

Intune Deployment and Configuration of the WiseMo Android Host

1.0 Overview

Automation of the installation and configuration of an app on multiple devices is often referred to as mass deployment. The acronyms in this area are often used at random but an overall term for this area is normally referred to as Enterprise Mobility Management. According to Microsoft the following definition applies.

Enterprise Mobility Management (EMM):

- Mobile Device Management (MDM)
- Mobile Application Management (MAM)
- Mobile Content Management (MCM)
- Mobile Information Management (MIM)

EMM tools are typically a combination of deployment of on-device applications, configurations, corporate policies and certificates, and backend infrastructure, for the purpose of simplifying and enhancing the IT management of end user devices. Through out this document the term MDM is used even though the topics also covers MAM.

WiseMo delivers state of the art remote control software. If remote control should be placed in one of the above groups it would be the MDM group but often remote control is not part of an EMM solution and hence WiseMo remote control is an important addition to these tools.

This document describes how to use Microsoft Intune to deploy the WiseMo Host to various devices with different requirements. It also describes how to manage the configuration on deployed Hosts using Android Managed Configuration.

1.1 Prerequisites

This document focuses on using Microsoft Intune for the deployment and configuration of a WiseMo Host.

This tutorial requires a Microsoft Intune subscription and it assumes general knowledge on how to use Intune and how to enroll devices in Intune.

Enrollment of Android devices in Intune generally requires at least Android 6.

The document also assumes knowledge of what remote control is and in particular the role and usage of the WiseMo Host.

Throughout this document the term device is used. A device is simply some hardware running an operating system – in this case Android - such as a phone, tablet, display, control unit or computer.

2.0 Overview of WiseMo Host

The WiseMo Host is the module that is installed on the device that should be remotely controlled.

To check the version of an installed Host, click the WiseMo Host icon and select the Info tab:



Full remote control requires that the Host module is capable of capturing the device screen and able to simulating input such as keyboard, mouse and touch events.

The WiseMo Host uses different techniques for capturing the screen and simulating input depending on the device manufacture and in some cases the Android version. The techniques falls into the following categories:

- Using an API provided by manufacturer
- Using an Add-on component that is developed and provided by WiseMo but signed by the manufacturer
- Using Android built-in method for capturing the screen
- A combination of the above
- Rooted devices

The following paragraphs gives an overview of the techniques and examples of manufactures that uses a particular technique.

2.1 Using an API provided by manufacturer

Samsung is an example of a manufacturer providing a built-in API for remote control of their devices. The remote control API is part of Knox framework which is installed on most devices, see [Knox supported devices](#).

Please refer to paragraph 5.0 *Deployment of the WiseMo Host to a Samsung device* to deploy the Host to a Samsung device using the Knox method.

2.2 Using an Add-on component

Capturing the screen and simulating input are restricted operations that a normal app isn't capable of doing.

The add-on technique uses an intermediate component (the add-on) to bridge access from the WiseMo Host App to the Android operating system. The Add-on component must be signed by the manufacturer of the Android device. Examples of devices using the add-on technique includes, Lenovo, Huawei, LG, Honeywell – and also Samsung until Android 9. To see a list of available add-ons, please see [WiseMo Add-ons](#)

Most add-ons are available via Google Play but due to various reasons some can only be downloaded directly from WiseMo. It is recommended to install an add-on from Google Play if it is available.

Please refer to paragraph 4.0 *Deployment to a device using an Add-on component* to deploy the Host to a device using the add-on method.

2.3 Using Android built-in method for capturing the screen

From version 5 (Lollipop) the Android operating system has had a built-in method for capturing the screen. This method does not provide a method for simulating input.

Please refer to paragraph 6.0 *Deploy the WiseMo Host to a device using built-in method for capturing the screen* to deploy the Host using the built-in capture method.

2.4 A combination of the above

From version 5 (Lollipop) the Android operating system has had a built-in method for capturing the screen. This method does not provide a method for simulating input. Therefore a manufacturer like Zebra provides an API to simulate input.

Please refer to paragraph 7.0 *Deploy the WiseMo Host to a Zebra device* to deploy the Host to a Zebra device.

2.5 Rooted devices

There are different ways to root a device but generally the root mechanism offers the ability to assign the necessary permissions to the Host that it needs to capture the screen and simulate input. In case of a rooted device only the Host should be deployed.

The table below lists the requested permission in the manifest necessary for full remote control support:

Permission name	Purpose
<code><uses-permission android:name = "android.permission.READ_FRAME_BUFFER" /></code>	To be able to capture the screen

<code><uses-permission android:name = "android.permission.INJECT_EVENTS" /></code>	To be able to simulate input events
<code><uses-permission android:name = "android.permission.SHUTDOWN"></code>	To be able to shut down the device. Can be omitted on request.
<code><uses-permission android:name = "android.permission.REBOOT"></code>	To be able to restart the device. Can be omitted on request.

WiseMo does not encourage rooting, and rooting is hence outside the scope of this document.

3.0 Intune in general

Some general principles about Microsoft Intune is common to all deployment and will covered in this paragraph.

According to Microsoft, Intune is a part of Microsoft Endpoint Manager and provides the cloud infrastructure, the cloud-based mobile device management (MDM), cloud-based mobile application management (MAM), and cloud-based PC management for your organization.

Intune provides a comprehensive set to tools for deployment of on-device applications, configurations, corporate policies and certificates, and backend infrastructure. This guide will only focus on the deployment of the WiseMo Host and the different steps depending on the method for capturing the screen and simulating input.

To sign up for Microsoft Intune and create your first users and groups please review [Intune Quick Start](#). To read more about Microsoft Intune please see [Intune Walkthrough](#)

Open Intune browser and login with your credentials. From the navigation pane to the left, select Dashboard to display details about the devices and apps in your Intune tenant. If you are starting with a new Intune tenant¹, you will not have any enrolled devices yet.

If the devices you want to deploy the WiseMo Host onto is not enrolled yet the first step will be to enroll them in Intune.

¹ A tenant is an instance of Azure Active Directory (Azure AD). Your subscription to Intune is hosted by an Azure AD Tenant.

3.1 Enroll a device in Intune

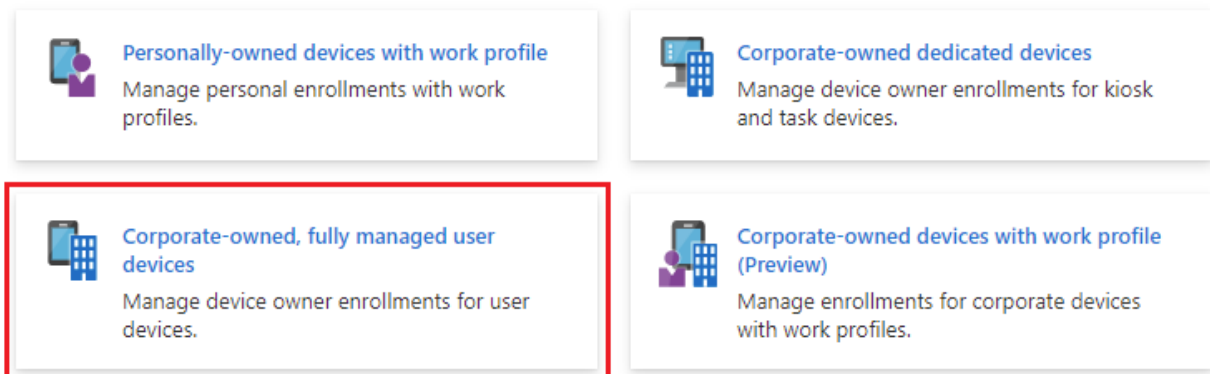
There are several methods to enroll a device into Intune. Each method depends on the device's ownership (personal or corporate), device operating system, and management requirements.

For different ways to enrol an Android device in Intune please refer to [What is device enrollment in Intune](#).

It is outside the scope of this document to describe the various enrollment methods available, so only enrollment with a token of Android Enterprise fully managed devices will be described here, see also the Intune help [here](#) for more details.

Before enrolling the device prepare Intune for enrollment like this:

1. From the navigation pane in Intune, select **Devices > Android > Android enrollment**
2. If Intune isn't already linked to your managed Google Play account, click **Link your managed Google Play account to Intune**. Follow the instructions in the pane that opens to the right.
3. Then select **Corporate-owned, fully managed user devices** under **Enrollment profiles**.



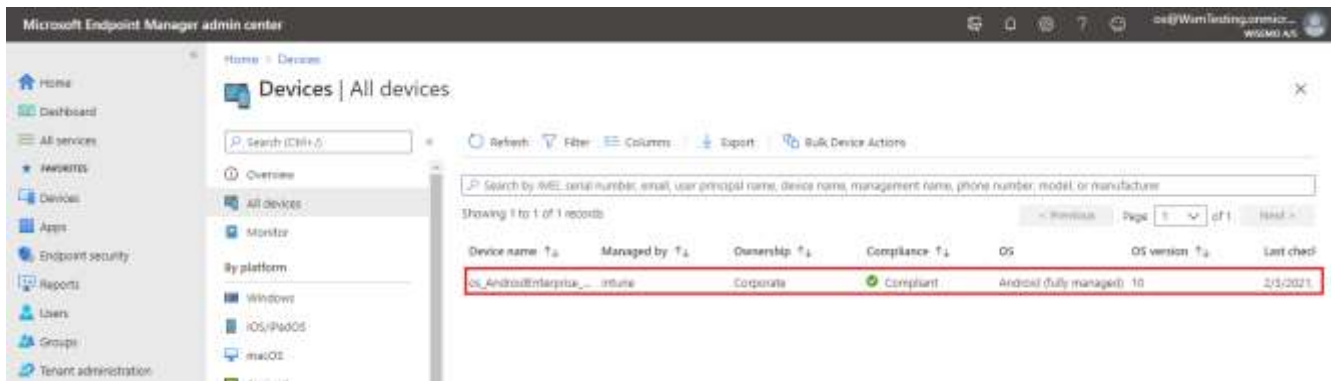
A pane should open to the right showing a QR code and a Token (partly masked out):



Now follow these steps on the device you want to enroll (details might vary depending on Android version):

1. If it is not a new device, factory reset it from Android settings.
2. Turn on the device. On the Welcome screen, select your language.
3. Connect to your Wifi, and then choose **Next**.
4. Accept the Google Terms and conditions, and then choose **Next**.
5. On the Google sign-in screen, enter **afw#setup** instead of a Gmail account, and then choose **Next**.
6. Accept and continue or click Next in the following Google/device manufacturer screens
7. On the **Enroll this device screen**, allow your device to scan the QR code (Android 6.1 or later) or choose to enter the token manually (Android 6.0 or later). Scan the QR code or enter the Token you prepared in Intune above.
8. Follow the on-screen prompts to complete enrollment. When asked to log in to microsoft, login with an Intune user account.

Your device should now be enrolled in Intune. From the navigation pane in Intune, select **Devices > All Devices** to display details about the enrolled devices. You should be able to see the device you just enrolled:



Click on the device and select **Properties**. Give it a meaningful full name in **Management name**:

os_AndroidEnterprise_2/3/2021_2:51 PM | Properties

Search (Ctrl+/)

Save Discard

Overview

Manage

Properties

Monitor

Hardware

Discovered apps

Device compliance

Device configuration

Device name

os_AndroidEnterprise_2/3/2021_2:51 PM

Management name *

Android Device 1

Device category

Unassigned

Device ownership

Corporate

Primary user

And save the changes.

To use Samsung's Knox Mobile Enrollment, the device must be running Android 6 or later and Samsung Knox 2.8 or higher. For more information, learn how to automatically enroll your devices with [Knox Mobile Enrollment](#).

3.2 Use Intune groups

From the navigation pane, select **Groups** to display details about the Azure Active Directory (Azure AD) groups included in Intune. As an Intune admin, you use groups to manage devices and users.

Select All Groups and click **+ Add new group** in the menu. Fill out the fields and give it the name "Android Devices":

New Group

Group type * ⓘ
 Security

Group name * ⓘ
 Android devices ✓

Group description ⓘ
 All Android devices ✓

Membership type * ⓘ
 Assigned

Owners
 No owners selected

Members
 No members selected

Click **No Memebers selected**. A pane opens to the right. Scroll down and locate the Android device you just enrolled. The names are not very user friendly so you might go back to devices to find the name of the device. Devices are indicated with the following image:

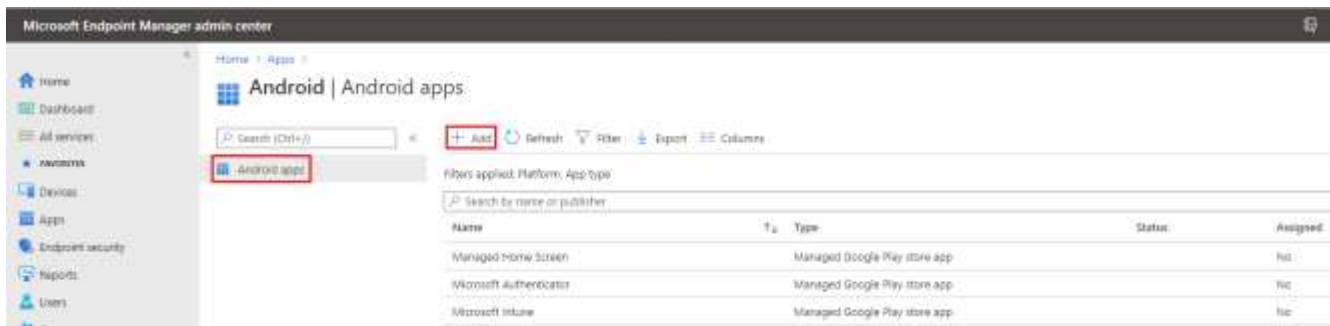


Please note that only the first 50 items are shown so you might have to search to limit the number of items shown. When you have found the device, click **Select** and then **Create** to create the group that we will use in the next paragraph.

3.3 Add apps to Intune

Before you assign an app to a device or a group of users, you must first add the app to Microsoft Intune.

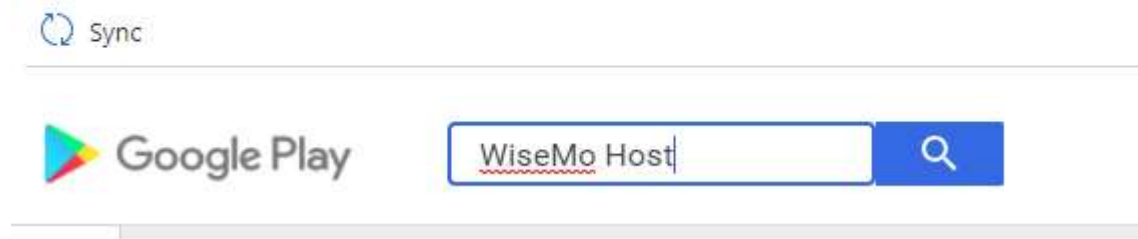
From the navigation pane, select **Apps > Android** to display an overview of Android apps and their status. Even in a fresh Intune installation you will see some Microsoft apps:



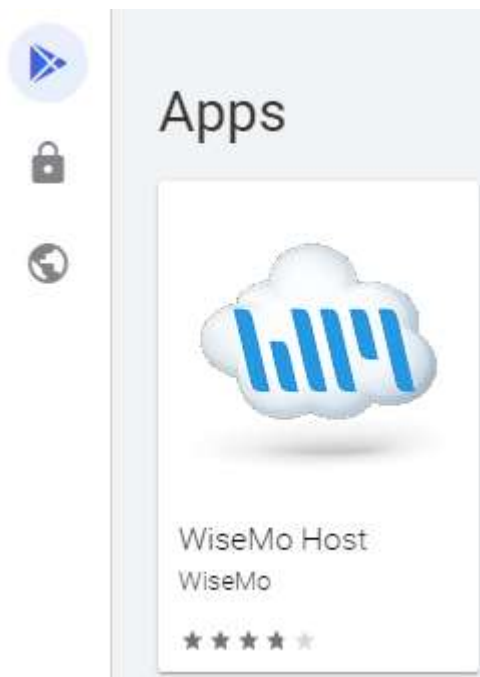
To add an app do the following:

1. Click **Add** from the menu.
2. In the **Select app type** pane, under the available **Store app** types, select **Managed Google Play app**.
3. Click **Select**.
4. Intune will open Google Play. In the search field, write **WiseMo Host** and click the search symbol:

Managed Google Play

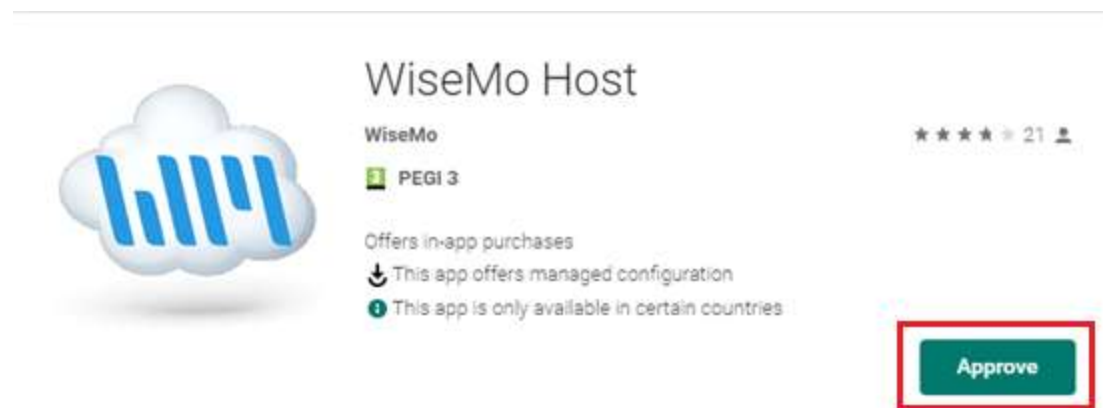


5. select the Host app by clicking on it:

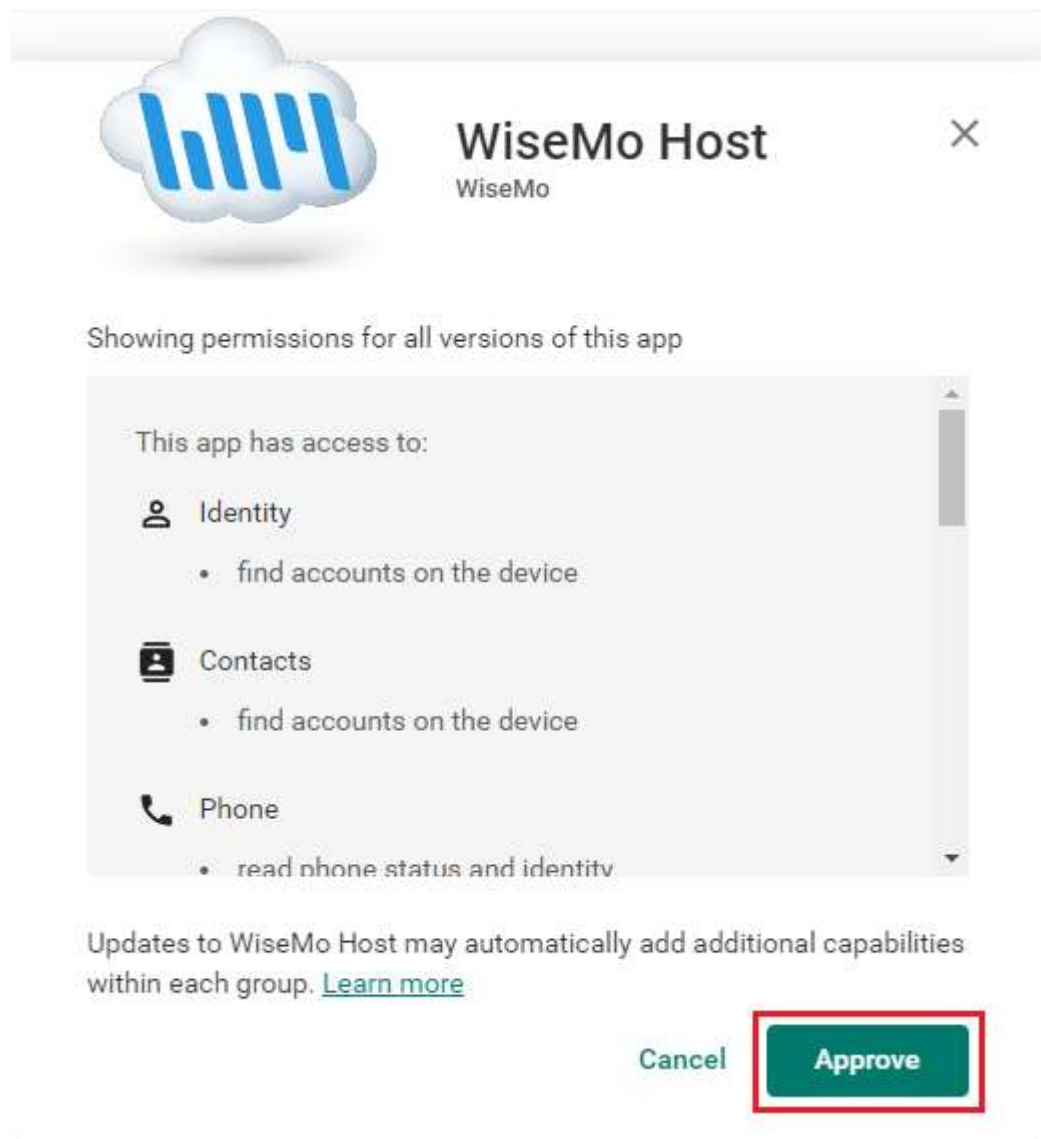


You will now go through a number of approval steps.

6. In the window that appeared click the **Approve** button:




and **Approve** again:



Then select Keep approved when app requests new permissions and click Done:

Approval Settings

Notifications



WiseMo Host

WiseMo

How would you like to handle new app permission requests?

☒ Keep approved when app requests new permissions.
 Users will be able to install the updated app.

☐ Revoke app approval when this app requests new permissions.
 App will be removed from the store until it is reapproved.

Done

And finally, now the WiseMo Host has been approved, click **Select** to select the app (the button should be highlighted):



WiseMo Host

WiseMo





Offers in-app purchases

 This app offers managed configuration

 This app is only available in certain countries

Select

Unapprove

Approval Preferences

Click **Sync** button at the top of the pane to synchronize the app with the Managed Google Play service.

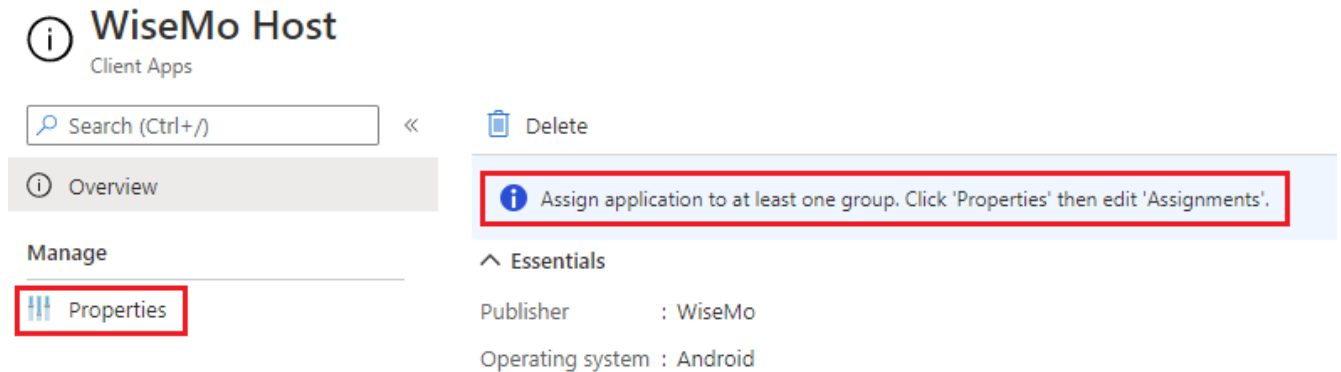
- Click **Refresh** to update the app list and display the WiseMo Host app. Sometimes it's slow and you might have to click **Refresh** several times.

8. When Intune has been synchronized with Google Play service, you should see the Host in the list:



Name	Type	Status	Assigned
WiseMo Host	Managed Google Play store app		Yes

9. The WiseMo Host app must now be assigned to one or more groups. Start by clicking the WiseMo Host app in the list.
10. In the screen that appears, select **Properties**:



WiseMo Host
Client Apps

Search (Ctrl+/) << Delete

Overview

Manage

Properties

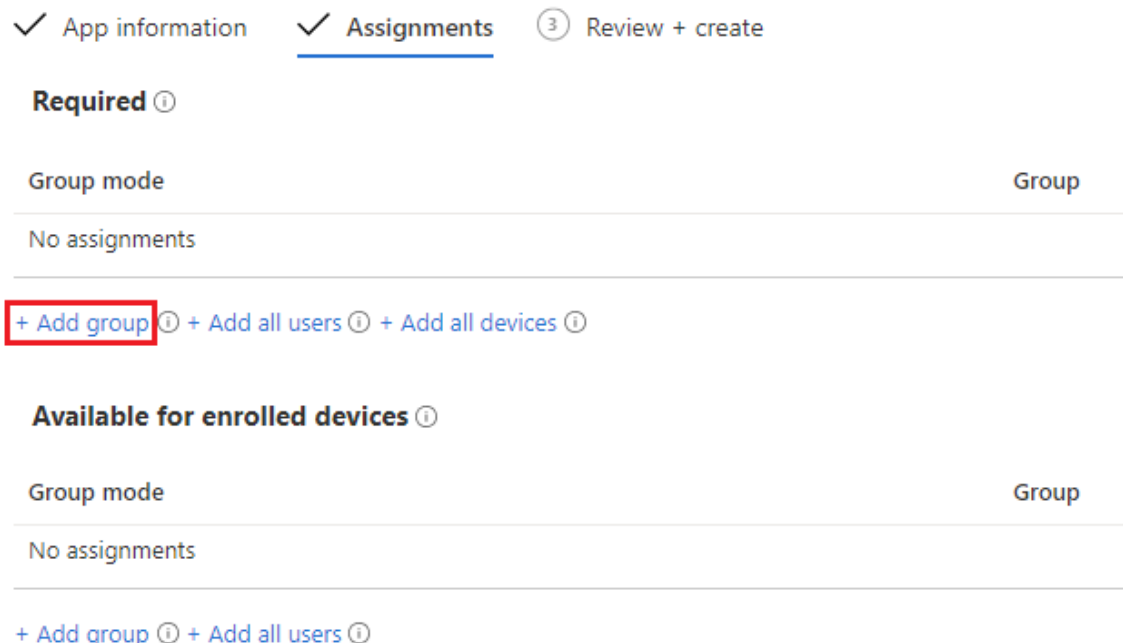
Assign application to at least one group. Click 'Properties' then edit 'Assignments'.

Essentials

Publisher : WiseMo

Operating system : Android

11. Scroll down and click **Edit** next to Assignments.
12. Click the **+ Add group** in group assignments:



✓ App information ✓ **Assignments** ③ Review + create

Required ⓘ

Group mode Group

No assignments

+ Add group ⓘ + Add all users ⓘ + Add all devices ⓘ

Available for enrolled devices ⓘ

Group mode Group

No assignments

+ Add group ⓘ + Add all users ⓘ

In the pane that appears to the right, select the group **Android devices** we created in the previous paragraph:

Select groups



Azure AD groups



Click **Select**

Do not add users unless you are sure what hardware they will be using.

For more information, see [Add groups to organize users and devices](#).

13. Click the **Review + save** button. Review the values and settings you entered for the app.
14. When you are done, click **Save**.

After some time the Host should appear on the Android device(s) you added to the **Android devices** group.

3.4 Managed configuration and app permissions

Android Managed Configuration is referred to as **App configuration policies** in Intune. App configuration policies helps you to setup up the configuration for an app. The configuration is supplied automatically to the app on the end-users device, and end-users don't need to take action.

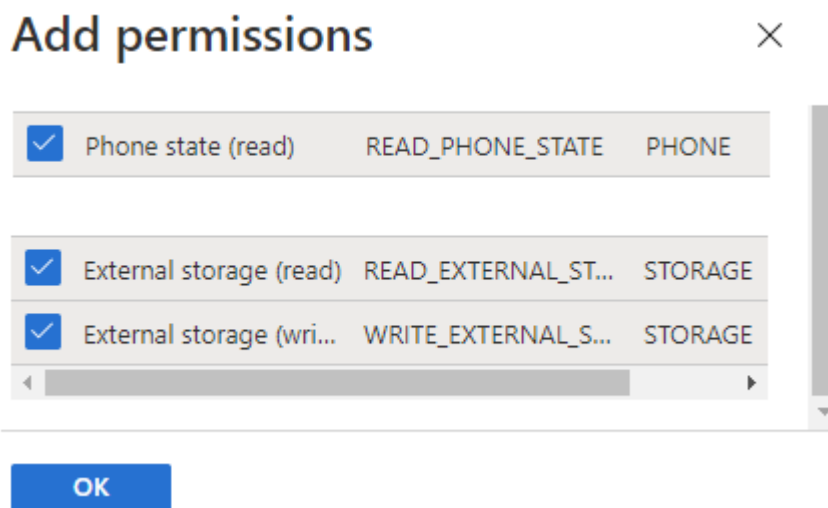
The other topic for this paragraph is predefined app permissions because it is configured in the same place as configuration settings. By predefined app permissions the user is alleviated from accepting permissions prompts from the app on the device.

Follow these steps to Create an app configuration policy and to preassign app permissions for the WiseMo Host:

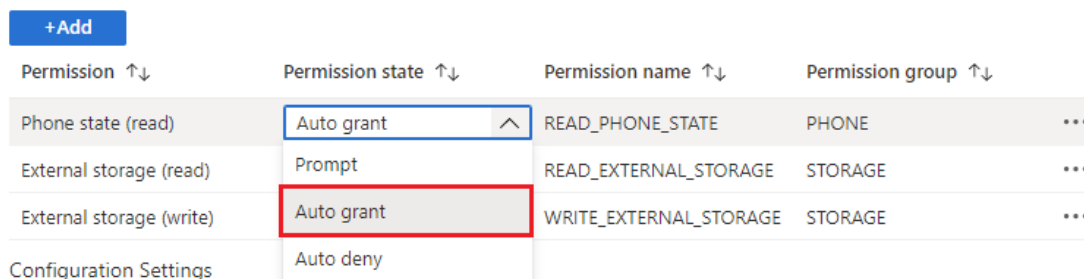
1. Choose the **Apps > App configuration policies > Add > Managed devices**. Note that you can choose between Managed devices and Managed apps.
2. On the Basics page, set the following details:
 - **Name** – Enter for example WiseMo Host Managed Configuration.
 - **Description** - The description of the profile that appears in the Intune.
 - **Device enrollment type** - This setting is set to Managed devices.
3. Select **Android Enterprise** as the **Platform**.
4. Select **All Profile Types** as the **Profile Type**.

5. Click **Select app** next to Targeted app. The Associated app pane is displayed. On the **Associated app** pane, choose the **WiseMo Host** and click OK.
6. Click **Next** to display the **Settings** page.
7. Click **Add** to display the **Add permissions** pane.
8. Click the permissions that you want to override. Permissions granted will override the "Default app permissions" policy for the selected apps. For the WiseMo Host you should enable:
 - Phone state (read)
 - External storage (read)
 - External storage (write)

Click **Ok** to continue:



9. Set the Permission state to **Auto grant** for each permission. You can choose from Prompt, Auto grant, or Auto deny.



10. Because the WiseMo Host supports configuration settings, the Configuration settings format dropdown box is visible. Intune offers a simple configuration design but it is too² simple for the Host configuration and you should therefore select **Enter JSON data**. Please refer to *Appendix A – Managed Configuration* for a description of keys and values.
11. In the editor, you can define JSON values for configuration settings. Click **Download JSON Template** to download a sample file that you can then configure. The settings consists of a key and a value. When

² The Intune configuration design doesn't support configuration types such as the Bundle.

editing this file it is crucial that only the values are edited., i.e. valueString, ValueInteger and valueBool. To set for example the password for Share Password mode, find **"key": "sharedPassword"** and modify the value string like this: **"valueString": "ASDF"** to set the password to ASDF. You might also want to change whether the Confirm Access security prompt should be shown on the Host before a connection. Find **"key": "sharedPasswordConfirmAccess"** and modify the value to false like this **"valueBool": false**

12. To configure the WiseMo Host name you should edit the JSON structure:

```
"key": "hostNaming",
"valueBundle": {
  "managedProperty": [
    {
      "key": "hostNamingMode",
      "valueString": "naming_mode_enter_or_leave_blank"
    },
    {
      "key": "hostNameSpecific",
      "valueString": "{{userprincipalname}}"
    }
  ]
}
```

The Host can be configured to the following naming modes:

- **naming_mode_computername**: Default. No additional configuration is necessary
- **naming_mode_enter_or_leave_blank**: specify the name in the valueString for HostNameSpecific.
- **naming_mode_imei_or_serial_number**: The IMEI (Android 9 and older) or the device serial number.

If you want to specify a name you would have to make sure the name is unique because you would otherwise not be able to identify which device was which. Therefore it's a good idea to use the Intunes variables, see tokens that can be used here [Supported variables for configuration values](#). To use Intunes **User Principal Name** you should specify **{{userprincipalname}}**.

13. To configure the myCloud profile for the device you should first login to your myCloud account and go to **Settings > Connection**:

Default connection account

Guest and Host must belong to the same domain to be able to connect to each other. All installation packages you download from the [Deploy](#) tab are preconfigured with the necessary parameters for the communication profiles. However you can configure the communication profile manually on the Guest and Host by using the following communication profile parameters:

```
myCloud Service URL: http://mycloud.wisemo.com/cm
Domain name: WiseMo
Account name: DefaultConnection
Password: ***** (Show)
```


Click Show next to **Password** to see the password. Fillout the JSON myCloud profile with following values for a **myCloudDomain** and **myCloudPassword** (in the example below the myCloud domain name is WiseMo and the myCloud password is 1234AbCd (this is not a user account password)):

```
"key": "myCloudProfile",
"valueBundle": {
  "managedProperty": [
    {
      "key": "myCloudUrl",
      "valueString": "http://mycloud.wisemo.com/cm"
    },
    {
      "key": "myCloudAccount",
      "valueString": "DefaultConnection"
    },
    {
      "key": "myCloudDomain",
      "valueString": "WiseMo"
    },
    {
      "key": "myCloudPassword",
      "valueString": "1234AbCd"
    }
  ]
}
```

14. The Host will also need to be license. A Host can be either licensed via a valid myCloud account or with a license key. To configure the Host for myCloud licensing edit the JSON structure like this:

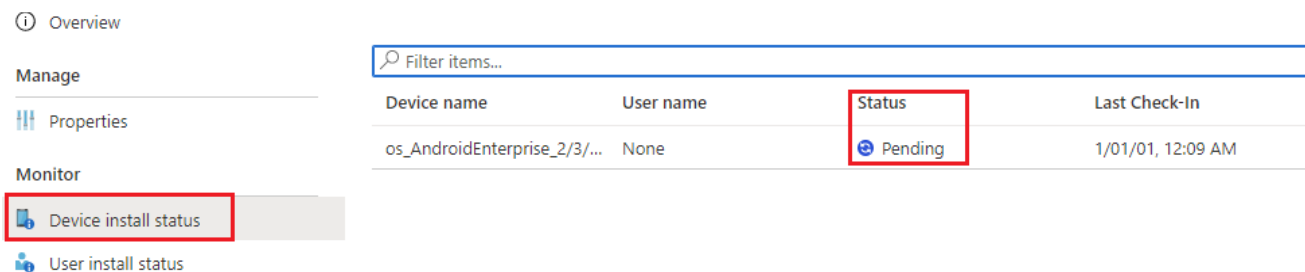
```
"key": "hostLicense",
"valueBundle": {
  "managedProperty": [
    {
      "key": "hostLicenseMode",
      "valueString": "license_mode_mycloud"
    },
    {
      "key": "hostLicenseKey",
      "valueString": ""
    }
  ]
}
```

15. When your are done, click **Next** to display the **Assignments** page.
16. On the **Assignment** page click **Add groups** and select the **Android devices** we created earlier.
17. Click **Next**.

18. Review the configuration and click **Create**.

The configuration and permissions will be applied to the WiseMo Host at first run. You can later modify the configuration and permission settings. Such changes will be deployed to the WiseMo Host automatically.

To see whether the policy has been deployed to a specific device, select the policy and click **Device install status**:



Device name	User name	Status	Last Check-In
os_AndroidEnterprise_2/3/...	None	Pending	1/01/01, 12:09 AM

Wait for the **Status** to switch to Success. When the Host receives a new configuration it will restart itself to apply the new settings (if the Host is connected to a Guest, it will restart after the connection).

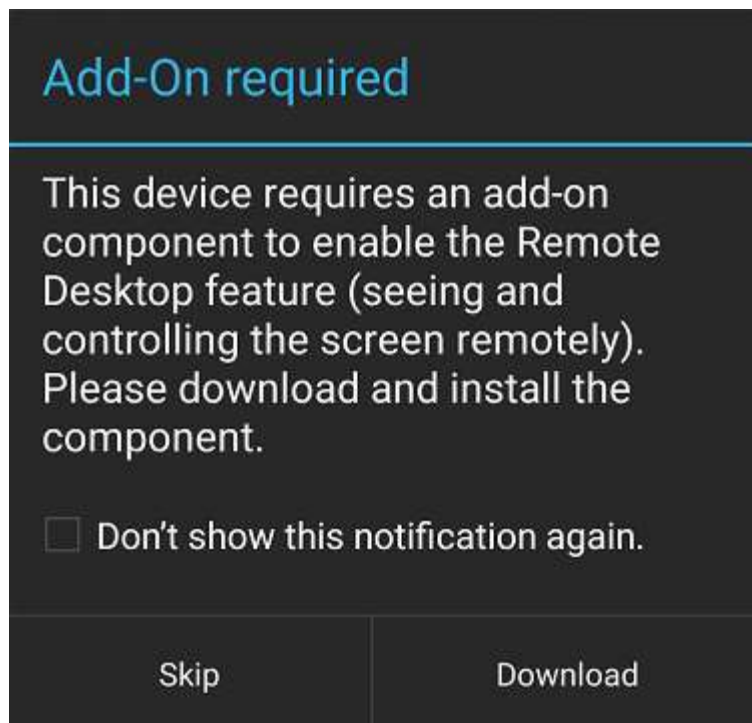
4.0 Deployment to a device using an Add-on component

The add-on technique uses an intermediate component (the add-on) to bridge access from the WiseMo Host App to the Android operating system. The Add-on is an extra app that needs to be installed on the device and hence the Add-on app is deployed in the same way as the Host app is deployed.

4.1 Choosing the right Add-on

While the Host app is generic the Add-on app is specific to each manufacturer of the device. Sometimes there are multiple Add-ons for a manufacturer depending on the specific device. It is *very* important to install the correct Add-on on a device – failure to do so will result in black screen on the Guest when trying to view the remote desktop. If an erroneous Add-on is installed, simply uninstall it again. Installing two Add-ons on the same device will also result in undetermined behavior.

The simplest way to find the correct Add-on for a device is to make a manual Host installation. If there's an Add-on available, the Host will suggest to download and install it (exit the Host from the menu and restart it).



Click **Download** to go to Google Play:



Make a note of the name and make sure to specify the same Add-on when configuring it in Intune.

4.2 Deploying the Add-on

Add the Add-on app that you have verified works on the device to Intune in the same way as you did in paragraph 3.3 *Add apps to Intune*. There are no permissions or managed configuration that should be set for the Add-on app.

In paragraph 3.2 *Use Intune groups* we created the group **Android Devices**. If you have only one type of Android devices you can assign the the Add-on to this group. If you have multiple devices or expect to get

multiple devices it makes sense to create a new group that you could give an informative name like **Android LG Devices**. Assign the Add-on app to this group and after some time the Add-on app will be deployed to the devices.

5.0 Deployment of the WiseMo Host to a Samsung device

Samsung is an example of a device that uses a built-in API for capturing the screen and simulating input. This Samsung API is called Knox and it provides a huge set of MDM features specific to Samsung devices.

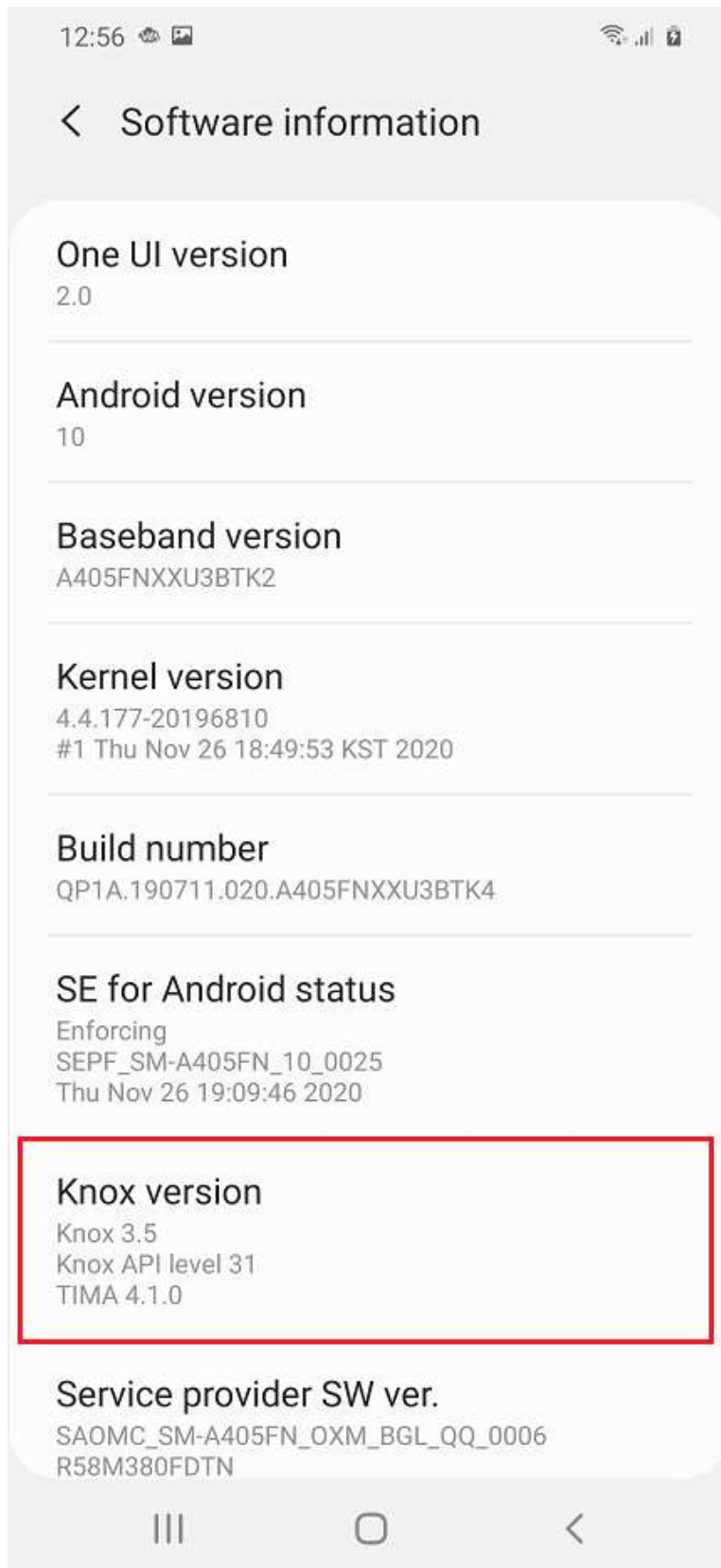
5.1 Add-on method

For Android 9 and older there's also an Add-on available and the method in *4.0 Deployment to a device using an Add-on component* should be used. Using the Add-on method might be preferred because it doesn't involve user interaction on the device on first run or the Add-on can be used on Samsung devices where the Knox API isn't available (see <https://www.samsungknox.com/en/knox-platform/supported-devices>).

5.2 Deploying the WiseMo Host using the Knox API method

This paragraph describes how to install the WiseMo Host on Samsung devices using the Knox API.

The forementioned link <https://www.samsungknox.com/en/knox-platform/supported-devices> specifies which devices support Knox and the required Knox version that must be available on the Samsung device. To see the Knox version and other version information that might be relevant, go to **Settings > About phone > Software Information** and locate the Knox version:



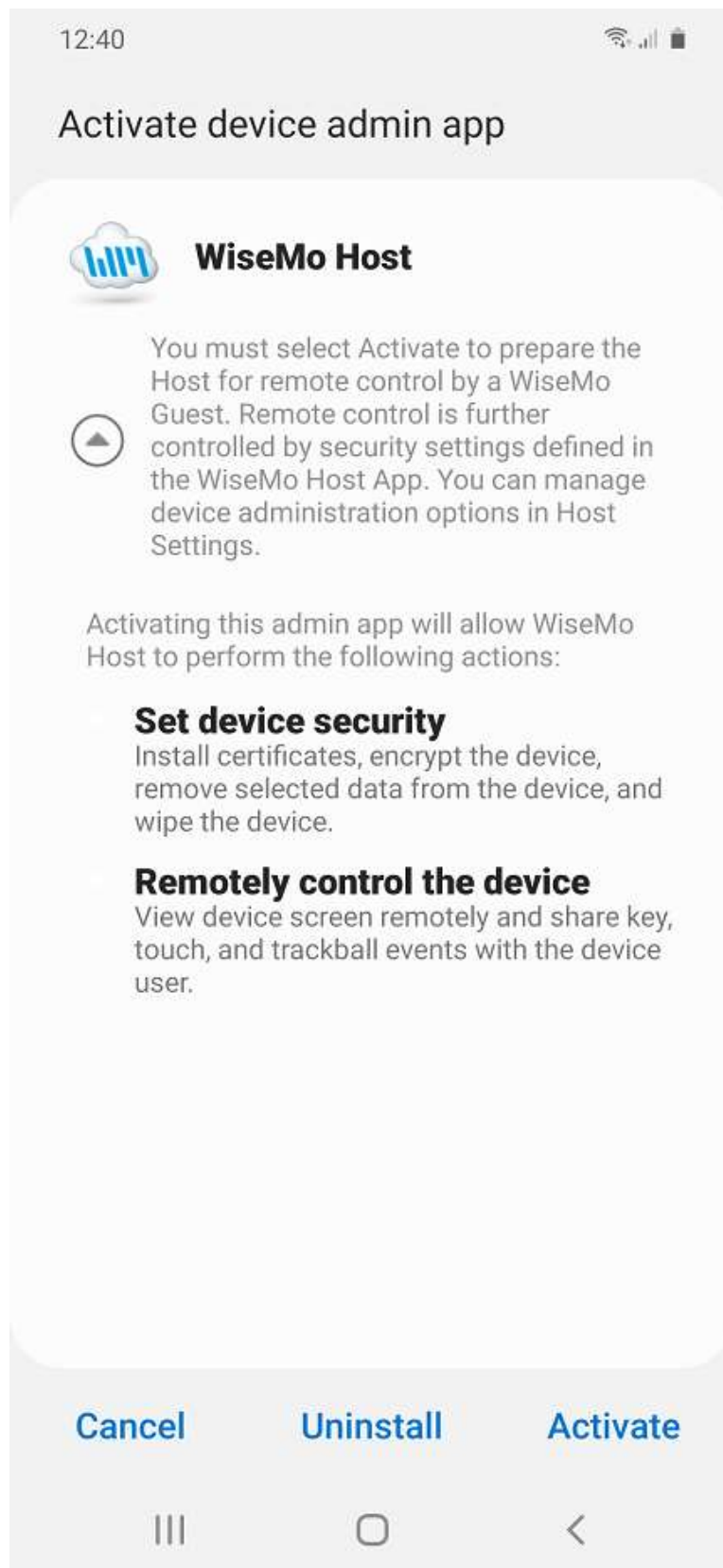
A Samsung device can be enrolled in Intune as described in paragraph 3.1 *Enroll a device in Intune*. A Samsung device can also be enrolled using Samsung Knox Mobile Enrollment (KME). Using Intune with Samsung KME, you can enroll large numbers of company-owned Android devices when end users turn on their devices for the first time and connect to a WiFi or cellular network. Also, devices can be enrolled using Bluetooth or NFC when using the Knox Deployment App. Please refer to <https://docs.microsoft.com/mem/intune/enrollment/android-samsung-knox-mobile-enroll>.

When the Samsung device is enrolled in Intune, you simply add the Samsung device to the **Android Devices** group.

Select **Groups > All Groups** and click and click on our group **Android Devices** and then **Members**. Click **+ Add members** in the menu. Find the Samsung device in the right pane that pops up and click the **Select** button.

Because you added the new Samsung device to a group that already had the WiseMo Host assigned to it, the WiseMo Host is automatically deployed to the device.

When you run the Host on the device the first time, the following prompt will appear:



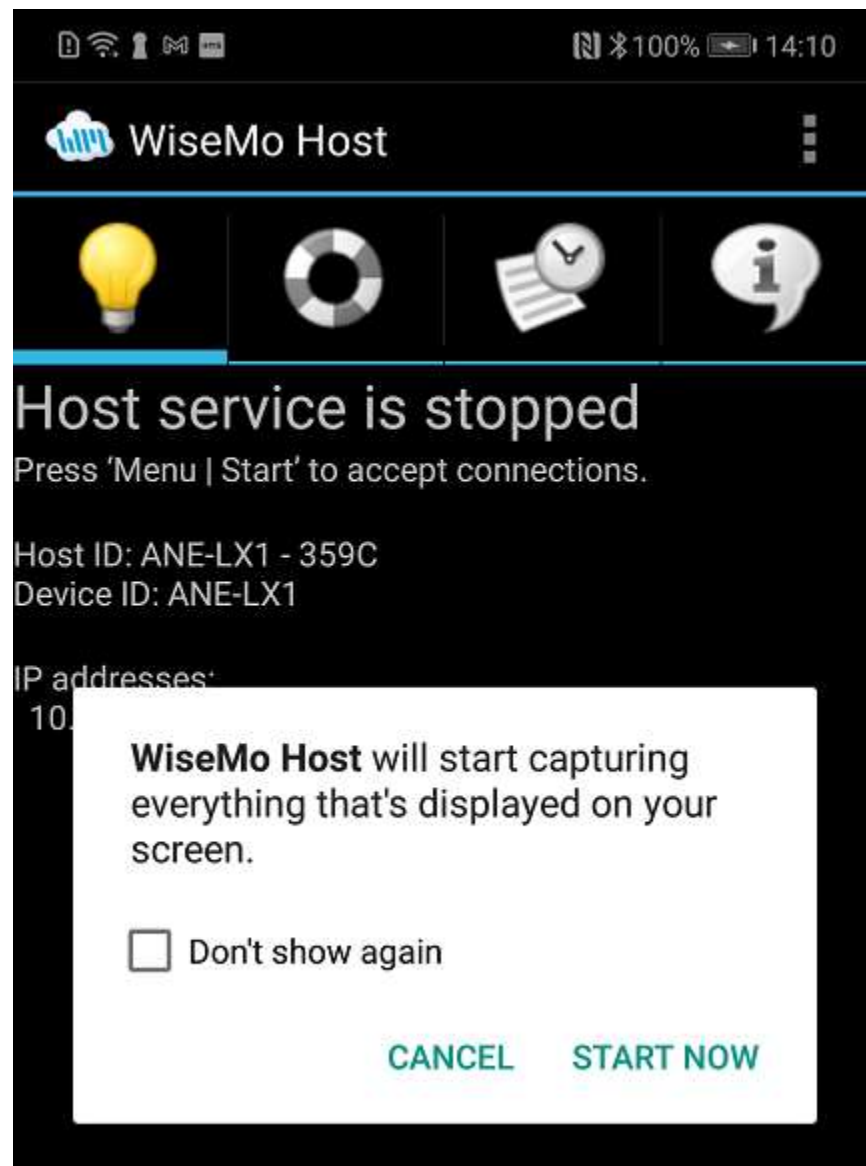
Click Activate and the Host is ready to be remote controlled with the Managed configuration that was assigned in paragraph 3.4 *Managed configuration and app permissions*.

6.0 Deploy the WiseMo Host to a device using built-in method for capturing the screen

From version 5 (Lollipop) the Android operating system has had a built-in method for capturing the screen. This method does not provide a method for simulating input. Therefore this method should only be used as the last method.

The WiseMo Host is deployed and configured like it is described in chapter 3.0 *Intune in general*.

After the Host has been deployed and the Host is started for the first time, the built-in capture method will prompt for permission to capture the screen:



Check the **Don't show again** check box and click **START NOW**. On Android 10 and later the **Don't show again** check box isn't available so the Host will show the prompt every time the Host is started.

The capture permission is unfortunately not a permission that can be preassigned in Intune or any other MDM tool.

WiseMo is always willing to investigate whether it is possible to get the manufacturer sign a WiseMo Add-on for the device in question.

7.0 Deploy the WiseMo Host to a Zebra device

This paragraph is not finished.

The WiseMo Android Host (version 18 and above) fully supports remote control of Zebra Android scanner devices if the device has MX 8.3 (Mobility Extensions) or greater installed. If a device uses an older MX version, please verify whether there's an upgrade available.

A Zebra device running Android 5.0 or newer can be remotely viewed but it requires additional configuration for full remote desktop control where also keyboard and touch events can be emulated.

Please refer to the detailed description [here](#).

7.1 Enroll a Zebra in Intune

Zebra devices can be enrolled in Intune as:

- **Android Enterprise** devices
- **Android device administrator** devices

7.2 Assigning the Zebra device to groups

When the device is enrolled, assign it to the **Android Devices** group.

1. Go to **groups > All groups** and select the **Android Devices** group.
2. Select **Members** and click **Add members** in the toolbar.
3. Find the Zebra device and click **Select**.

This will automaticall deploy the WiseMo Host to the device after some time.

In the following you'll need a group to assign a Zebra configuration policy to. If you have only one type of Android devices you can use the **Android Devices** group. If you have multiple devices or expect to get multiple devices it makes sense to create a new group that you could give an informative name like **Android Zebra Devices**. Follow the procedure described in 3.2 *Use Intune groups* we created to create a new group if necessary.

7.3 Create a profile in Intune – for Android device administrator devices

In order to simulate input using the WiseMo Host, the local Zebra client on the device, StageNow, needs to be configured to allow such input.

This guide shows you how to configure Zebra Mobility Extensions (MX) on Zebra devices enrolled as Android device administrator devices. For Android Enterprise devices, use [OEMConfig](#).

In Intune, create a device configuration profile:

4. Go to **Devices > Configuration profiles** and select **Create profile**.
5. Enter the following properties:

Platform: Select **Android device administrator**.

Profile: Select **MX profile (Zebra only)**.



Select **Create**.

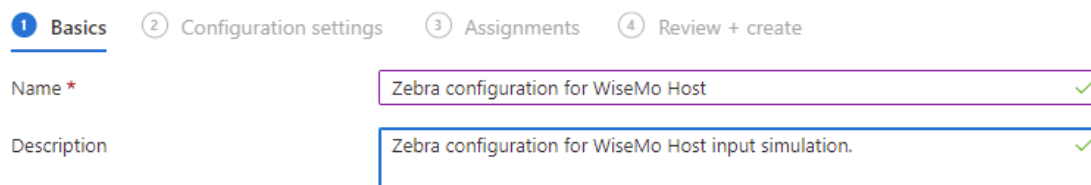
6. In **Basics**, enter the following properties:

Name: Enter a descriptive name for the new profile, e.g. Zebra configuration for WiseMo Host.

Description: Enter a description for the profile. This setting is optional, but recommended.

MX profile (Zebra only)

Android device administrator



Select **Next**.

7. Download the WiseMo configuration file [here](#) and
8. In **Configuration settings > Choose a valid Zebra MX XML file**, and specify WiseMo configuration file you just downloaded.

MX profile (Zebra only)

Android device administrator

✓ Basics **2 Configuration settings** ③ Assignments ④ Review + create

[Learn more about managing Zebra devices](#)

MX profile in .xml format *

Not configured

Choose a valid Zebra MX XML file *

"WiseMoHostRCZebra.xml"



```

1 <wap-provisioningdoc>
2   <characteristic version="8.3" type="AccessMgr">
3     <parm name="OperationMode" value="1" />
4     <parm name="ServiceAccessAction" value="4" />
5     <parm name="ServiceIdentifier" value="com.zebra.eventinjectionservice" />

```

When done, select **Next**.

Please note that for security reasons, you can't see the profile XML text after you save it. The text is encrypted, and you only see asterisks (****).

9. In **Scope tags** (optional) > **Select scope tags**, choose your scope tags to assign to the profile. For more information, see [Use RBAC and scope tags for distributed IT](#).

Select **Next**.

10. In **Assignments**, click **Add groups** and select the group, **Android Zebra devices** you defined above. Select **Next**.

11. In **Review + create**, when you're done, choose **Create**. The profile is created, and shown in the list.

You can also [monitor its status](#).

The next time the device checks for configuration updates, the MX profile is deployed to the device. Devices sync with Intune when devices enroll, and then approximately every 8 hours. You can also [force a sync in Intune](#). Or, on the device, open the **Company Portal app** > **Settings** > **Sync**.

Appendix A – Managed Configuration

The WiseMo Host supports Android Managed Configuration which can be configured in most MDM tools.

The basis for Managed Configuration is a JSON file that can either be modified directly or via built-in UI in the MDM tool. This appendix describes the WiseMo Hosts Managed Configuration JSON file.

The JSON file consists of a number of sections with keys and corresponding value. When editing this file it is crucial that only the values are edited, i.e. valueString, ValueInteger and valueBool. To set for example the password for Share Password mode, find "key": "sharedPassword" and modify the value string like this: "valueString": "ASDF" to set the password to ASDF.

The file looks like this with its default values inserted:

```
{
  "kind": "androidenterprise#managedConfiguration",
  "productId": "app:com.wisemo.host.v10",
  "managedProperty": [
    {
      "key": "authMode",
      "valueString": "shared_password"
    },
    {
      "key": "sharedPasswordSection",
      "valueBundle": {
        "managedProperty": [
          {
            "key": "sharedPassword",
            "valueString": ""
          },
          {
            "key": "sharedPasswordConfirmAccess",
            "valueBool": true
          }
        ]
      }
    },
    {
      "key": "myCloudProfile",
      "valueBundle": {
        "managedProperty": [
          {
            "key": "myCloudUrl",
            "valueString": "http://mycloud.wisemo.com/cm"
          },
          {
            "key": "myCloudAccount",
```

```

        "valueString": "DefaultConnection"
    },
    {
        "key": "myCloudDomain",
        "valueString": ""
    },
    {
        "key": "myCloudPassword",
        "valueString": ""
    }
]
}
},
{
    "key": "hostNaming",
    "valueBundle": {
        "managedProperty": [
            {
                "key": "hostNamingMode",
                "valueString": "naming_mode_computername"
            },
            {
                "key": "hostNameSpecific",
                "valueString": ""
            }
        ]
    }
},
{
    "key": "hostLicense",
    "valueBundle": {
        "managedProperty": [
            {
                "key": "hostLicenseMode",
                "valueString": "license_mode_dont_change"
            },
            {
                "key": "hostLicenseKey",
                "valueString": ""
            }
        ]
    }
},
{

```

```

    "key": "stopHostIfNoActivity",
    "valueInteger": 0
  }
]
}

```

Table of keys and values:

Key	Type	Values	Comment
authMode	valueString (choice)	shared_password username_and_password	The user names and passwords cannot be configured for the username_and_password mode and must already exists in the host configuration file (host.xml).
sharedPassword	valueString		By default no password
sharedPasswordConfirmAccess	valueBool (choice)	true false	
myCloudUrl	valueString		By default: http://mycloud.wisemo.com/cm
myCloudDomain	valueString		Your myCloud domain name
myCloudAccount	valueString		By default: DefaultConnection
myCloudPassword	valueString		The myCloud Connection account password. Login into myCloud and go to Settings > Connection and see the the settings for the Default connection account. This is not your myCloud user account.
hostNamingMode	valueString (choice)	naming_mode_computername naming_mode_enter_or_leave_blank naming_mode_altername naming_mode_imei_or_serial_number	
hostNameSpecific	valueString		The name when hostNamingMode is naming_mode_enter_or_leave_blank
hostLicenseMode	valueString (choice)	license_mode_dont_change license_mode_mycloud license_mode_key	Configure license mode. For license_mode_mycloud a valid myCloud account must be defined. For license_mode_key the

			hostLicenseKey must be defined.
hostLicenseKey	valueString		Set WiseMo license key for license_mode_key.
stopHostIfNoActivity	valueInteger		0: (default) means do not stop -1: do no change host config >0: number of seconds to stop host when idle (i.e. no connections)

When (choice) is specified in the **Type** column, only the values in the **Values** column are valid.