Windows Host Intune Deployment

0.0 Table of Contents

Click on the table to jump directly to the paragraph.

1.0 Introduction

Installing applications using Microsoft Intune requires target Windows computers to be managed devices within the organization's Microsoft Intune environment.

Devices must be joined to Microsoft Entra ID and enrolled in Microsoft Intune. This is typically achieved using either **Microsoft Entra ID join** or **Hybrid Microsoft Entra ID join**, depending on the organization's device management architecture. Microsoft Entra ID was previously known as Azure Active Directory[1] (Azure AD).

This guide covers deployment of the WiseMo Host using Microsoft Intune Win32 applications for software installation and Intune scripts for configuration management.

If enrollment in Microsoft Intune is not possible or not desired, the WiseMo Host can alternatively be deployed using traditional Active Directory Group Policy–based software deployment. Please refer to the corresponding deployment guide for this method.

This guide assumes that target devices are joined to Microsoft Entra ID and enrolled in Microsoft Intune.

1.1 Prerequisites

The following prerequisites must be met before deploying the WiseMo Windows Host using Microsoft Intune.

1.1.1 Microsoft Infrastructure Requirements

- A valid Microsoft Intune subscription
  - Microsoft 365 Business Premium, E3/E5, or
  - Standalone Intune license together with Microsoft Entra ID Premium (P1 or P2), as automatic MDM enrollment requires Entra ID Premium
- Access to the Microsoft Intune Admin Center
- Microsoft Entra ID (Azure Active Directory) tenant configured
- Devices must be Microsoft Entra ID joined and enrolled in Microsoft Intune.
- Administrative permissions:
  - Intune Administrator or Global Administrator role

1.1.2 Device Requirements

Target computers must:

- Run a supported Windows version:
  - Windows 10 Pro/Enterprise/Education or newer
  - Windows 11 supported editions

[1] Some tools and system outputs (e.g., dsregcmd) still use the legacy naming.

- Be registered in Microsoft Entra ID
- Be enrolled in Microsoft Intune
- Devices may be **Microsoft Entra ID joined**[2] or **Hybrid Microsoft Entra ID joined**[3], depending on the organization's device management architecture.
- Have internet connectivity to Microsoft cloud services

Note: WiseMo Host deployment through Microsoft Intune requires that the devices are enrolled in Intune. The underlying device join model does not affect application deployment.

1.1.3 WiseMo Host Deployment Files

Before packaging the application, prepare:

- WsmHost.msi
- HostConfig.mst
- Host configuration file: Host.xml
- Host license file: Host.lic

A Windows Host should first be installed and configured on a reference machine to generate the required configuration files. Please refer to this guide for a description about how to retrieve the configuration and license files.

The WiseMo HostConfig.mst transform file can be downloaded here.

1.1.4 Packaging Tool

Download Microsoft Win32 packaging utility:

- **IntuneWinAppUtil.exe**

The IntuneWinAppUtil.exe can be downloaded here.

This tool converts MSI-based installations into the *.intunewin* format required by Intune deployment.

[2] Previously called **Azure AD joined**
[3] Previously called **Hybrid Azure AD joined**

1.2 Deployment Security and Permissions

Deployment of the WiseMo Host through Microsoft Intune uses standard Windows and Microsoft Intune application deployment mechanisms.

The installation:

- Executes using the Microsoft Intune Management Extension under the Windows **System** context
- Requires no interactive user permissions
- Does not modify Microsoft Entra ID or Active Directory configuration
- Does not require firewall, security policy, or privilege exceptions beyond normal Intune-managed application deployment

All installation and configuration actions are performed locally on the managed device using administrator privileges provided by Microsoft Intune.

1.3 Supported Deployment Scenarios

The WiseMo Host supports deployment through Microsoft Intune using standard Win32 application deployment mechanisms.

Supported deployment scenarios include:

- Microsoft Entra ID joined devices managed by Microsoft Intune
- Hybrid Microsoft Entra ID joined devices managed by Microsoft Intune
- Domain-joined devices enrolled into Microsoft Intune through automatic MDM enrollment
- Unattended or user-independent installations

No WiseMo-specific extensions or custom deployment components are required.

Sections 2 and 3 describe preparation of **Microsoft Entra ID joined** and **Hybrid Microsoft Entra ID joined** devices respectively.

All deployment scenarios described in this document use identical WiseMo Host installation and packaging procedures.

Once device enrollment is complete, deployment of the WiseMo Host can proceed as described in Section *4.0 WiseMo Host* deployment.

2.0 Preparing Microsoft Entra ID joined Devices for Intune Deployment

Organizations using cloud-managed environments without on-premises Active Directory can deploy the WiseMo Host using Microsoft Intune with devices joined directly to Microsoft Entra ID.

In this scenario, devices are managed entirely through Microsoft cloud services and no domain infrastructure or Microsoft Entra ID Connect configuration is required.

2.1 Overview

Microsoft Entra ID Join allows Windows devices to:

- Join Microsoft Entra ID directly
- Automatically enroll into Microsoft Intune
- Receive application and policy deployments from Intune

This deployment model is commonly used with:

- Microsoft Autopilot provisioning
- Cloud-first organizations
- Remote or distributed workforces

2.2 Device Enrollment

Devices may be enrolled using one of the following methods:

- Windows Out-of-Box Experience (OOBE)
- Microsoft Autopilot
- Manual enrollment from Windows Settings

Example manual enrollment:

Settings → Accounts → Access work or school → Connect → **Join this device to Microsoft Entra ID**

After sign-in with organizational credentials, the device becomes:

- Microsoft Entra ID joined
- Automatically enrolled in Intune (if automatic enrollment is enabled)

2.3 Verification

On the client device in a command prompt, run:

```
dsregcmd /status
```

Expected result:

```
AzureAdJoined  : YES

DomainJoined   : NO

MDMUrl         : is populated
```

Once device enrollment is complete, deployment of the WiseMo Host can proceed as described in Section *4.0 WiseMo Host* deployment.

3.0 Preparing Domain-Joined Devices for Intune Deployment

Organizations using Active Directory domains may want Microsoft Intune to deploy applications to existing domain-joined PCs.

The following section describes how to enable Intune management for such environments.

For domain-joined computers, this is typically implemented using Hybrid Microsoft Entra ID Join, which allows devices to remain joined to the on-premises Active Directory domain while also being registered in Microsoft Entra ID and enrolled in Microsoft Intune.

This section applies only to environments where computers are joined to an on-premises Active Directory domain. Organizations using Microsoft Entra ID joined devices should proceed to section *2.0 Preparing Microsoft Entra ID joined Devices for Intune* Deployment if enrollment has not yet been completed, or to Section *4.0 WiseMo Host* deployment if devices are already enrolled.

3.1 Overview

Hybrid Microsoft Entra ID Join allows domain-joined computers to:

- Remain members of the on-premises Active Directory domain
- Automatically register in Microsoft Entra ID
- Become manageable by Microsoft Intune

This enables centralized cloud deployment without replacing existing domain infrastructure.

3.2 Required Components

The following components must exist:

Component	Purpose
Active Directory Domain	Existing computer management
Microsoft Entra ID	Cloud identity
Microsoft Entra ID Connect	Synchronizes identities
Intune	Device & application management

3.3 Configure Microsoft Entra ID Connect

Install **Microsoft Entra ID Connect** on a domain server.

During setup:

1. Enable **Hybrid Microsoft Entra ID Join**
2. Select:
    - Windows 10 or later domain-joined devices
3. Configure device synchronization

Microsoft Entra ID Connect will automatically register domain computers in Entra ID.

3.4 Enable Automatic MDM Enrollment

In Microsoft Entra Admin Center:

Devices → Enrollment → Automatic enrollment

Configure:

- MDM User Scope = **All** (or selected users)
- MDM Authority = Microsoft Intune

This allows devices to automatically enroll into Intune after Microsoft Entra ID registration.

3.5 Configure Group Policy for Enrollment

On the domain controller:

Open:

Group Policy Management

Create or edit a GPO:

Computer Configuration
└ Administrative Templates
└ Windows Components
└ MDM

Enable:

✅ **Enable automatic MDM enrollment using default Microsoft Entra ID credentials**

Apply policy to target computer OUs.

3.6 Device Enrollment Flow

After configuration:

1. Domain computer starts
2. User signs in with domain credentials
3. Device registers in Entra ID
4. Device automatically enrolls into Intune
5. Device appears in:

Intune Admin Center → Devices

6. Assigned Win32 apps (WiseMo Host) install automatically

3.7 Verification

On a client PC:

Run:

dsregcmd /status

Expected result:

AzureAdJoined : YES
DomainJoined : YES
MDMUrl : present with some URL

This confirms Hybrid Join + Intune enrollment.

3.8 Deployment Best Practices

- Deploy to **device groups**, not users
- Test using pilot group before production rollout
- Use detection rules carefully (as shown on page 13 detecting WsmHostSvc.exe)
- Assign applications as Required for unattended Host deployment
- Avoid manual installation outside Intune once managed

4.0 WiseMo Host deployment

The Host should be deployed as a Win32 app in Intune.

Generate the Intune installation package in the following way:

1. Install a Host and configure it as per your requirements
2. Copy the **Host.xml**, **Host.lic** and **HostConfig.mst** to the same folder (c:\temp\wisemo\host in this example)
3. Copy the **Host MSI installer** file to the same folder (c:\temp\wisemo\host in this example).
4. Using the **IntuneWinAppUtil** application, run the following command to package up the Host files:

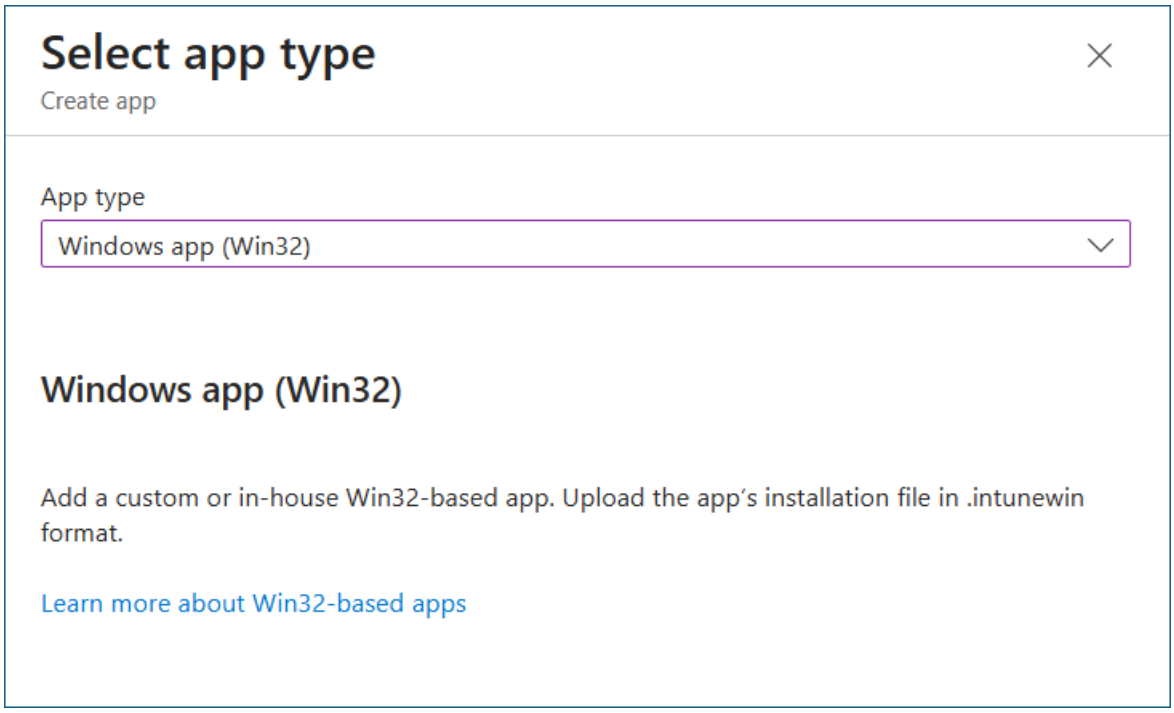
   IntuneWinAppUtil -c "c:\temp\wisemo\host" -s "c:\temp\wisemo\host\WsmHost.msi" -o "c:\temp\wisemo\Output"

   A file called **WsmHost.intunewin** will be created in the Output folder

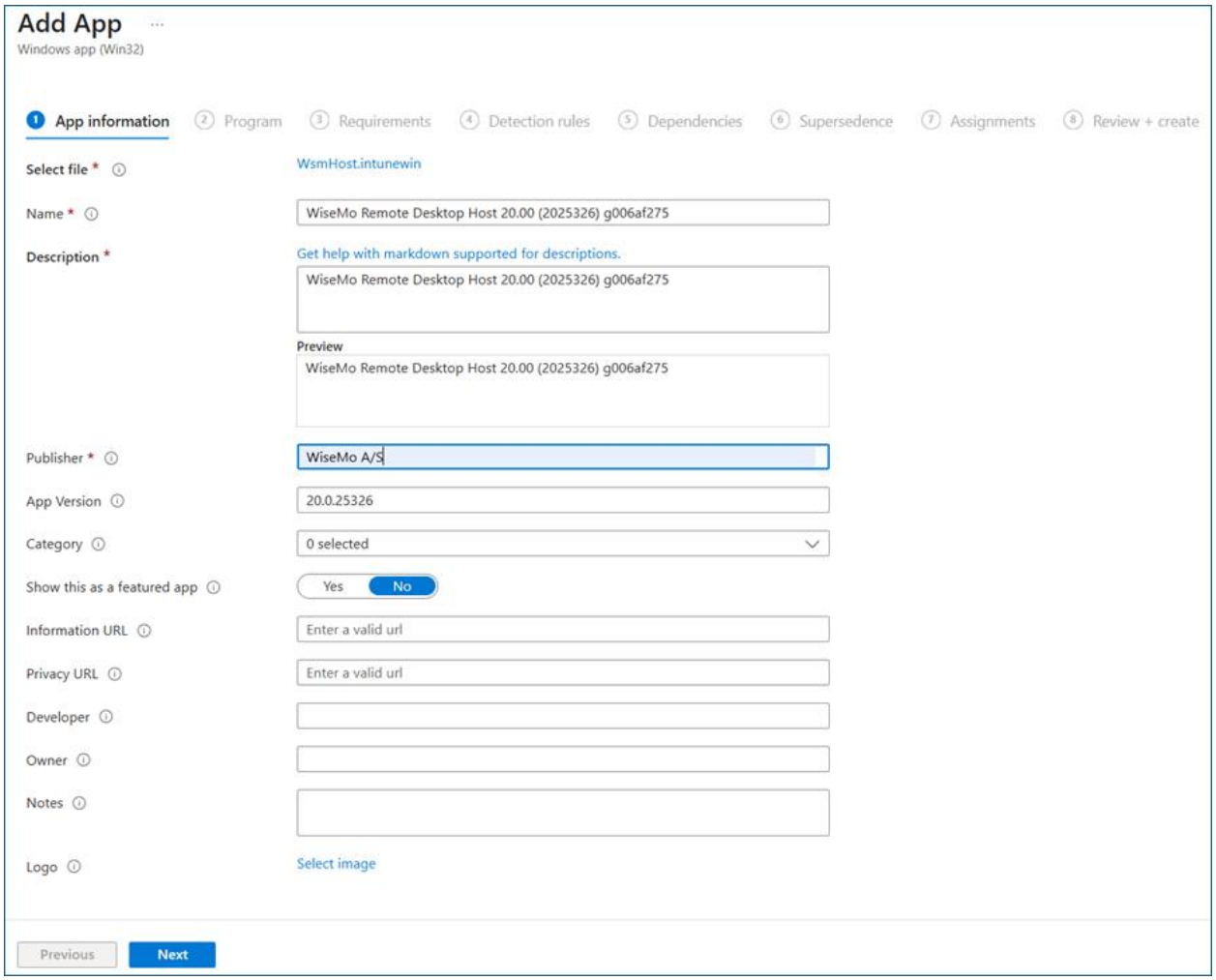
WiseMo Host deployment uses Microsoft Intune standard Win32 application deployment and requires the Microsoft Intune Management Extension (IME), which is automatically installed on devices when Win32 applications are assigned.

Within Intune, do the following:

- Go to **Apps > Windows**
- Create a new Windows app by clicking **+ Create**
- Select app type as Windows app (Win32)



- Click **Select** in the bottom of the screen
- Click **Select app package file** and locate your **WsmHost.intunewin** file and select **Ok**. Add the Publisher info:



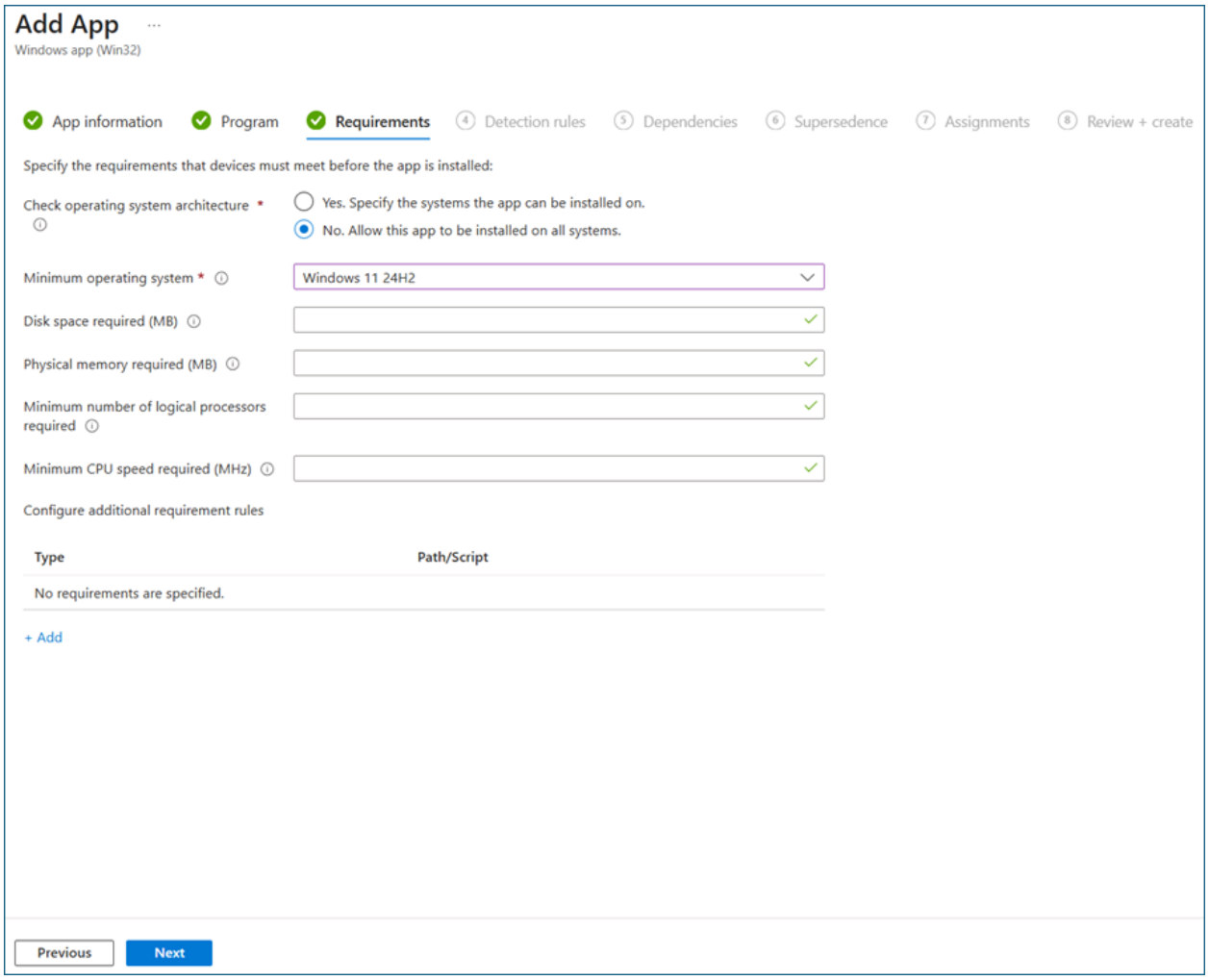
- In the Program tab, replace the Install Command with the following (you can also replace the /quiet switch with /qn):

  msiexec /i "WsmHost.msi" TRANSFORMS=HostConfig.mst /quiet

  The msi file name should correspond to the name you used when creating the .intunewin file.

- Still on the **Program** tab, set *Install behavior* to **System** to ensure the WiseMo Host installs unattended and independently of the logged-on user. Installing in User context may prevent successful deployment on shared or unattended devices.

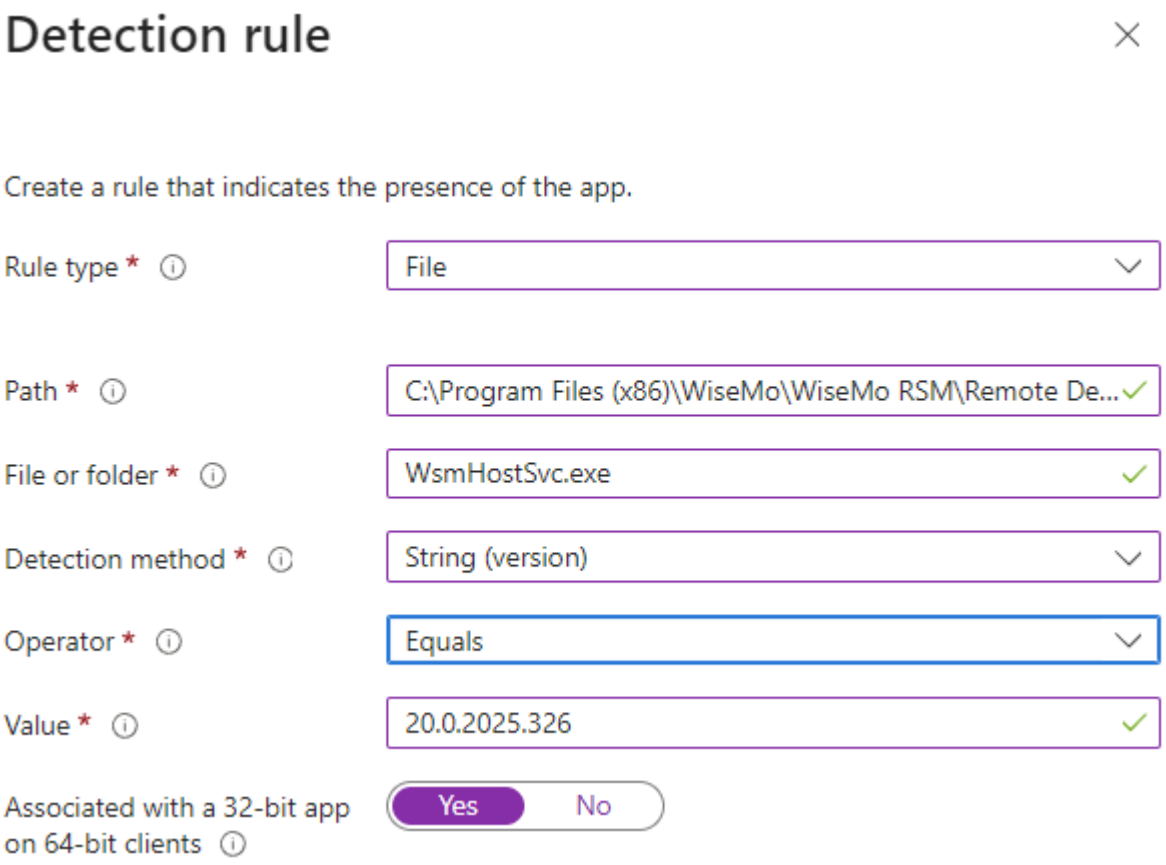
- In the Requirements tab, select your minimum operating system level:



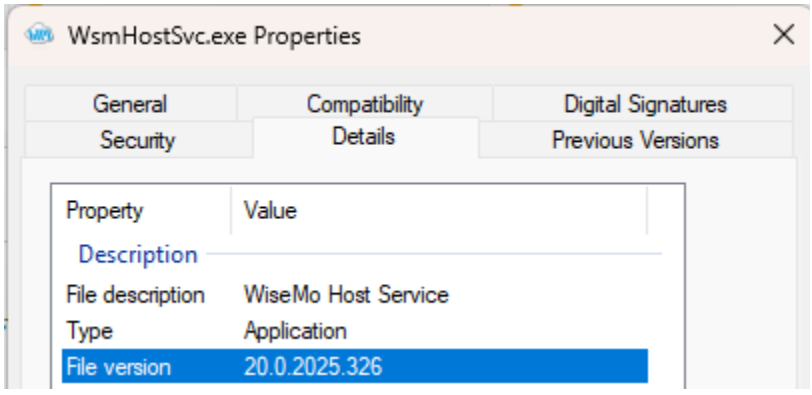
The WiseMo Host MSI installer supports from Windows 7.

- In the Detection rules tab, select Manually configure detection rules and define a rule that detects the presence of the app. Detection rules determine whether Intune considers the app installed.
  The detection rule must uniquely identify a successful Host installation to prevent repeated installations.
  Also, as we want to upgrade the Host app later, it is important that the detection rule can distinguish this app and a later upgrade version from each other.

  A safe method is to use the file version of the **WsmHostSvc.exe** file on file path **C:\Program Files (x86)\WiseMo\WiseMo RSM\Remote Desktop Host**. Fill out the Detection rule like this:

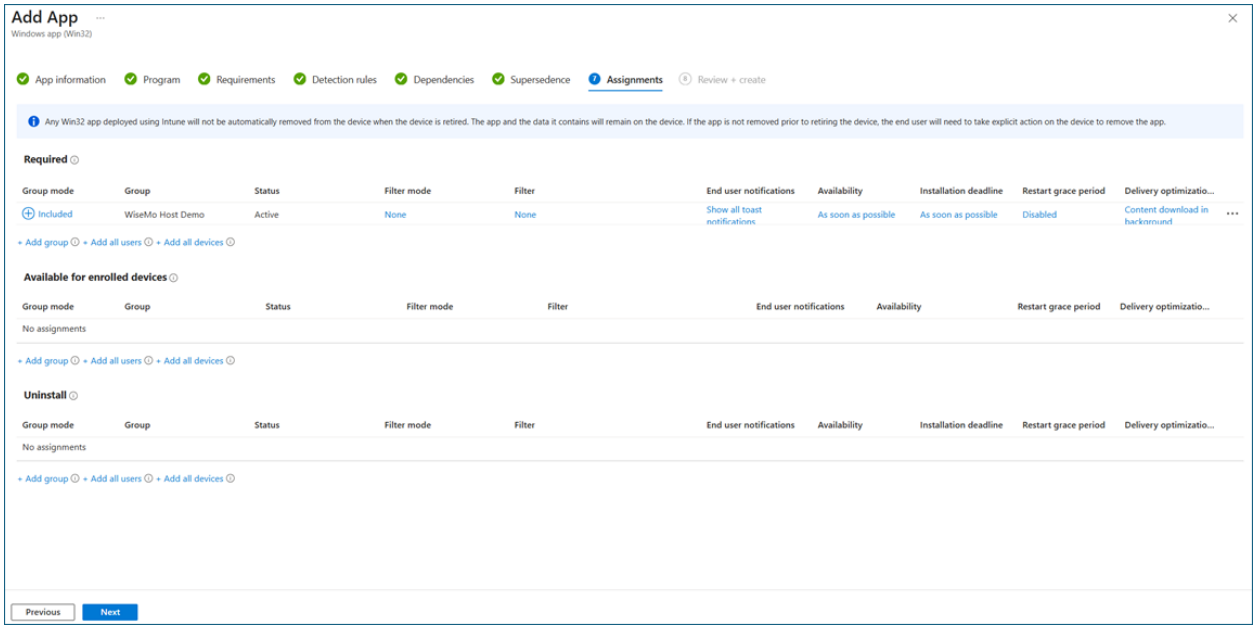


The file version can be found by right clicking WsmHostSvc.exe app and select **Properties > Details**:



- Skip the Dependencies tab.

- Also skip the **Supersedence** tab for now. We'll need the **Supersedence** tab when we need to upgrade the Host to a newer version/build.
- In the Assignments tab, select which target groups should receive the new app package. In this example, the device group is called WiseMo Host Demo:



- Select the Review + Create tab to preview your options and create the Win32 app.

5.0 Verifying WiseMo Host Installation

Deployment doesn't happen immediately. Often, it's necessary to wait 5-10 minutes or even an hour.
If deployment shows an error or if the Host is not deployed after an hour, please see the *8.0 Troubleshooting – Microsoft Intune Deployment*.

After deployment, confirm successful installation:

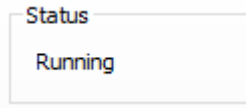
- Verify deployment status in Microsoft Intune
  - Open **Microsoft Intune Admin Center**
  - Navigate to: **Apps → Windows apps → WiseMo Host → Device install status**
  - Confirm the device reports **Installed**

Device name	UPN	Device platform	App version	Status	Status details	Filter
VM-WIN11-PRO	OleSetnes@WiseMo652.onı	Windows 10.0.22631.6199	20.0.25326	Installed		

- Log on to the user desktop
  - Locate the WiseMo Host icon in the notification area:

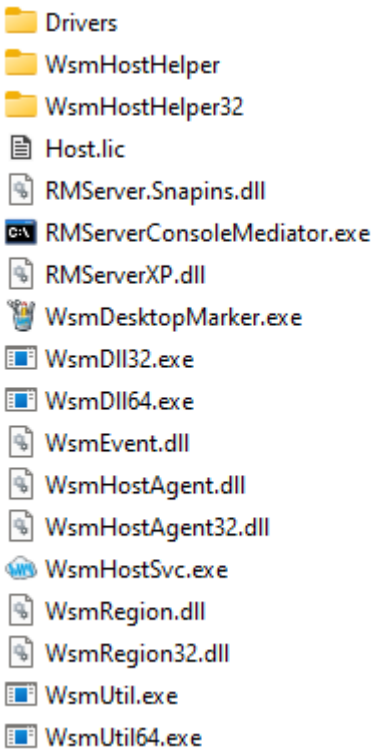
    

  - Open the Host user interface (e.g. by double clicking the icon) and verify that the **Status** is shown as *Running*.

- Confirm device connectivity from the WiseMo Guest
  - Start a WiseMo Guest on another computer and try to connect to the Host computer via myCloud or the local LAN via an IP address or computer name.
  - If you can connect and everything is working as expected you can skip the rest of this list.

- To dive a step deeper you can verify the installation path on the target computer:
  C:\Program Files (x86)\WiseMo\WiseMo RSM\Remote Desktop Host\

- And verify the WiseMo Remote Desktop Host service is running
  - Open **Task Manager** (Ctrl+Shift+Esc) → **Details** tab and confirm that the following processes are running:

WsmHostManager.exe	22744	Running	os	1	00
WsmHostSvc.exe	103240	Running	SYSTEM	0	00

Successful completion of the above steps confirms that the WiseMo Host has been correctly deployed and is operational under Microsoft Intune management.

6.0 Updating the WiseMo Host Using Microsoft Intune

This section describes recommended methods for updating the WiseMo Host when deployed as a Win32 application via Microsoft Intune.

6.1 Overview

Microsoft Intune does not provide native “in-place update” logic for Win32 applications.
Instead, updates are handled by:

- Uploading a new application package
- Using detection rules to determine installation state
- Superseding previous versions

6.2 Recommended Approach – Supersedence

The recommended method for updating the WiseMo Host is to use **application supersedence** in Microsoft Intune.

Concept

- A new WiseMo Host version supersedes an older version
- Intune automatically upgrades existing installations based on assignment and detection rules

When configuring supersedence, you control the update behavior using the **“Uninstall previous version”** option:

- **Uninstall previous version = No (recommended)**
  → Performs an in-place upgrade of the existing installation
- **Uninstall previous version = Yes**
  → Uninstalls the previous version before installing the new version

Since the WiseMo Host installer supports in-place upgrades, the recommended setting is No

Configuration Steps

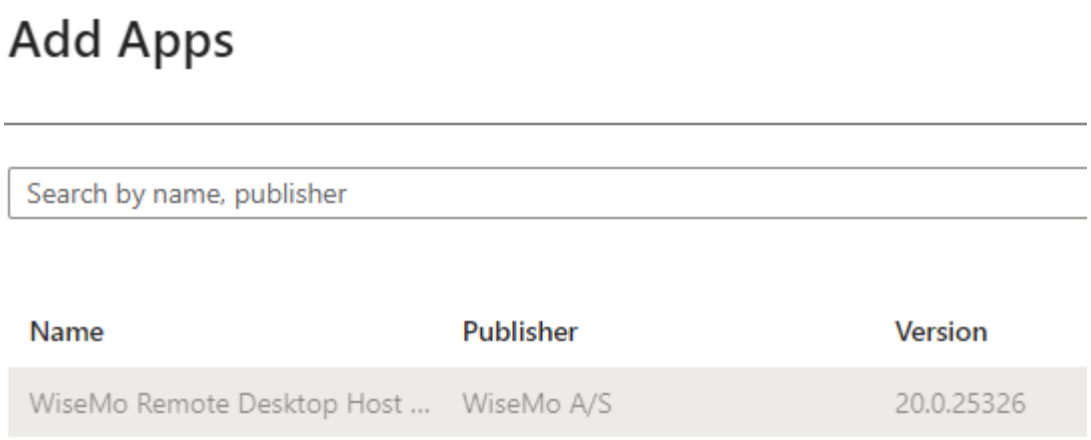
In Intune, create a new Win32 application using the updated WiseMo Host MSI installer by following the same steps described in Section *4.0 WiseMo Host deployment* until reaching the Supersedence tab.

Important: The new application must use the same installation logic and detection method as the previous version. Remember to use the File version of the new WsmHostSvc.exe app – in this case 20.0.2026.76.

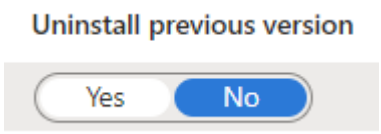
Incorrect detection rules may prevent proper upgrade behavior.

On the **Supersedence** tab:

1. Select the previous app
    - Click **+Add**
    - Select the previous WiseMo Host app



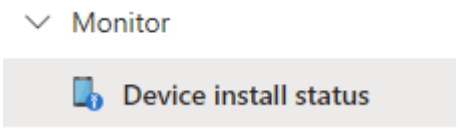
    - Choose **No** for **Uninstall previous version**:



2. Assign the new app:
    - Same target group as before
    - Assignment type: **Required**

3. Finally, click the **Create** button.

Open the new WiseMo Host app and select Device Install Status to monitor the installation/upgrade progress.



If the Host app is not updated after 5-10 minutes, verify that the app detection rules for the existing and new apps are configured correctly for Intune to be able to distinguish the two apps from each other.

Result

- Devices with an existing WiseMo Host installation → automatically upgraded
- Devices without WiseMo Host → latest version installed
- The previous version is updated as part of the upgrade process

7.0 Updating WiseMo Host Configuration via Microsoft Intune (PowerShell Scripts)

This section describes how to update the WiseMo Host configuration file (host.xml) using Microsoft Intune PowerShell Scripts and applying the changes by restarting the WiseMo Host service.

7.1 Overview

Microsoft Intune supports deployment of PowerShell scripts to managed Windows devices through the Intune Management Extension (IME).

This method allows:

- Deployment of updated configuration files without reinstalling the Host
- Rapid rollout of configuration changes
- Execution of administrative actions such as restarting services

WiseMo Host updates are handled via Intune application supersedence, while configuration changes are deployed independently using PowerShell scripts executed by the Intune Management Extension.

7.2 Creating the update Script

The script must:

1. Copy the updated host.xml to the WiseMo Host configuration directory
2. Restart the WiseMo Host service
3. Optionally log execution for troubleshooting

Intune PowerShell scripts do not support distributing additional files alongside the script. Therefore, the configuration must be embedded in the script or retrieved from an external source (e.g. download or server share). The approach below is to include file directly in the script.

Open for example Notepad and create a new file. Copy and paste the script below:

```
# WiseMo Host configuration update script
# - Writes host.xml only if content has changed
# - Restarts the Host service only when needed
# - Logs actions to %ProgramData%\WiseMo\Logs\ConfigUpdate.log

$targetPath = "$env:ProgramData\WiseMo\WiseMo RSM\Remote Desktop Host\host.xml"
$targetDir = Split-Path $targetPath -Parent
$logPath = "$env:ProgramData\WiseMo\Logs\ConfigUpdate.log"

# Adjust service name if needed
$serviceName = "WsmHostAgent"

# Embed the desired configuration here between @" and "@
$newXmlContent = @"
  [Copy-paste the host configuration file here and delete this line]
"@

function Write-Log {
  param([string]$Message)

  try {
    $timestamp = Get-Date -Format "yyyy-MM-dd HH:mm:ss"
    $logDir = Split-Path $logPath -Parent

    if (-not (Test-Path $logDir)) {
      New-Item -ItemType Directory -Path $logDir -Force | Out-Null
    }

    "$timestamp - $Message" | Out-File -FilePath $logPath -Encoding utf8 -Append
```

```
  }
  catch {
    # Do not fail the script only because logging failed
  }
}

try {
  Write-Log "Starting WiseMo Host configuration update."

  # Ensure target directory exists
  if (-not (Test-Path $targetDir)) {
    New-Item -ItemType Directory -Path $targetDir -Force | Out-Null
    Write-Log "Created directory: $targetDir"
  }

  $needsUpdate = $true

  if (Test-Path $targetPath) {
    # Read current file as a single string
    $existingXmlContent = Get-Content -Path $targetPath -Raw -ErrorAction Stop

    if ($existingXmlContent -eq $newXmlContent) {
      $needsUpdate = $false
      Write-Log "No configuration change detected. Existing host.xml matches desired content."
    }
    else {
      Write-Log "Configuration change detected. host.xml will be updated."
    }
  }
  else {
    Write-Log "No existing host.xml found. A new file will be created."
  }

  if ($needsUpdate) {
    # Write updated XML
    $newXmlContent | Out-File -FilePath $targetPath -Encoding utf8 -Force
    Write-Log "Wrote updated host.xml to: $targetPath"

    # Restart service only if config changed
    $service = Get-Service -Name $serviceName -ErrorAction SilentlyContinue
    if ($null -ne $service) {
       Restart-Service -Name $serviceName -Force -ErrorAction Stop
      Write-Log "Restarted service: $serviceName"
    }
    else {
      Write-Log "Service not found: $serviceName"
```

```
    }
  }

  Write-Log "WiseMo Host configuration update completed successfully."
  exit 0
}

catch {
  Write-Log "ERROR: $($_.Exception.Message)"
  exit 1
}
```

Open the host.xml file that should be deployed. Copy the entire content of the file, e.g. Ctrl-A and then Ctrl-C.

Switch to the script and replace the line “[Copy-paste the host configuration file here and delete this line]” with the clipboard content, e.g. Ctrl-V. Everything between @" and "@ in the script is treated as the **raw** Host configuration file. The construction (@" ... "@) preserves quotes and formatting exactly and no escaping of quotation marks is required. It’s important that the “[Copy-paste the host configuration file here and delete this line]” line is removed.

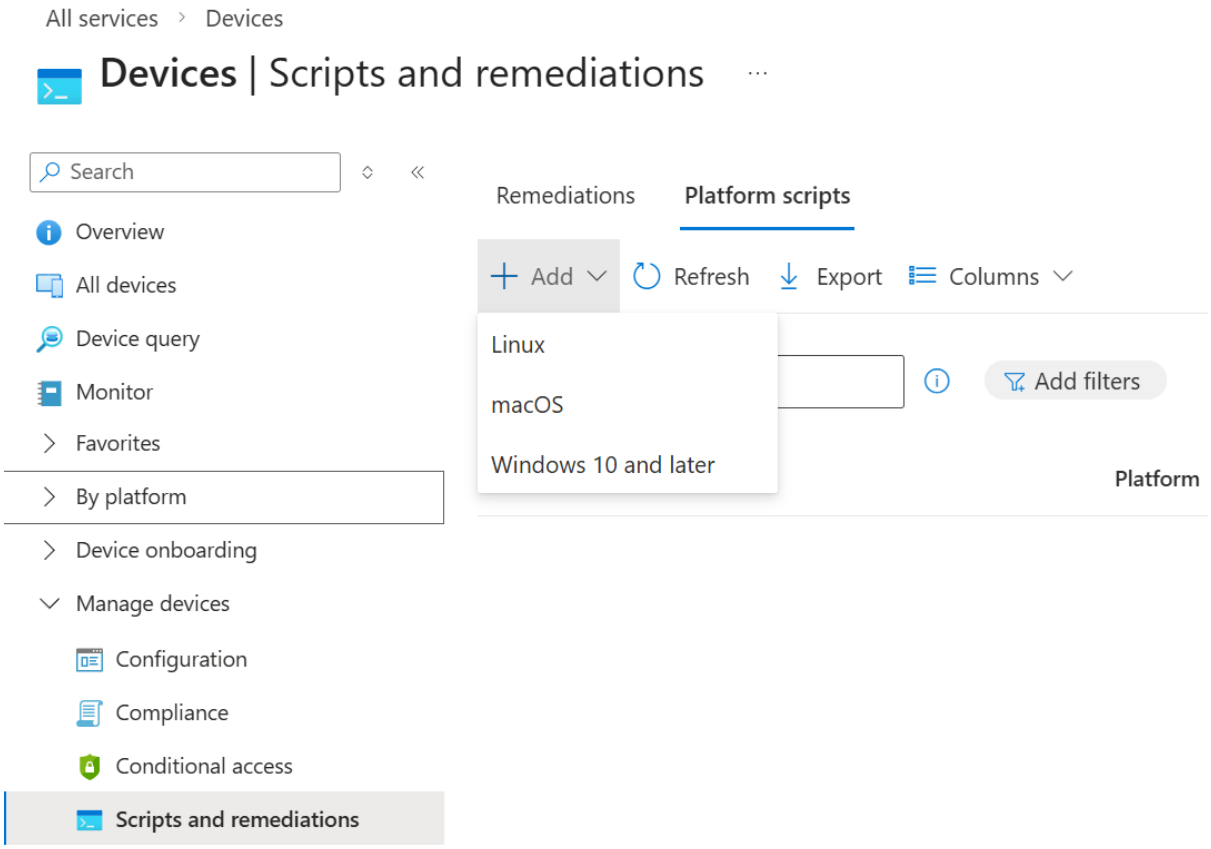
Save the script:

- Select **File > Save As...**
- Select an applicable folder
- Select **Save As Type > All Files** (*.*)
- **Give the file a name**, for example UpdateWiseMoHostConfiguration.ps1

7.3 Deployment Steps in Intune

Open the Microsoft Intune and do the following

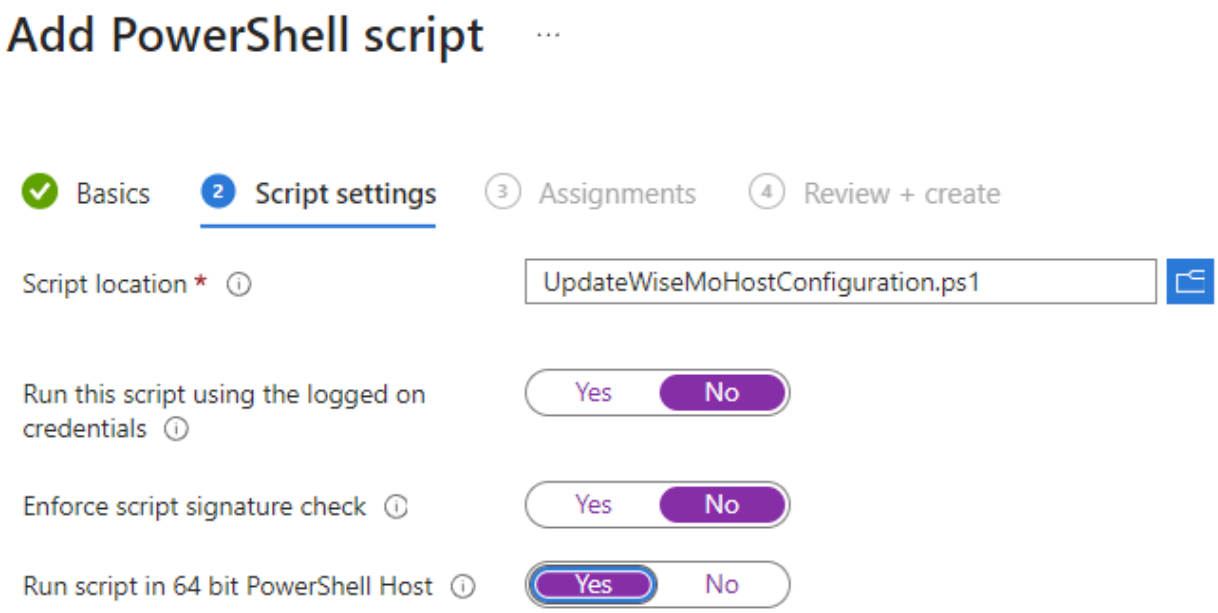
1. Navigate to **Devices > Manage Devices > Scripts and remediations**
2. Select **Platform scripts** and click +**Add** > **Windows 10 and later**.



3. In **Basics**, enter the following properties, and select **Next**:
    - **Name**: Enter a name for the PowerShell script.
    - **Description**: Enter a description for the PowerShell script. This setting is optional but recommended.
4. In **Script settings**, enter the following properties, and select **Next**:
    - **Script location**: Browse to the PowerShell script.
    - **Run this script using the logged on credentials**: Choose **No** to run the script in the system context. The script must run in the SYSTEM context to have permission to write to %ProgramData% and restart services.
    - **Enforce script signature check**: Select **No** (recommended for most deployments) to allow unsigned scripts, which simplifies deployment as no certificate infrastructure is required.

      Select **Yes** to require that the script is digitally signed using a trusted code-signing certificate. This is recommended in high-security environments.

- **Run script in 64-bit PowerShell host**: Select **Yes** to run the script in a 64-bit PowerShell host on a 64-bit client architecture. If the client is 32 bit the script runs in a 32-bit PowerShell host.



5. Assign the script to a **device group** that should receive the new configuration.
6. In **Review + add**, a summary is shown of the settings you configured. Select **Add** to save the script. When you select **Add**, the policy is deployed to the groups you chose.

You can monitor the run status of PowerShell scripts for users and devices in the portal.

In **PowerShell scripts**, select the script to monitor, choose **Monitor**, and then choose the following report:

- **Device status**

After a while you will see the result (you might have to press F5 to refresh the page):

Device name	User name	OS Version	Status	Last Updated
VM-WIN11-PRO	OleSetnes@WiseMo652.onmicro...	10.0.22631.6199	Succeeded	2026-03-20T12:20:00Z

Note: Intune PowerShell scripts may not automatically rerun on a device simply because the administrator expects a retry. If the script content changes, or if a new script object is created, Intune will treat it as a new execution target.

7.4 Troubleshooting

If the Host configuration is not updated, check the following:

- Script not running:
    - Check: C:\ProgramData\Microsoft\IntuneManagementExtension\Logs\IntuneManagementExtension.log

      You will not find the script name as Intune uses a GUID. You can instead search for .ps1.
- Script executed but no change:
    - Check ConfigUpdate.log in %ProgramData%\WiseMo\Logs\
- XML not updated:
    - Ensure formatting matches exactly (string comparison is strict)
    - Verify that you have copied the entire content of the Host configuration file.
- If Intune reports script failure even though host.xml was updated, verify that the log directory exists. Out-File does not create missing directories and may cause the script to be reported as failed after the configuration has already been written.

8.0 Troubleshooting – Microsoft Intune Deployment

This section describes common issues encountered when deploying the WiseMo Host using Microsoft Intune, along with recommended diagnostics and resolutions.

8.1 No Application Deployment Occurs

If the WiseMo Host is not installed and no errors are shown in Intune, verify the device state.

Diagnostic

Open a command prompt and run the following command on the target device:

dsregcmd /status

Expected values

The following conditions must be met:

- AzureAdJoined : YES
- AzureAdPrt : YES
- MdmUrl is populated (i.e. shows an URL)

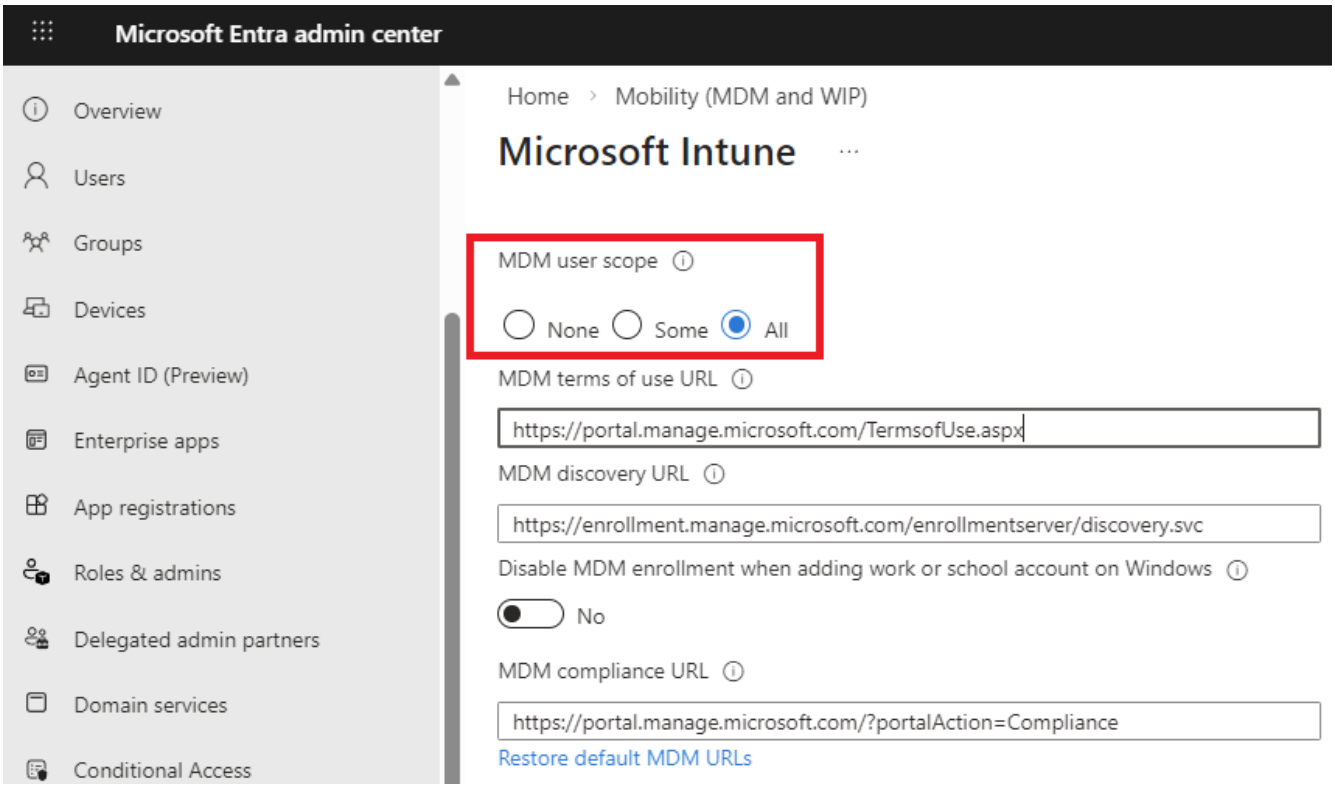
- WorkplaceJoined : NO

Explanation

- If AzureAdJoined is **NO**, the device is not properly joined to Microsoft Entra ID
- If AzureAdPrt is **NO**, the user is not signed in with an Microsoft Entra ID account
- If MdmUrl is empty, the device is not enrolled in Intune
- If WorkplaceJoined is **YES**, the device is only registered (BYOD scenario), not joined

Resolution

- Ensure the device is joined using:
  Settings → Accounts → Access work or school → Connect → "Join this device to Microsoft Entra ID"
- Sign in using a Microsoft Entra ID account (not a local account)
- Verify MDM scope is configured in Microsoft Entra ID (https://entra.microsoft.com/)



8.2 Device is Only Microsoft Entra ID Registered (Workplace Join)

Symptoms

- AzureAdJoined : NO
- WorkplaceJoined : YES
- Applications do not deploy
- Intune Management Extension is not installed

Explanation

The device is only registered (Workplace Join), which does not support Win32 application deployment.

Resolution

- Disconnect the work account
- Reconnect using **“Join this device to Microsoft Entra ID”**
- Do not use “Enroll only in device management”

8.3 User Logged in with Local Account

Symptoms

- AzureAdJoined : YES
- AzureAdPrt : NO
- Executing Account Name shows a local user
- No application deployment occurs

Explanation

Win32 application deployment requires a valid Microsoft Entra ID user session (Primary Refresh Token).

Resolution

- Sign out
- Log in using the Microsoft Entra ID account:

  AzureAD\<username>

8.4 Intune Management Extension Not Installed or Not Active

Symptoms

- Folder is missing or empty:

  C:\Program Files (x86)\Microsoft Intune Management Extension\

- No log folder:

  C:\ProgramData\Microsoft\IntuneManagementExtension\Logs

Explanation

The Intune Management Extension (IME) has not yet initialized or received policy.

Resolution

- Ensure device is correctly joined and user is signed in (see *8.1 No Application Deployment Occurs*)
- Restart the IME service:

  net stop IntuneManagementExtension

  net start IntuneManagementExtension

- Trigger manual sync:

  Settings → Accounts → Access work or school → [Account name] → Info → Sync

8.5 Application Not Assigned Correctly

Symptoms

- No activity in IME logs
- No download or install attempt

Explanation

The application is not targeted to the device.

Resolution

- Verify assignment in Intune:

  - Assignment type must be **Required**
  - Ensure the device is a member of the assigned group
- For testing, assign the application to **All devices**

8.6 Incorrect Detection Rule

Symptoms

- Application never installs
- IME logs show detection as already satisfied

Explanation

Intune believes the application is already installed.

Resolution

- File exists → always TRUE → no upgrade triggered. Use a **version-aware detection rule**.
- Recommended: detect the **file version** of WsmHostSvc.exe in
  C:\Program Files (x86)\WiseMo\WiseMo RSM\Remote Desktop Host\
  A file existence rule is not suitable for distinguishing between installed versions.

8.7 Installation Command Issues

Symptoms

- Application downloads but does not install
- IME logs show execution failure

Resolution

Use a silent MSI installation:

msiexec /i "WsmHost.msi" /qn /norestart

Optional logging:

msiexec /i "WsmHost.msi" /qn /norestart /l*v %ProgramData%\WiseMoInstall.log

8.8 Expected Deployment Timeline

Intune deployments are not immediate. Typical delays:

- Policy sync: ~5–15 minutes
- App deployment: up to 30–60 minutes
- Script execution: similar intervals

Manual sync can be triggered, but does not guarantee immediate execution.

8.9 Summary Checklist

Before troubleshooting packaging, ensure:

- Device is Microsoft Entra ID joined
- User is signed in with a Microsoft Entra ID account
- Device is enrolled in Intune (MDM URL present)
- Intune Management Extension is installed and running
- Application is assigned as **Required**
- Detection rule is correct

If all conditions are met, the WiseMo Host will deploy automatically.

8.10 Lessons Learned

- A device can appear in Intune without being fully MDM enrolled
- dsregcmd /status is the authoritative validation tool
- Detection rules control upgrade behavior
- Intune deployments are asynchronous and may take time
- Scripts do not automatically include external files
- Logs may not update immediately after assignment