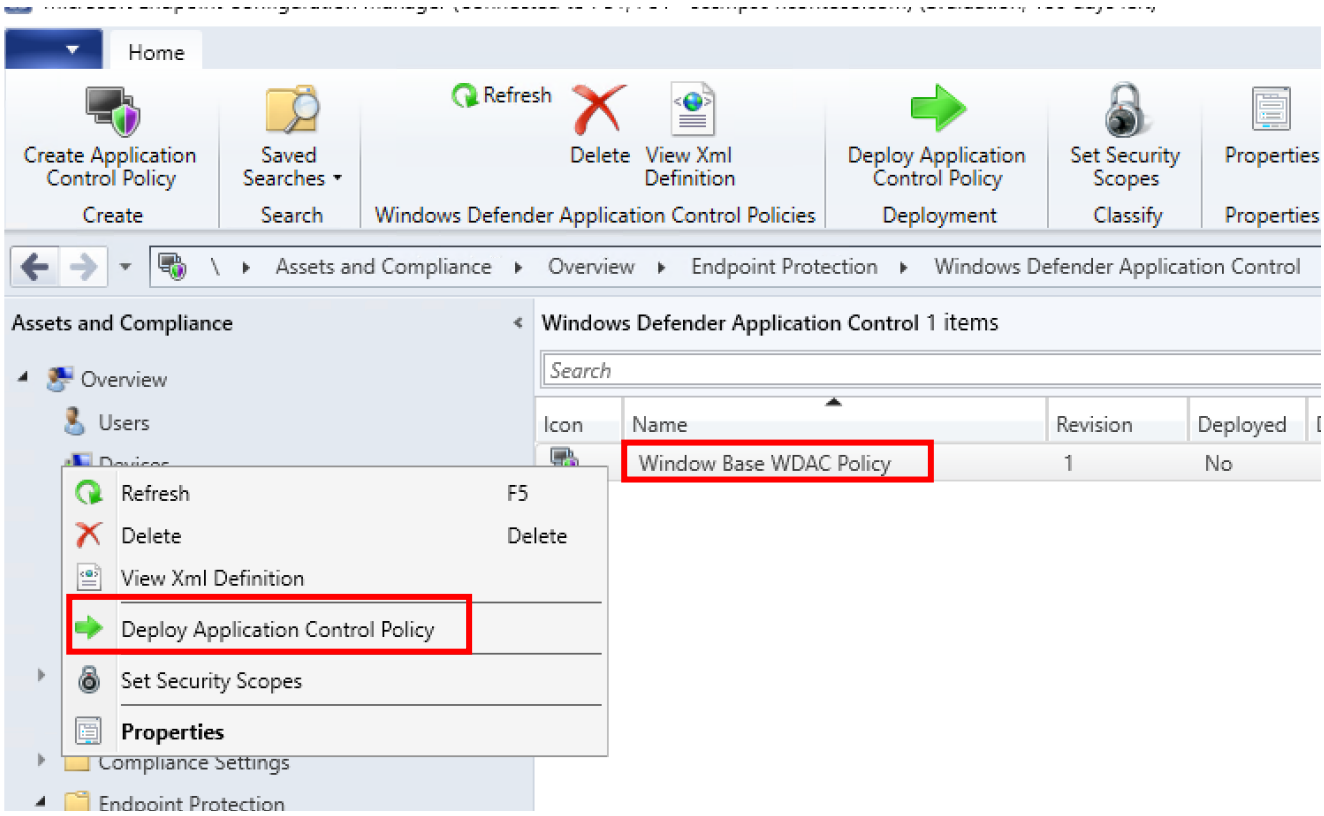



36. Select **Close**

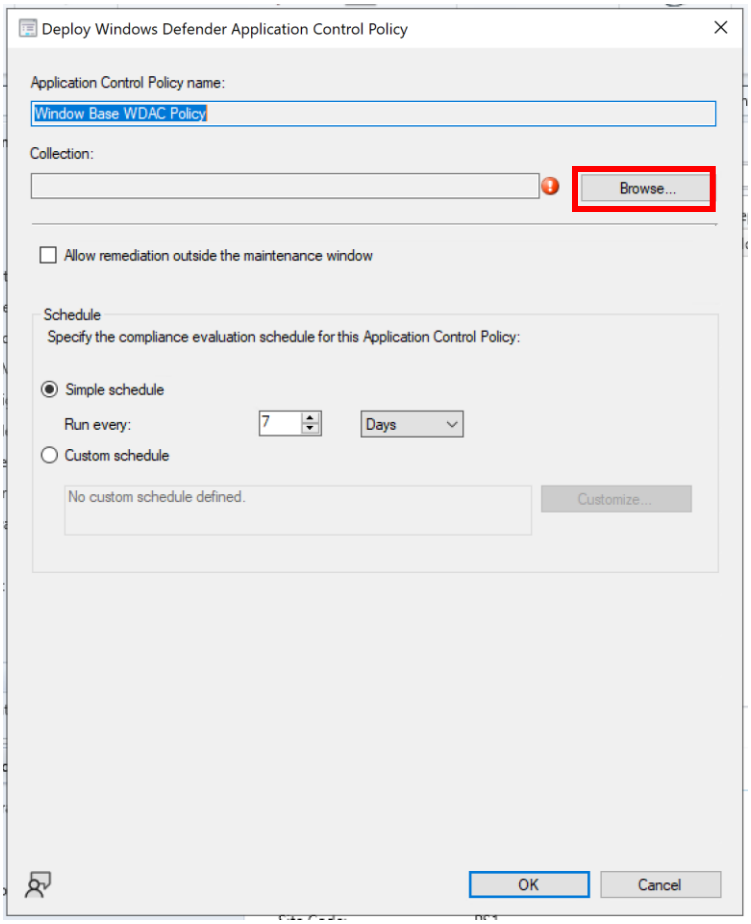



## Deploy WDAC Policy to Device Collection

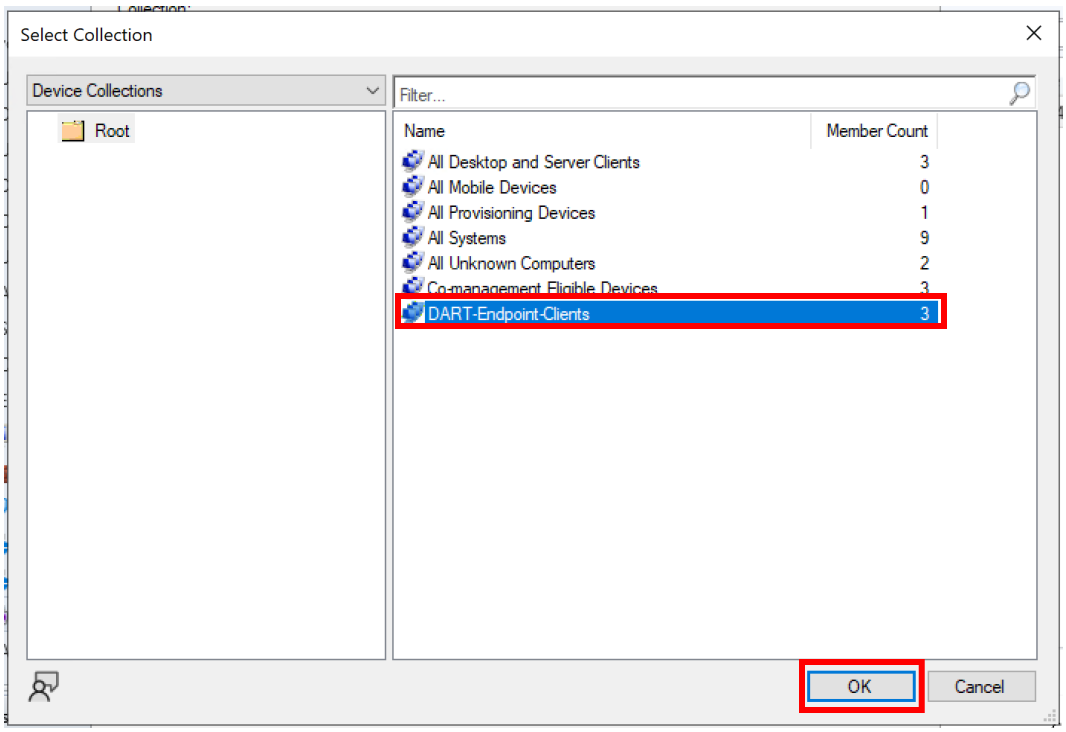
1. Right-click the newly created policy → **Deploy Application Control Policy**




2. Select **Browse**




3. Select the Device Collection you created earlier → **OK**



4. Change the Schedule → **OK**