

AQUILA EDR Mass Deployment Via Microsoft Intune (EXE)

Overview

This guide explains how to install an Endpoint Detection & Response (EDR) solution on all devices managed through **Microsoft Intune**. The process ensures consistent protection across your organization's endpoints by using Intune's **Endpoint security policies** and **app deployment** features.

Prerequisites

Before beginning, confirm the following:

- You have **Global Administrator** or **Intune Administrator** rights in Microsoft 365.
 - Devices are already **enrolled and compliant** in Microsoft Intune.
 - You have the **installer package (IntuneWin format)** for your chosen EDR solution.
-

Step 1: Prepare the EDR Installer

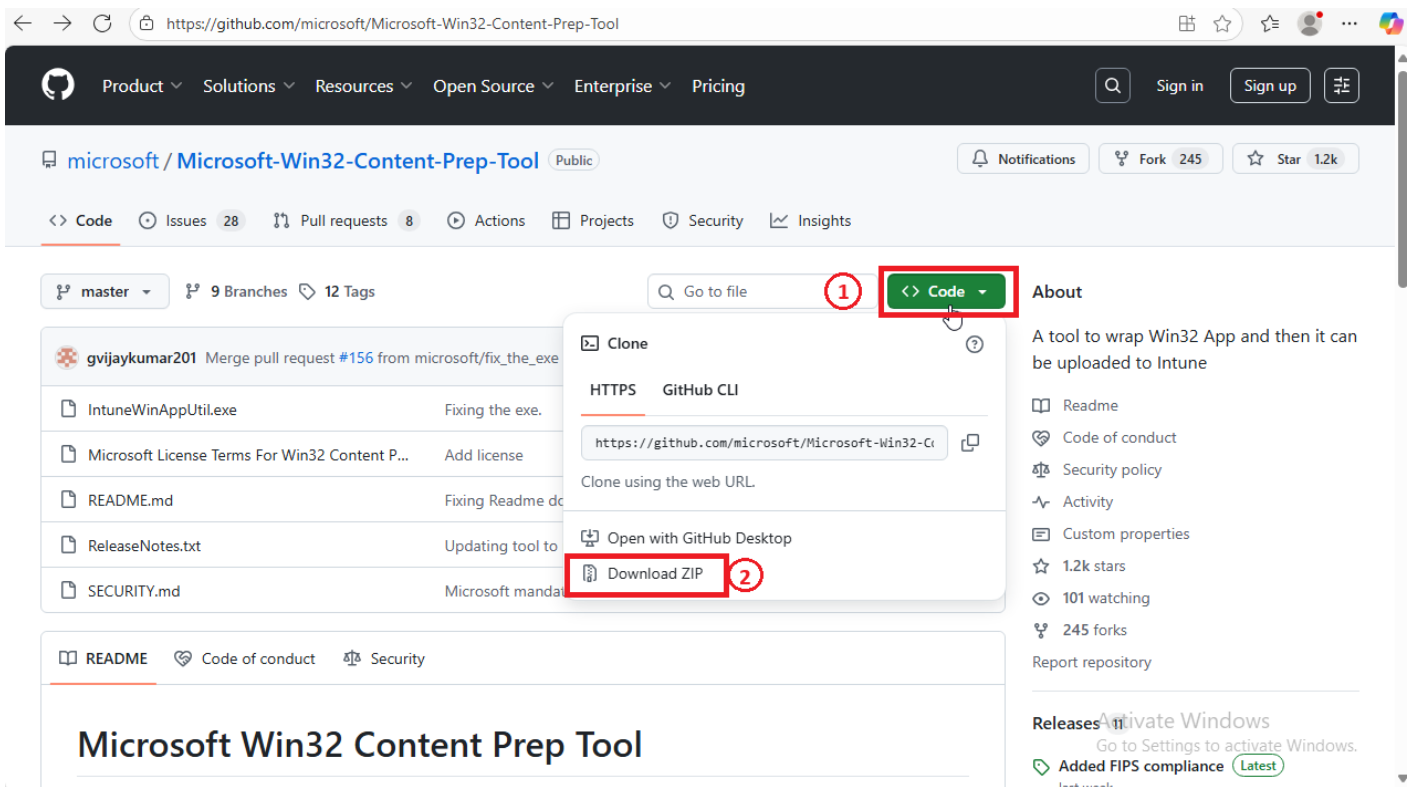
1. Obtain the official EDR installation package (MSI) from AQUILA
2. If the installer is not in **.intunewin** format, convert it using the **Microsoft Win32 Content Prep Tool**.

Download tool: [*GitHub - microsoft/Microsoft-Win32-Content-Prep-Tool: A tool to wrap Win32 App and then it can be uploaded to Intune*](#)

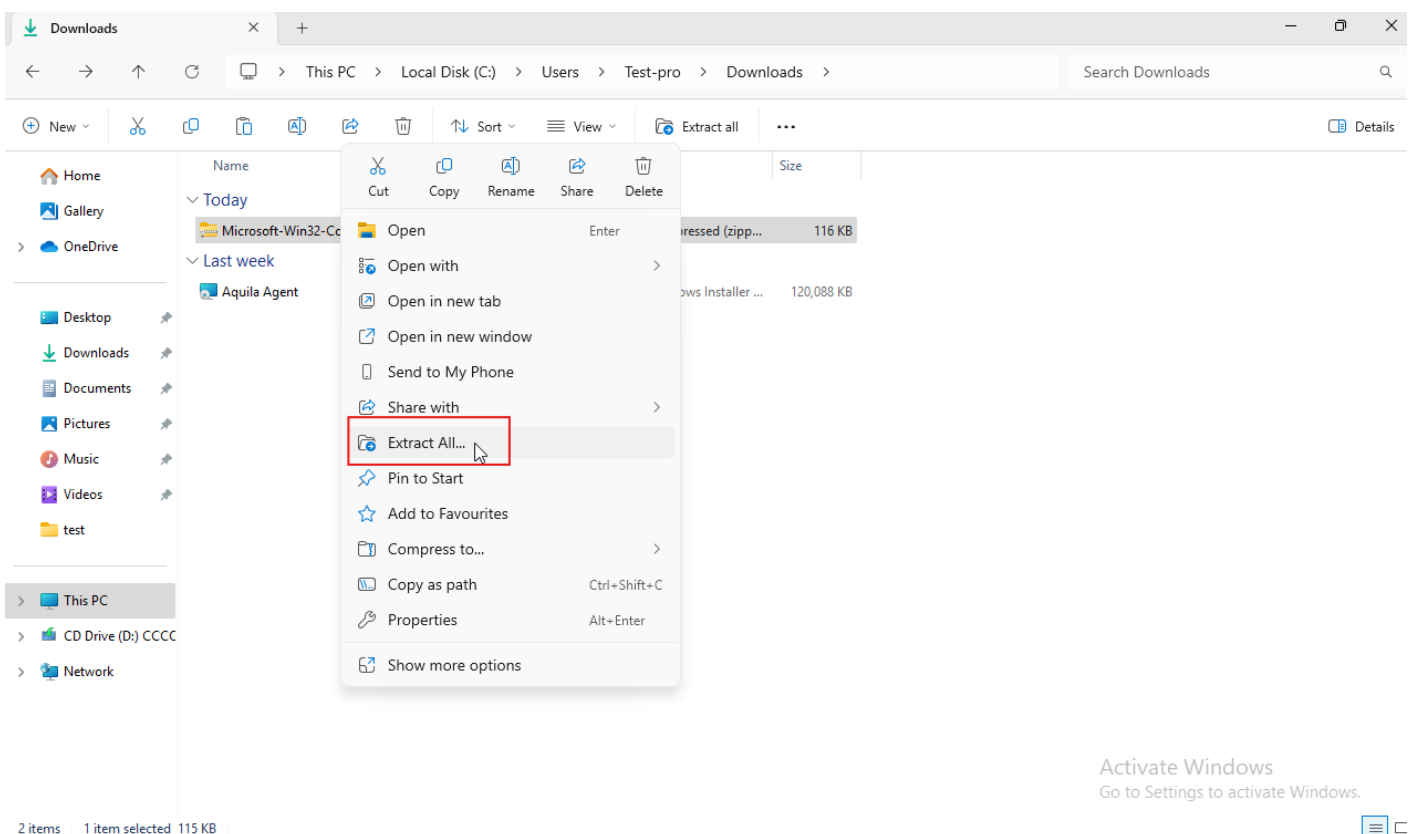
Steps for preparing an Installer for Intune (.intunewin format)

Download the Packaging Tool

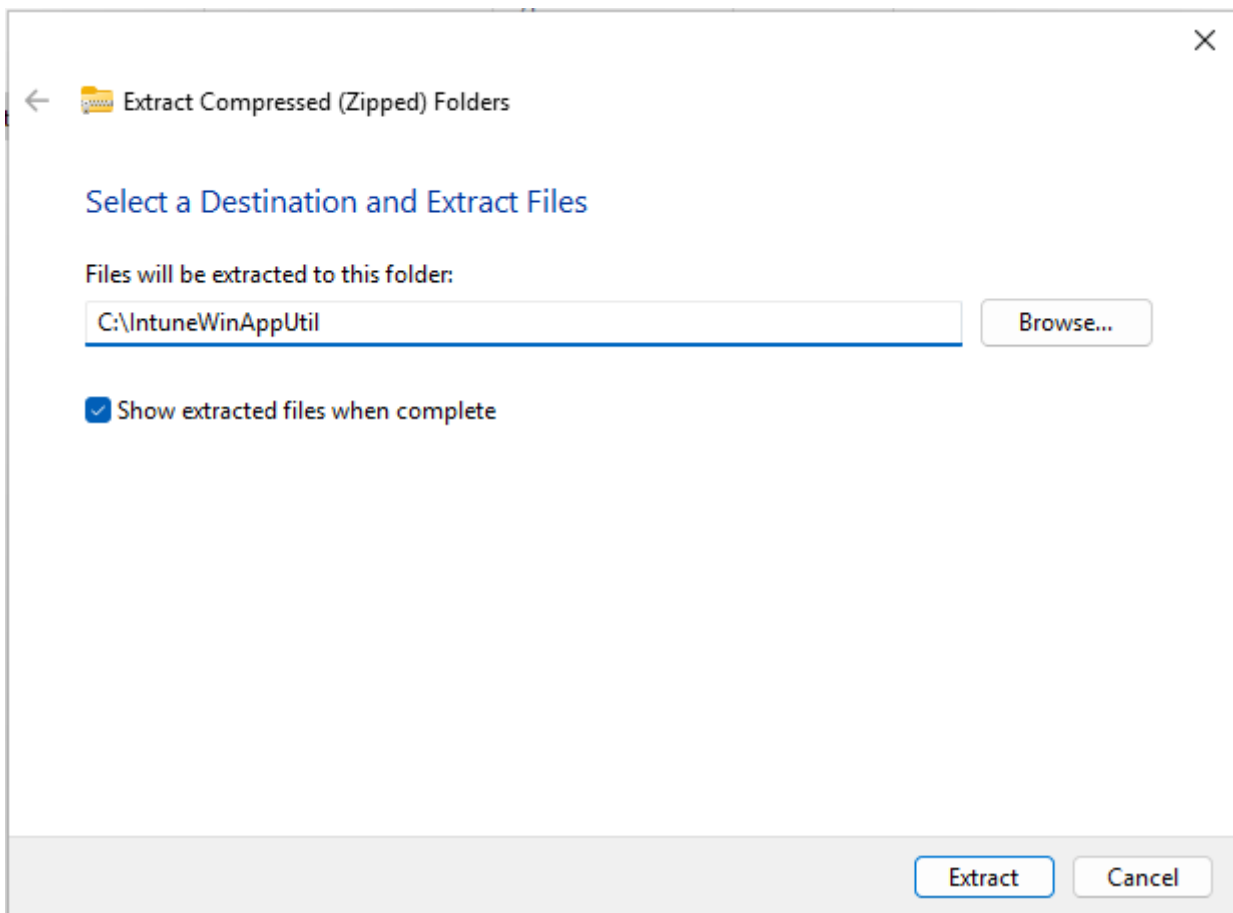
1. Go to Microsoft's official download page: [Win32 Content Prep Tool \(GitHub\)](#)
2. Download the ZIP file to your computer.



3. Right-click the ZIP → **Extract All...**

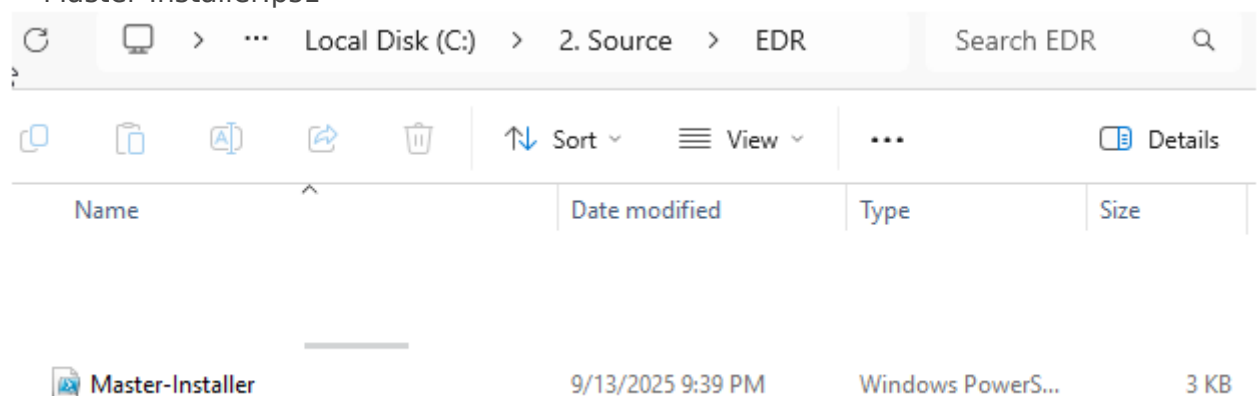


4. Choose a location (for example: `C:\IntuneWinAppUtil`).



Prepare Your Installer Files

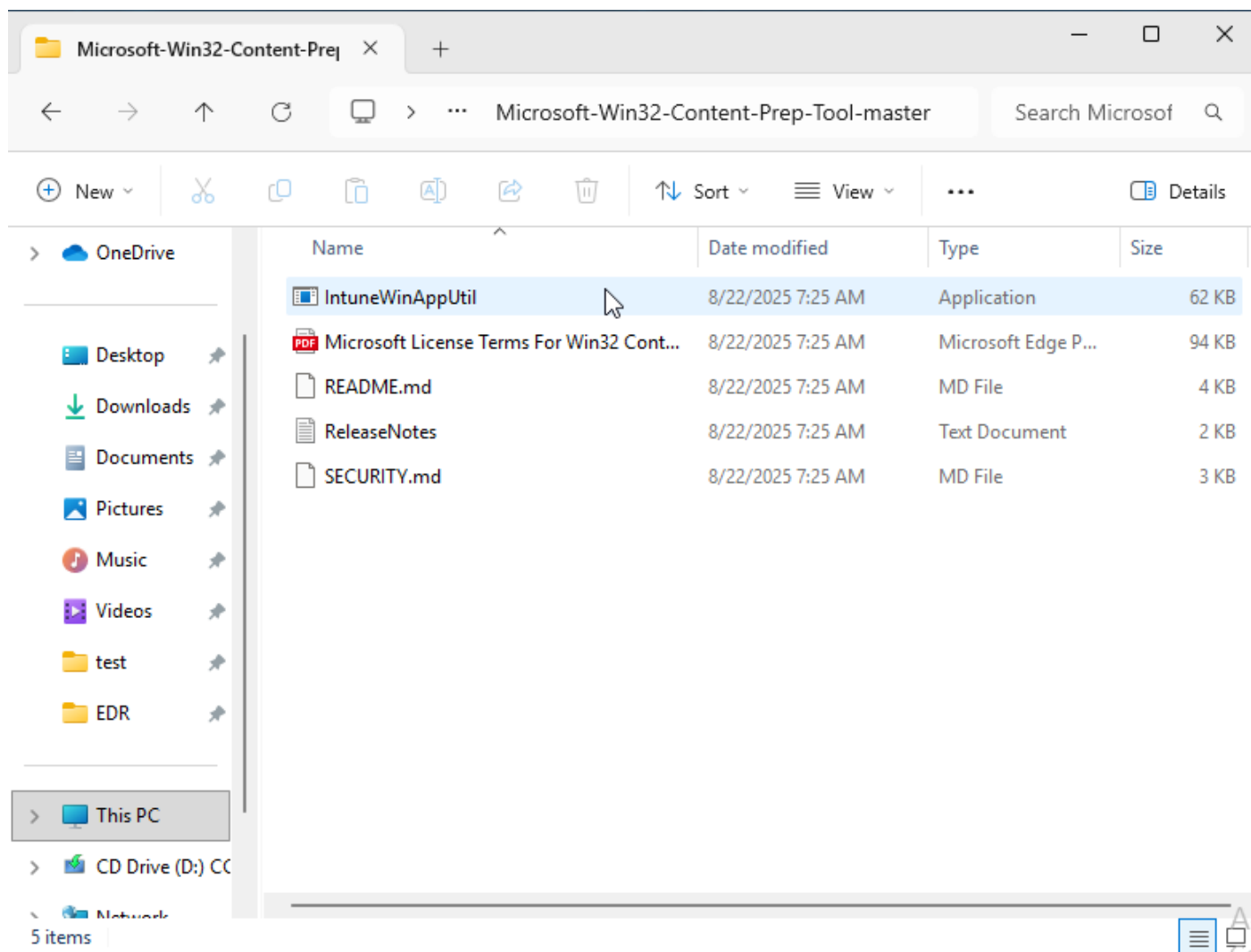
1. Create a folder for your installer, for example:
 - C:\2. Source\EDR
2. Place the Master installer inside that folder. Place also here the script files.
 - Master-Installer.ps1



3. Create another empty folder where the packaged file will be saved, for example:
 - C:\3. Output

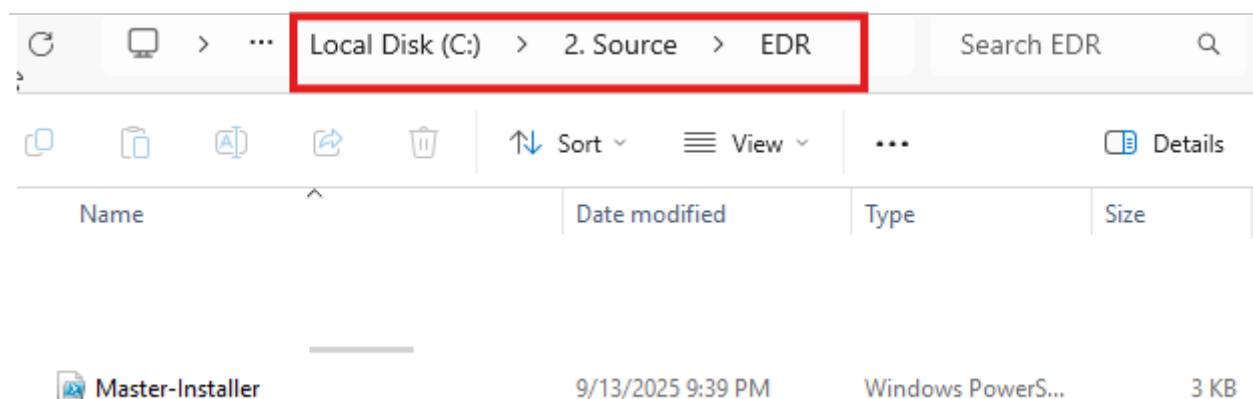
Run the Packaging Tool

1. Go to the folder where you extracted the tool and run it

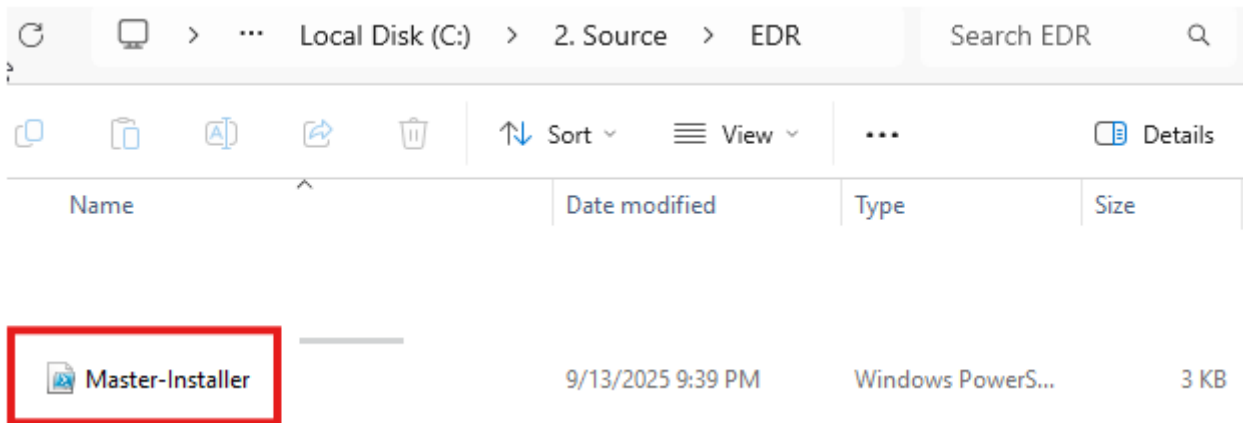


2. The tool will ask you a few questions. Enter the following:

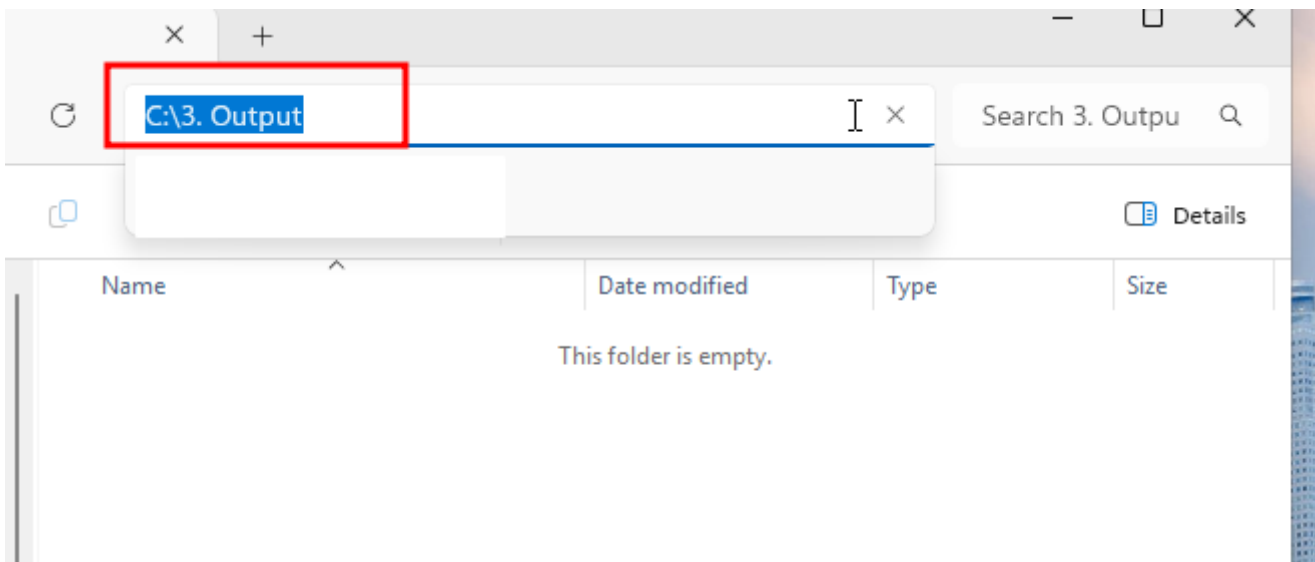
- **Source folder:** type the path to your installer folder. Click the path and copy. (e.g., `C:\Source\EDR`).



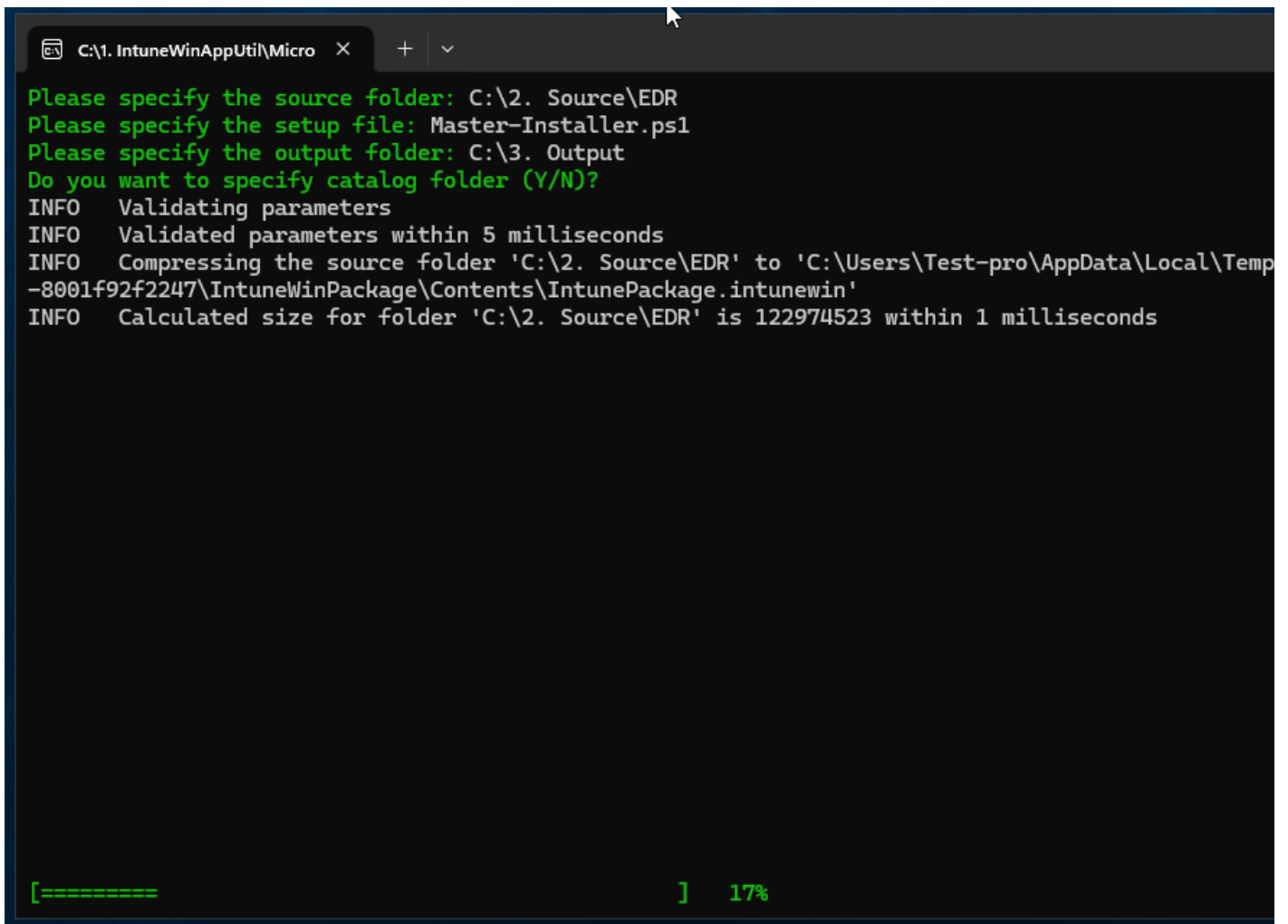
- **Setup file:** type the name of the Master Script and include its file type (e.g., Master-Installer.ps1).



- **Output folder:** type the path to your empty folder. Click the path and copy. (e.g., `C:\3. Output`).



- **Catalog folder:** just press **Enter** to skip.

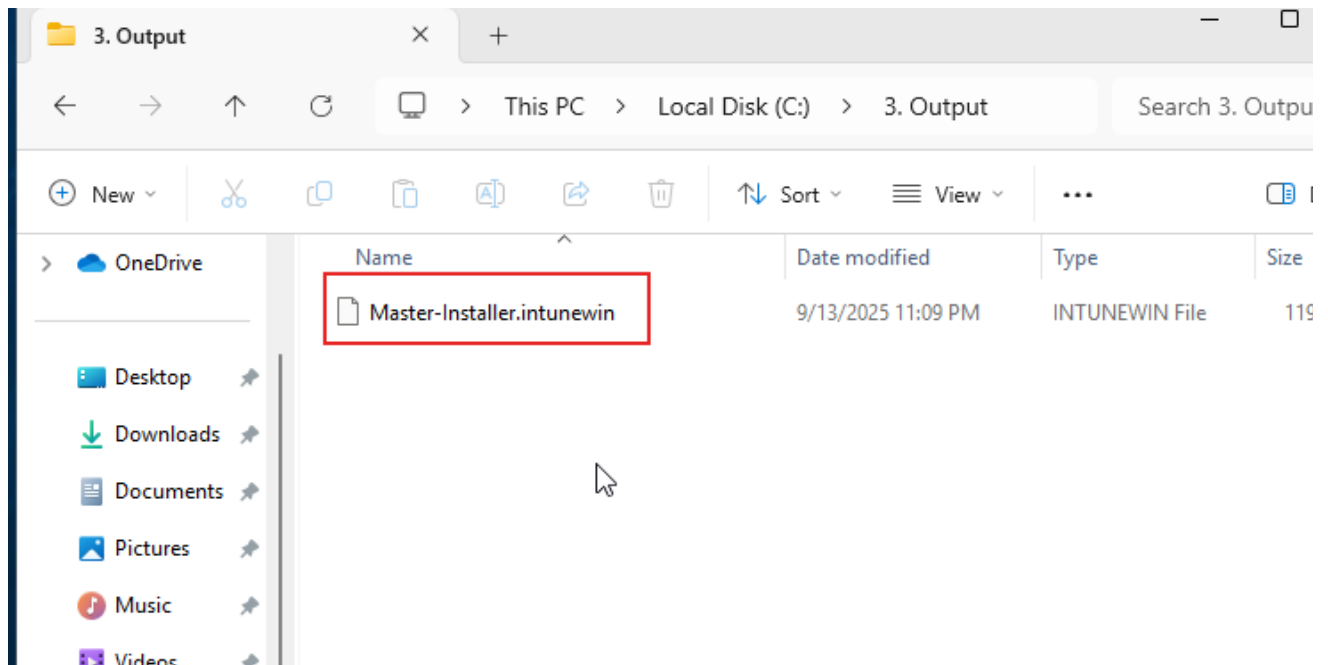


```
C:\1. IntuneWinAppUtil\Micro X + v
Please specify the source folder: C:\2. Source\EDR
Please specify the setup file: Master-Installer.ps1
Please specify the output folder: C:\3. Output
Do you want to specify catalog folder (Y/N)?
INFO Validating parameters
INFO Validated parameters within 5 milliseconds
INFO Compressing the source folder 'C:\2. Source\EDR' to 'C:\Users\Test-pro\AppData\Local\Temp-8001f92f2247\IntuneWinPackage\Contents\IntunePackage.intunewin'
INFO Calculated size for folder 'C:\2. Source\EDR' is 122974523 within 1 milliseconds

[=====] 17%
```

Check the Result

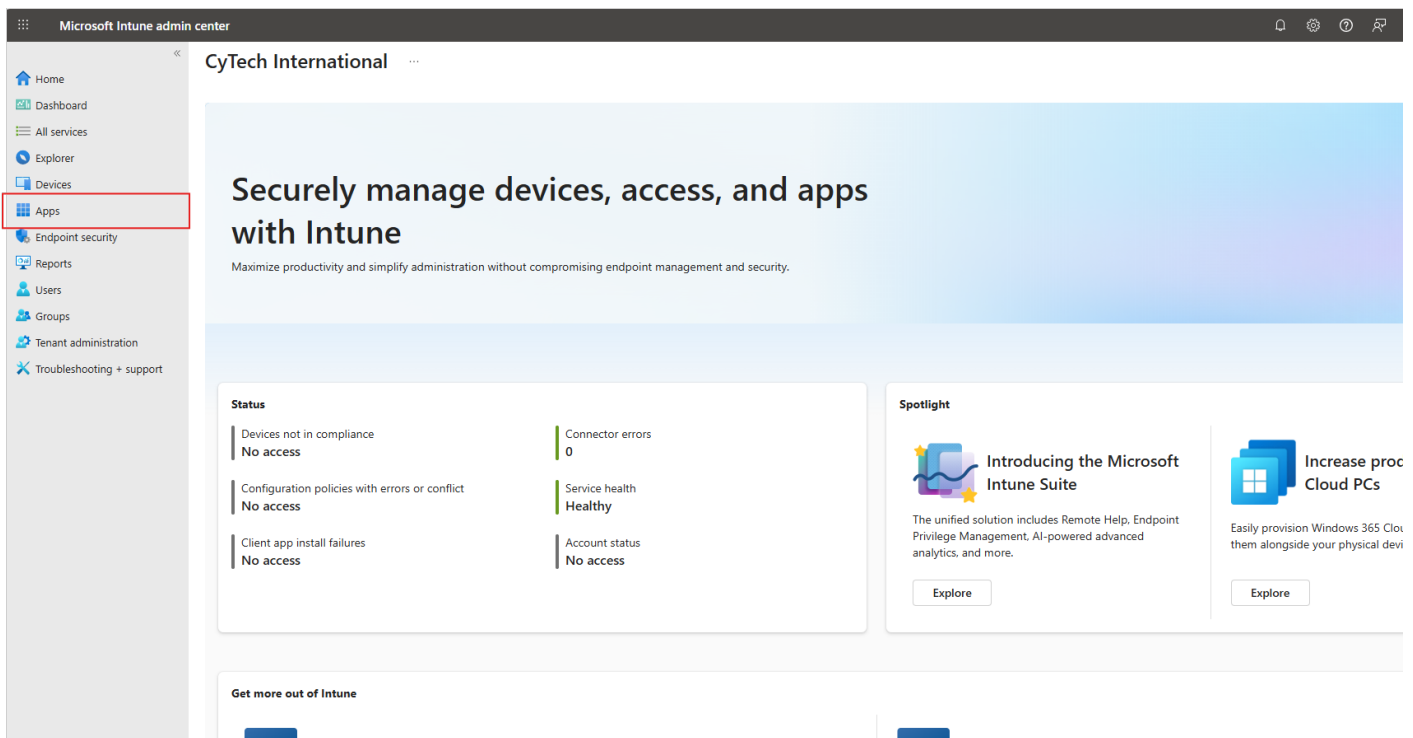
- Open your output folder (`C:\Output`).
- You should now see a file ending in `.intunewin`, for example:
- This is the file you'll upload into Microsoft Intune.



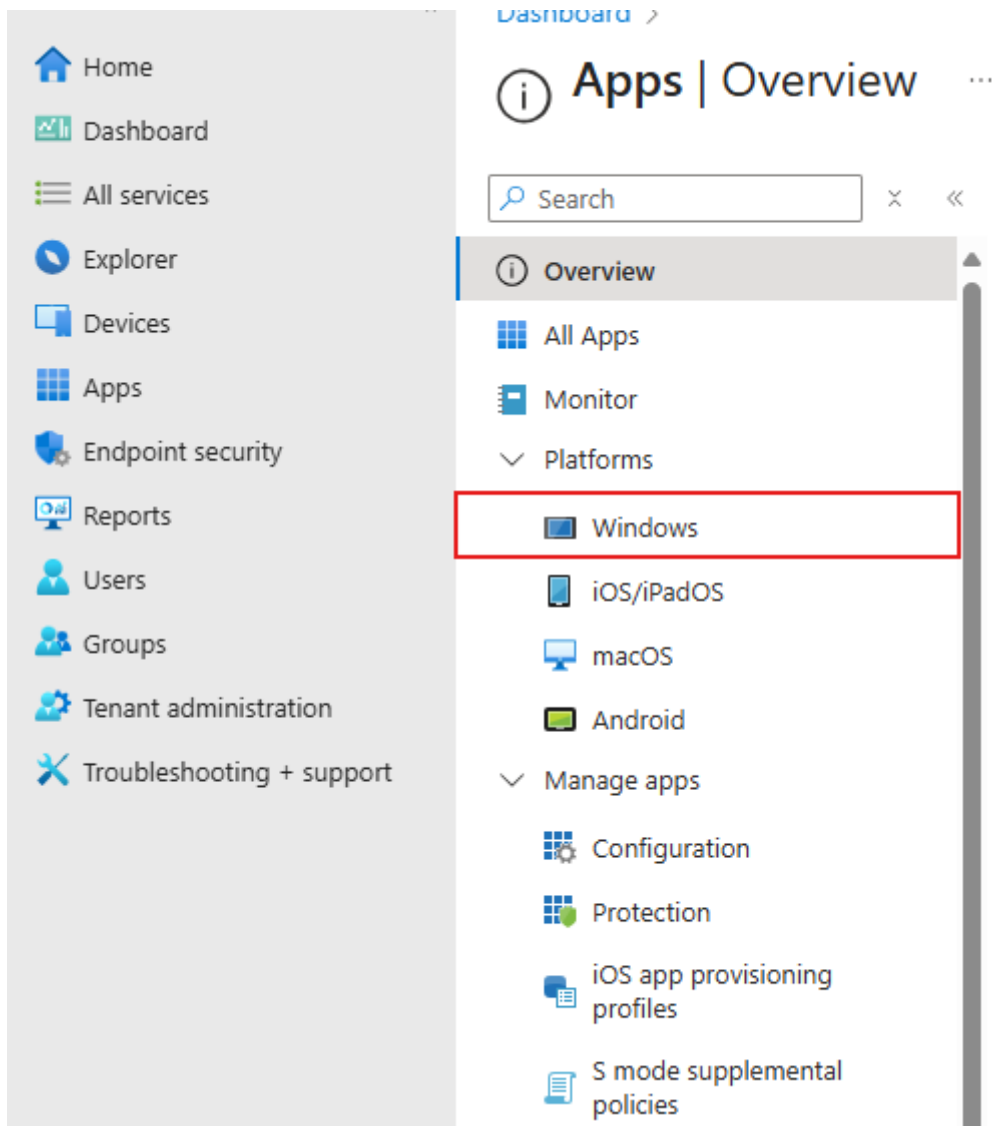
Step 2: Add the EDR App to Intune

Sign in to the **Microsoft Intune admin center** → <https://intune.microsoft.com>

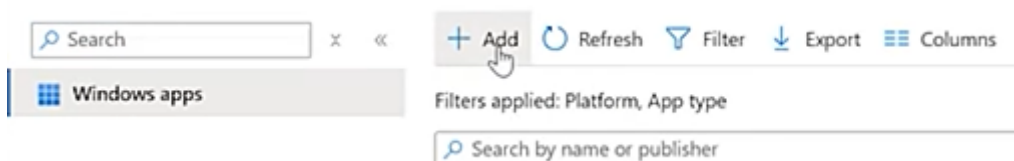
- Go to **Apps**.



- **Windows App**

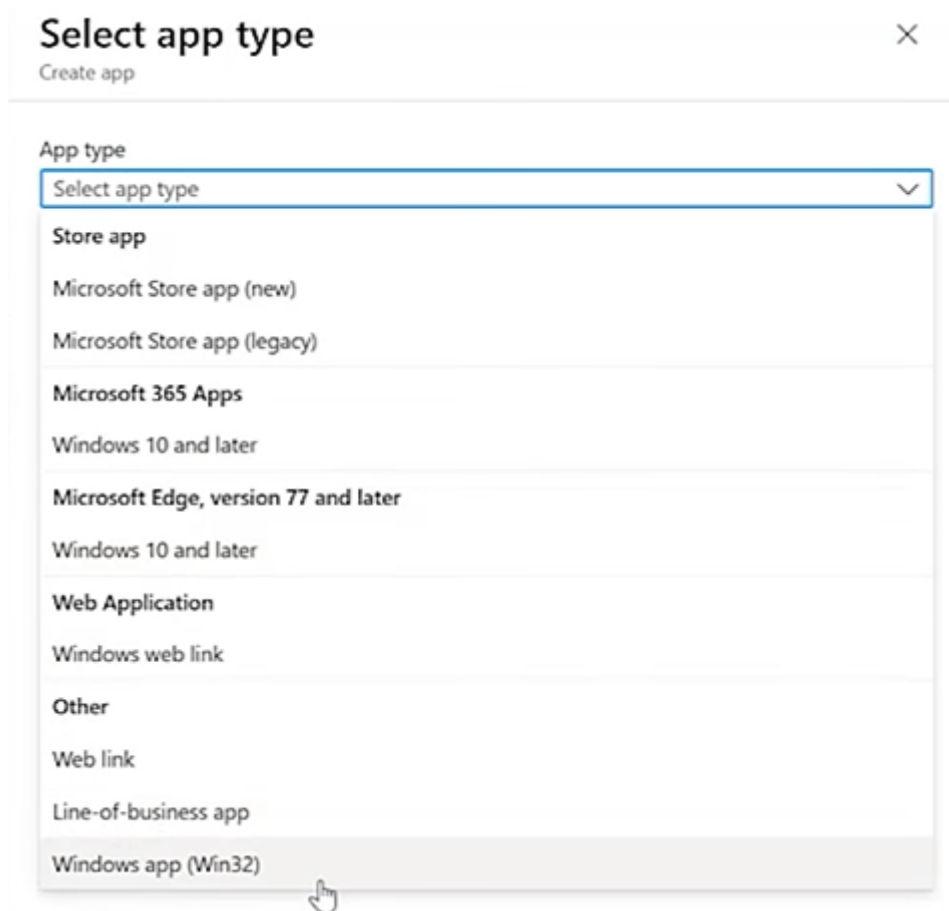


- **Add**



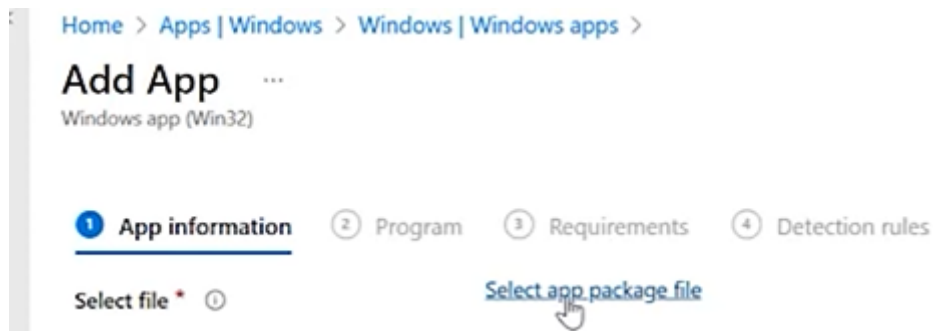
1. Select the app type:

- **Windows app (Win32)** for most EDR installers.



2. **Upload** the prepared installer package.

- **Find and upload** the **.intune** file on the **Output** folder from earlier.



3. Configure **App Information**: name, description, publisher.

1 App information
2 Program
3 Requirements
4 Detection rules
5 Dependencies
6 Supersedence
7 Assignments
8 Review + create

Select file * ⓘ

firefox installer.intunewin

Name * ⓘ

Description * ⓘ

Get help with markdown supported for descriptions.

Preview

Publisher * ⓘ

App Version ⓘ

Category ⓘ

0 selected

Show this as a featured app in the Company Portal ⓘ

Yes No

Information URL ⓘ

Privacy URL ⓘ

Developer ⓘ

Owner ⓘ

Notes ⓘ

Logo ⓘ

Select image

4. Set Program Install/Uninstall Commands.

✓ App information
2 Program
3 Requirements
4 Detection rules
5 Dependencies
6 Supersedence
7 Assignments
8 Review + create

Specify the commands to install and uninstall this app:

Install command * ⓘ

Uninstall command * ⓘ

Installation time required (mins) ⓘ

Allow available uninstall ⓘ

Yes No

Install behavior ⓘ

System User

Device restart behavior ⓘ

App install may force a device restart

Specify return codes to indicate post-installation behavior:

Return code	Code type
0	Success ✕
1707	Success ✕
3010	Soft reboot ✕
1641	Hard reboot ✕
1618	Retry ✕

+ Add

• Install Command:

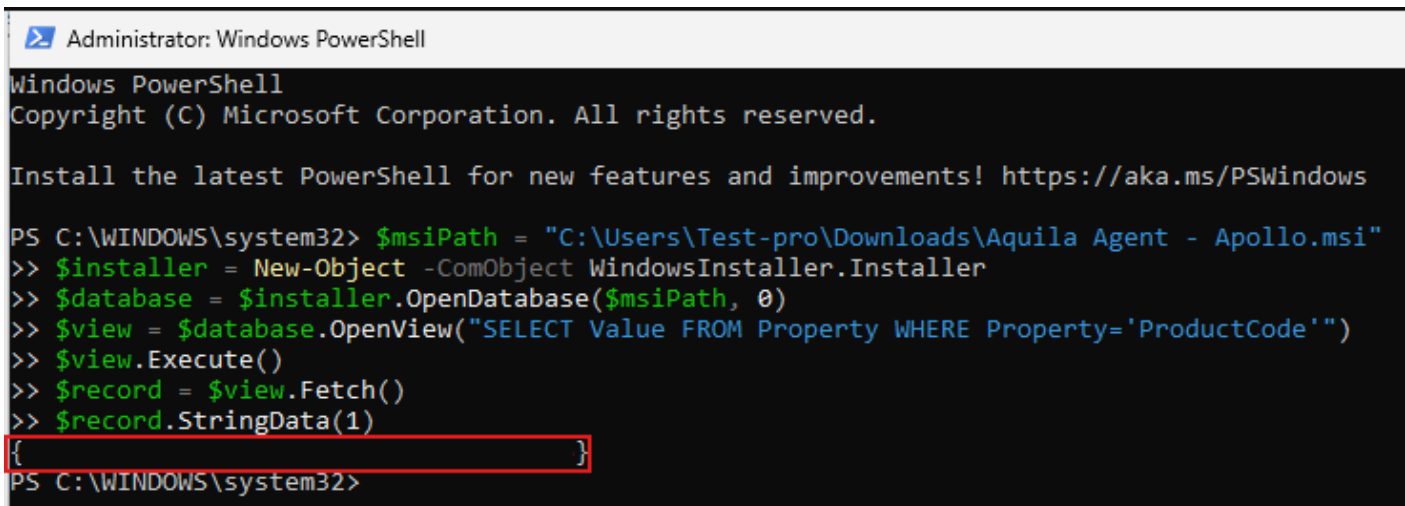
```
powershell.exe -ExecutionPolicy Bypass -File .\Master-Installer.ps1
```

- **Uninstall command:**

-Open the PowerShell as Administrator.

-Get the Product Code of Aquila Agent by running this command on **PowerShell**.

```
$msiPath = "C:\Path\To\Aquila Agent.msi"
$installer = New-Object -ComObject WindowsInstaller.Installer
$database = $installer.OpenDatabase($msiPath, 0)
$view = $database.OpenView("SELECT Value FROM Property WHERE Property='ProductCode'")
$view.Execute()
$record = $view.Fetch()
$record.StringData(1)
```

A screenshot of a Windows PowerShell window titled "Administrator: Windows PowerShell". The window shows the following text: "Windows PowerShell", "Copyright (C) Microsoft Corporation. All rights reserved.", and a message to install the latest PowerShell. Below this, the user has entered a series of PowerShell commands to retrieve the product code of an MSI file. The commands are: `$msiPath = "C:\Users\Test-pro\Downloads\Aquila Agent - Apollo.msi"`, `>> $installer = New-Object -ComObject WindowsInstaller.Installer`, `>> $database = $installer.OpenDatabase($msiPath, 0)`, `>> $view = $database.OpenView("SELECT Value FROM Property WHERE Property='ProductCode'")`, `>> $view.Execute()`, `>> $record = $view.Fetch()`, and `>> $record.StringData(1)`. The output of the last command is shown as `{` followed by a redacted area (indicated by a red box) and `}`. The prompt `PS C:\WINDOWS\system32>` is visible at the bottom.

- **For Example,** this is now your **Uninstall Command:**

```
msiexec /x "{Product Code}" /quiet /norestart
```

5. Under Requirements, select OS architecture and minimum version.

✓ App information ✓ Program **3 Requirements** ④ Detection rules ⑤ Dependencies ⑥ Supersedence ⑦ Assignments ⑧ Review + create

Specify the requirements that devices must meet before the app is installed:

Operating system architecture * ⓘ

Minimum operating system * ⓘ

Disk space required (MB) ⓘ

Physical memory required (MB) ⓘ

Minimum number of logical processors required ⓘ

Minimum CPU speed required (MHz) ⓘ

Configure additional requirement rules

Type	Path/Script
No requirements are specified.	
+ Add	

6. Add **Detection Rules** to confirm successful installation.

Add App ...

Windows app (Win32)

✓ App information ✓ Program ✓ Requirements **4 Detection rules** ⑤ Dependencies ⑥ Supersedence ⑦ Assignments ⑧ Review + create

Configure app specific rules used to detect the presence of the app.

Rules format * ⓘ

Detection rule



Create a rule that indicates the presence of the app.

Rule type * ⓘ

- MSI
- File
- Registry

Option A: **MSI Product Code** (recommended if you know it)

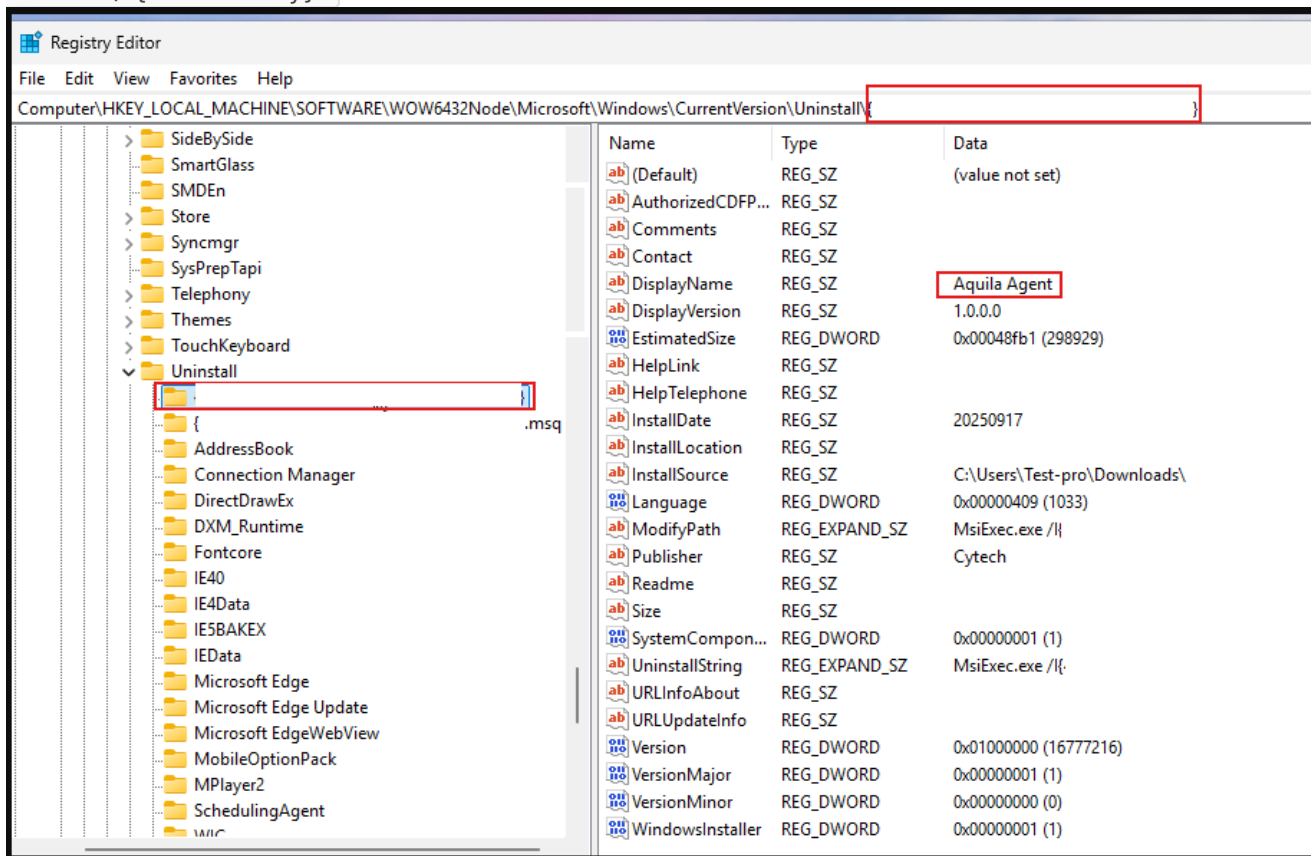
- Rule type: **MSI**
- MSI product code: `{Aquila Agent Product code}` (Please refer to **4.** on how to get the Aquila Agent Product Code:)

Option B: File exists

- Rule type: **File**
- Path: `C:\Program Files\Aquila Agent`
- File or folder: `Aquila Agent.exe`
- Detection method: File or folder exists

Option C: Registry key

- Rule type: **Registry**
- Path:
`Computer\HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\Uninstall\{Aquila Product Code}`
 - This is how to get the **Aquila Product Code**.
 - This is now the path to input: "
`Computer\HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\Uninstall\{Product Key}`"



- Value: `DisplayName` → check exists or equals "Aquila Agent"

7. Set Dependencies

Add App ...

Windows app (Win32)

- ✓ App information ✓ Program ✓ Requirements ✓ Detection rules **5 Dependencies** 6 Supersedence

Software dependencies are applications that must be installed before this application can be installed. To automatically install a child dependency app before installing the current parent app, enable the automatically install option. To only install the current parent app if the child dependency app is already detected on the device, disable the automatically install option. There is a maximum of 100 child dependency apps, including references to other apps outside of this view, forming a graph of apps. The total size of the dependency app graph is limited to the maximum of 100 plus the parent app (101 total). [Learn more](#)

Name	Automatically Install
------	-----------------------

No results.

+ Add ⓘ

8. Supersedence

Add App ...

Windows app (Win32)

- ✓ App information ✓ Program ✓ Requirements ✓ Detection rules ✓ Dependencies **6 Supersedence**

When you supersede an application, you can specify which apps will be directly updated or replaced. To update an app, disable the uninstall previous version option. To replace an app, enable the uninstall previous version option. There is a maximum of 10 updated or replaced apps, including references to other apps outside of this view, forming a graph of apps. The total size of the supersedence app graph is limited to the maximum of 10 plus the parent app (11 total). [Learn more](#)

Apps that this app will supersede

Name	Publisher	Version	Uninstall previous version
------	-----------	---------	----------------------------

No results.

+ Add ⓘ

9. Assignments

• Required

- Intune **automatically installs the app** on targeted devices.
- Perfect for your Elastic/Aquila Agent deployment.

• Available for enrolled devices

- Users can see the app in **Company Portal** and install it manually.
- Good for optional apps or testing.

- **Uninstall**
 - Used if you want to remove the app from certain groups.

Home > Apps | Windows > Windows | Windows apps >

Add App ...

Windows app (Win32)

- ✔ App information

✔ Program

✔ Requirements

✔ Detection rules

✔ Dependencies

✔ Supersedence

7 Assignments

8 Review + create

Any Win32 app deployed using Intune will not be automatically removed from the device when the device is retired. The app and the data it contains will remain on the device. If the app is not removed prior to retiring the device, the end user will need to take explicit action on the device to remove the app.

Required ⓘ					
Group mode	Group	Filter mode	Filter	End user notifications	Availability
No assignments					
+ Add group ⓘ + Add all users ⓘ + Add all devices ⓘ					
Available for enrolled devices ⓘ					
Group mode	Group	Filter mode	Filter	End user notifications	Availability
No assignments					
+ Add group ⓘ + Add all users ⓘ + Add all devices ⓘ					
Uninstall ⓘ					
Group mode	Group	Filter mode	Filter	End user notifications	Availability
No assignments					
+ Add group ⓘ + Add all users ⓘ + Add all devices ⓘ					

10. Review + Create

Add App ...

Windows app (Win32)

Notes

No Notes

Logo

No logo

Program

Install command

Uninstall command

Installation time required (mins)

Allow available uninstall

Install behavior

Device restart behavior

Return codes

60

No

System

Determine behavior based on return codes

0 Success

1707 Success

3010 Soft reboot

1641 Hard reboot

1618 Retry

Requirements

Operating system architecture

Minimum operating system

Disk space required (MB)

Physical memory required (MB)

Minimum number of logical processors required

Minimum CPU speed required (MHz)

Additional requirement rules

x64

Windows 10 22H2

No Disk space required (MB)

No Physical memory required (MB)

No Minimum number of logical processors required

No Minimum CPU speed required (MHz)

No Additional requirement rules

Detection rules

Rules format

Detection rules

Dependencies

Dependencies

No Dependencies

Supersedence

Supersedence

No Supersedence

Assignments

Group mode

Group

Filter mode

Filter

End user notifications

> Required

Available for enrolled devices

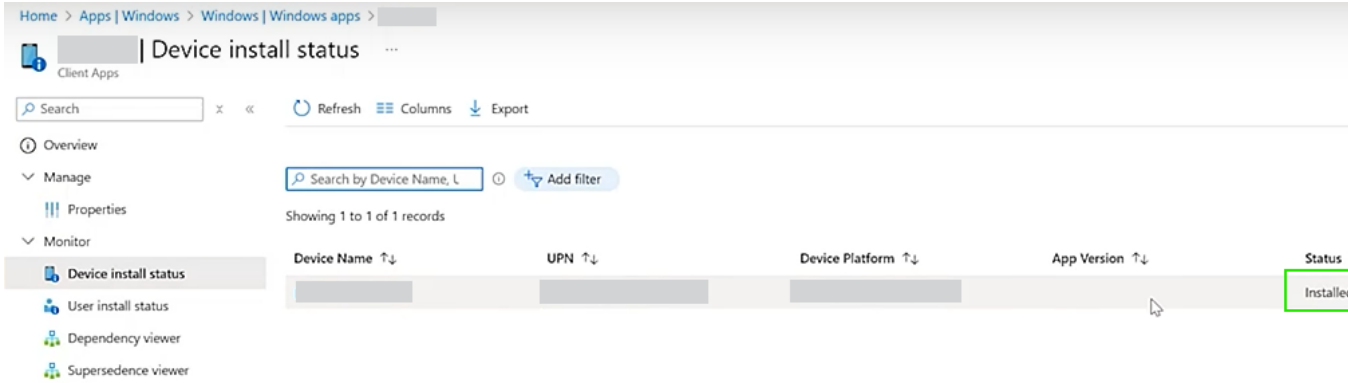
Uninstall

Previous

Create

Step 3: Monitor Deployment

- 1. In the Intune admin center, go to **Apps** → select your EDR app.
- 2. Check **Device install status** to confirm successful installations.



If you need further assistance, kindly contact our technical support at support@cytechint.com for prompt assistance and guidance.