

## Intune Deployment and Configuration of the WiseMo Android Host

### 0.0 Table of Contents

Click in the table to jump directly to the paragraph.

0.0	Table of Contents .....	1
1.0	Overview.....	3
1.1	Prerequisites.....	3
2.0	Overview of WiseMo Host.....	4
2.1	Using an API provided by the manufacturer .....	4
2.2	Using a manufacturer specific Add-on component.....	5
2.3	Using the Universal Add-on component .....	5
2.4	Using Android built-in method for capturing the screen .....	5
2.5	A combination of the above .....	6
2.6	Rooted devices .....	6
3.0	Management tools in general .....	6
3.1	Android Enterprise .....	7
3.2	Android Enrollment .....	8
4.0	Intune in general .....	10
4.1	Link managed Google Play to Intune.....	10
4.2	Enroll a device in Intune .....	12
4.3	Intune groups .....	15
4.4	Add apps to Intune .....	15
4.5	Managed configuration and app permissions.....	19
4.6	Completing device installation .....	23
4.7	A note about device restrictions and blocking screen capture .....	23
5.0	Deployment to a device using an Add-on component.....	24
5.1	Choosing the right Add-on.....	24
5.2	Deploying the Add-on.....	25
6.0	Deployment to a device using the Universal Add-on component .....	26
6.1	Deploying the Universal Add-on.....	26
6.2	Enabling the Universal Add-on .....	27

6.3	Remote desktop controlling with the Universal Add-on .....	30
7.0	Deployment of the WiseMo Host to a Samsung device .....	31
7.1	Add-on method .....	31
7.2	Deploying the WiseMo Host using the Knox API method .....	32
8.0	Deploy the WiseMo Host to a device using built-in method for capturing the screen.....	36
9.0	Deploy the WiseMo Host to a Zebra device .....	38
9.1	Enroll a Zebra in Intune .....	38
9.2	Method and configuration files depend on Android version .....	38
9.3	Create groups for your Zebra device and assign devices .....	39
9.4	Create a configuration profile in Intune – for Android Enterprise devices .....	39
9.5	Create a profile in Intune – for Android device administrator devices.....	44
Appendix A – Managed Configuration .....		47

## 1.0 Overview

Automation of the installation and configuration of an app on multiple devices is often referred to as mass deployment but an overall term for this area is normally referred to as Enterprise Mobility Management (EMM).

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.

Remote Control is an EMM tool but often remote control is not part of an EMM solution and hence WiseMo remote control is an important addition.

When it comes to managing Android devices, all EMM solutions are based on Android Enterprise.

The WiseMo Android Host fully supports Android Enterprise. It supports Android Managed configuration for the most important Host settings, it allows both installation in the Personal and Work profile and it offers an API that can be used by the EMM Agent to Start, Restart and set configuration directly.

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.

This is not a document you need to read from the start to the end. Read paragraph *2.0 Overview of WiseMo Host* and then jump top the paragraphs that makes sense for your deployment scenario and your knowledge about deployment in general, Intune and the WiseMo Host.

### 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.

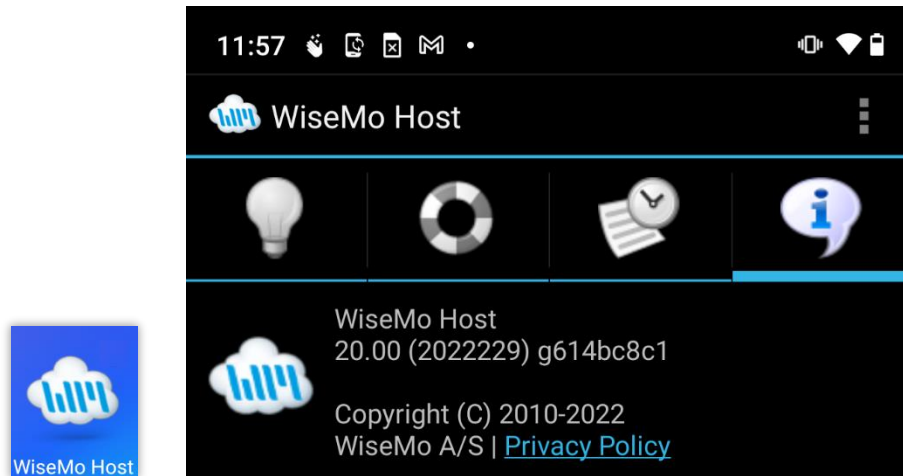
Enrollment of Android devices in Intune generally requires at least Android 6.

The document also assumes knowledge of what remote control is and in particular the role and usage of the WiseMo Host. Moreover, it is assumed that you have general knowledge about WiseMo's solution for remote control over the internet, myCloud.

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 simulate 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 fall into the following categories:

- Using an API provided by manufacturer
- Using a manufacturer specific Add-on component that is developed and provided by WiseMo but signed by the device manufacturer
- Using a Universal Add-on that's generic and uses Android's Accessibility Service.
- Using Android built-in method for capturing the screen
- A combination of the above
- Rooted devices

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

### 2.1 Using an API provided by the 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 the Knox framework which is installed on most devices, see [Knox supported devices](#).

Please refer to paragraph 7.0 *Deployment of the WiseMo Host to a Samsung device* to deploy the Host to a Samsung device using the Knox method.

## 2.2 Using a manufacturer specific Add-on component

Capturing the screen and simulating input are restricted operations that a normal app isn't capable of doing. The Add-on component comes in two fundamentally different versions, a manufacture specific add-on and a generic add-on. This paragraph is about the manufacture specific add-on while the next is about the generic add-on generally referred to as the Universal Add-on.

The manufacturer specific add-on component 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 manufactures 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 5.0 *Deployment to a device using an Add-on component* to deploy the Host to a device using the add-on component method.

## 2.3 Using the Universal Add-on component

Viewing and controlling the remote screen is generally supported from Android 8. Screen capture is supported in a combination of Android's built-in screen capture and the WiseMo Universal Add-on.

The Universal Add-on uses Android's accessibility service to simulate input. Touch input is fully supported while injection of keyboard input is partly supported (depending on app and text input control) from the Guest. The Android onscreen keyboard can be opened and used remotely.

The universal Add-on is available via Google Play but can be sideloaded by downloaded from WiseMo. Sideloaded an Accessibility app is not supported by Android from version 13.

Please refer to paragraph 6.0 *Deployment to a device using the Universal Add-on component* to deploy the Host to a device using the Universal add-on component method.

## 2.4 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 8.0 *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.

## 2.5 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 9.0 *Deploy the WiseMo Host to a Zebra device* to deploy the Host to a Zebra device.

If using Android 8 or newer, please try to use the Universal Add-on method before settling with only screen capture.

Please refer to paragraph 6.0 *Deployment to a device using the Universal Add-on component* to deploy the Host to a device using the Universal add-on component method.

## 2.6 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>&lt;uses-permission android:name = "android.permission.READ_FRAME_BUFFER" /&gt;</code>	To be able to capture the screen
<code>&lt;uses-permission android:name = "android.permission.INJECT_EVENTS" /&gt;</code>	To be able to simulate input events
<code>&lt;uses-permission android:name = "android.permission.SHUTDOWN"&gt;</code>	To be able to shut down the device. Can be omitted on request.
<code>&lt;uses-permission android:name = "android.permission.REBOOT"&gt;</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 Management tools in general

The acronyms in this area are often used at random but an overall term 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.

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.

### 3.1 Android Enterprise

When it comes to managing Android devices, all EMM solutions are based on Android Enterprise. Android Enterprise is built in to Android (from Android version 5) and offers functionality, APIs and other tools for developers to integrate management support for Android into their enterprise mobility management (EMM) solutions. So, when an EMM solution manages an Android device that is enrolled in Android Enterprise, the EMM solution uses APIs on Google servers and does not interact directly with the device. It is the Google servers that subsequently send out management instructions to the Android device. An exception to this is the EMM Device Policy Controller app (DPC app) that is installed on the device when it is enrolled in Android Enterprise. The EMM solution can interact directly with the DPC app for special purposes.

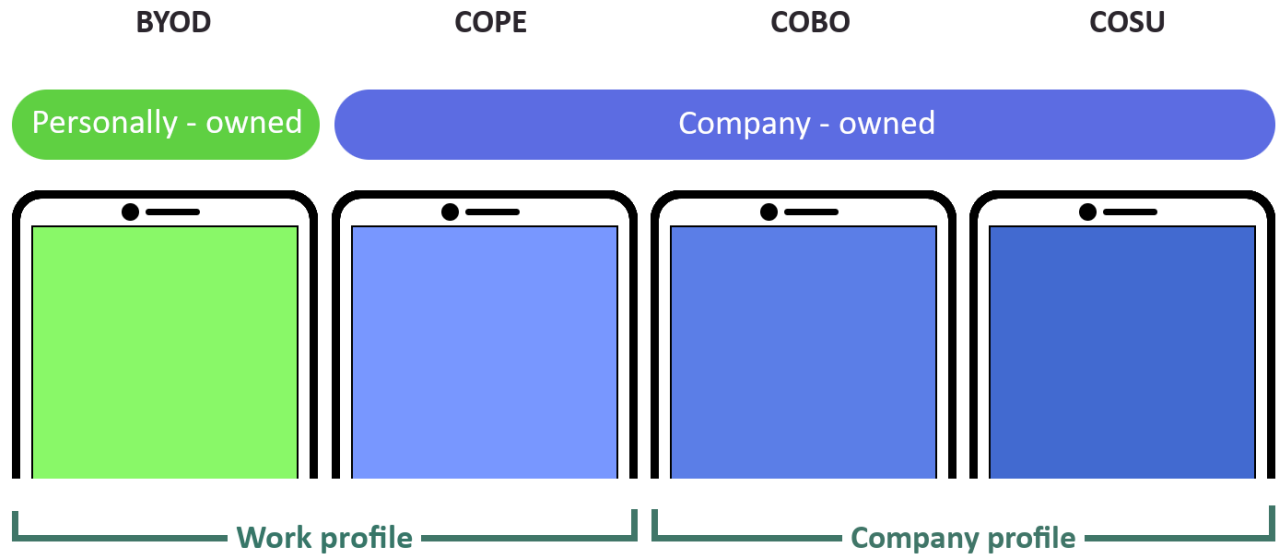
An installed app can be unaware of the management environment or it can integrate more or less into it like the WiseMo Host app does. One of the most notable features seen from the app perspective is whether it can be managed in terms of configuration (Android Managed Configuration), whether it is able to interact with both Personal and Work profile (see below) and whether it offers APIs for various special management features.

Managed Google Play is an enterprise version Google Play. Unlike the public version of Google Play, users can only install apps from Managed Google Play that their organization has approved for them via the EMM solution.

The WiseMo Android Host fully supports Android Enterprise. It supports Android Managed configuration for the most important Host settings, it allows both installation in the Personal and Work profile and it offers an API (via Intents) that can be used by the EMM Agent to Start, Restart and set the configuration directly. The WiseMo Host is available in Managed Google Play.

### 3.2 Android Enrollment

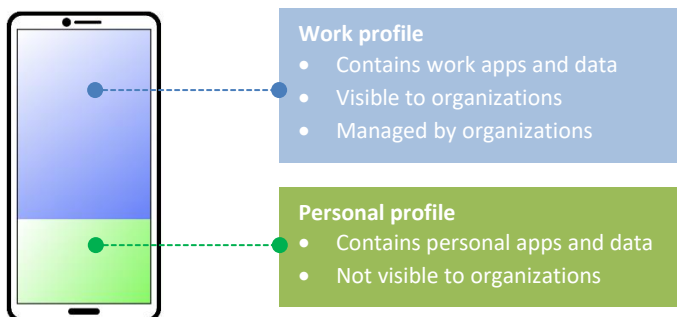
An Android device can be enrolled in different Android Enterprise modes that depending on mode have a Personal profile and a Work profile:



The most common Android Enterprise modes are described below.

#### Bring your own device (BYOD)

A personally owned device that an employee also uses for work. Work profiles, which separate work apps from personal apps, are the recommended deployment method for BYOD devices.



#### Corporate-owned, personally enabled (COPE)

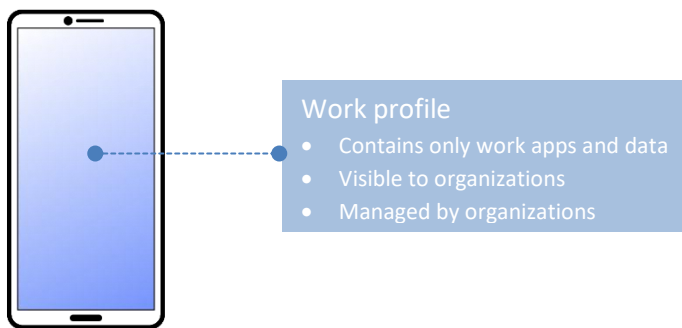
A fully managed device that is also provisioned with a work profile. Intended for company-owned devices that are used for both work and personal purposes.





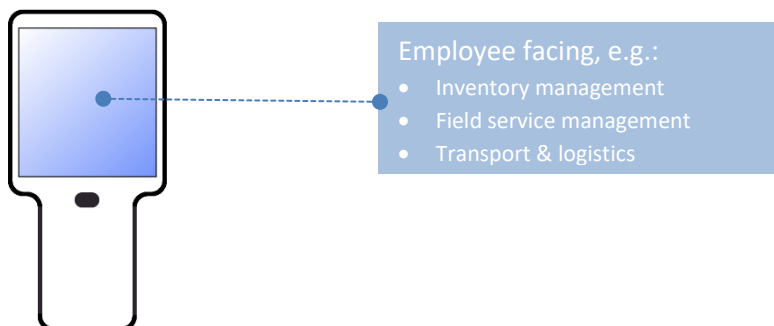
### Corporately Owned, Business Only (COBO)

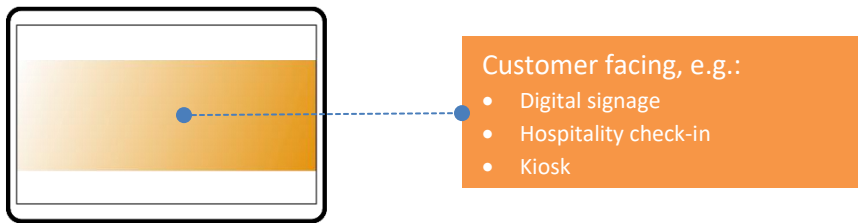
A device owned and fully managed by an employee's organization. Company-owned devices are set up exclusively for work use with a work profile.



### Corporate-owned, single use (COSU)

Full management for dedicated devices. A subset of company-owned devices that are locked down to a limited set of apps to serve a dedicated purpose, such as a check-in kiosk or digital signage.





## 4.0 Intune in general

Some general principles about Microsoft Intune are common to all deployment and are 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 of 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 in your 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 tenant<sup>1</sup>, 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.

### 4.1 Link managed Google Play to Intune

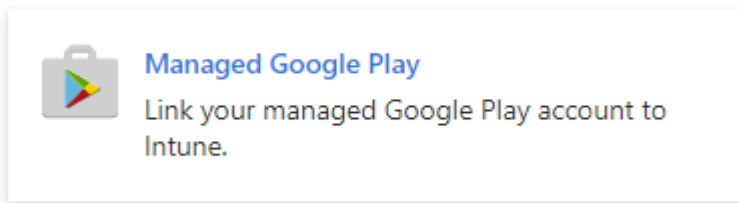
You must link Intune to your company's managed Google Play account to manage Android enterprise devices. Follow the steps below to enable Android enterprise enrollment:

1. From the navigation pane in Intune, select **Devices > Android > Android enrollment**  
The latest version of Intune might have the Android selection in the right pane as a big button.

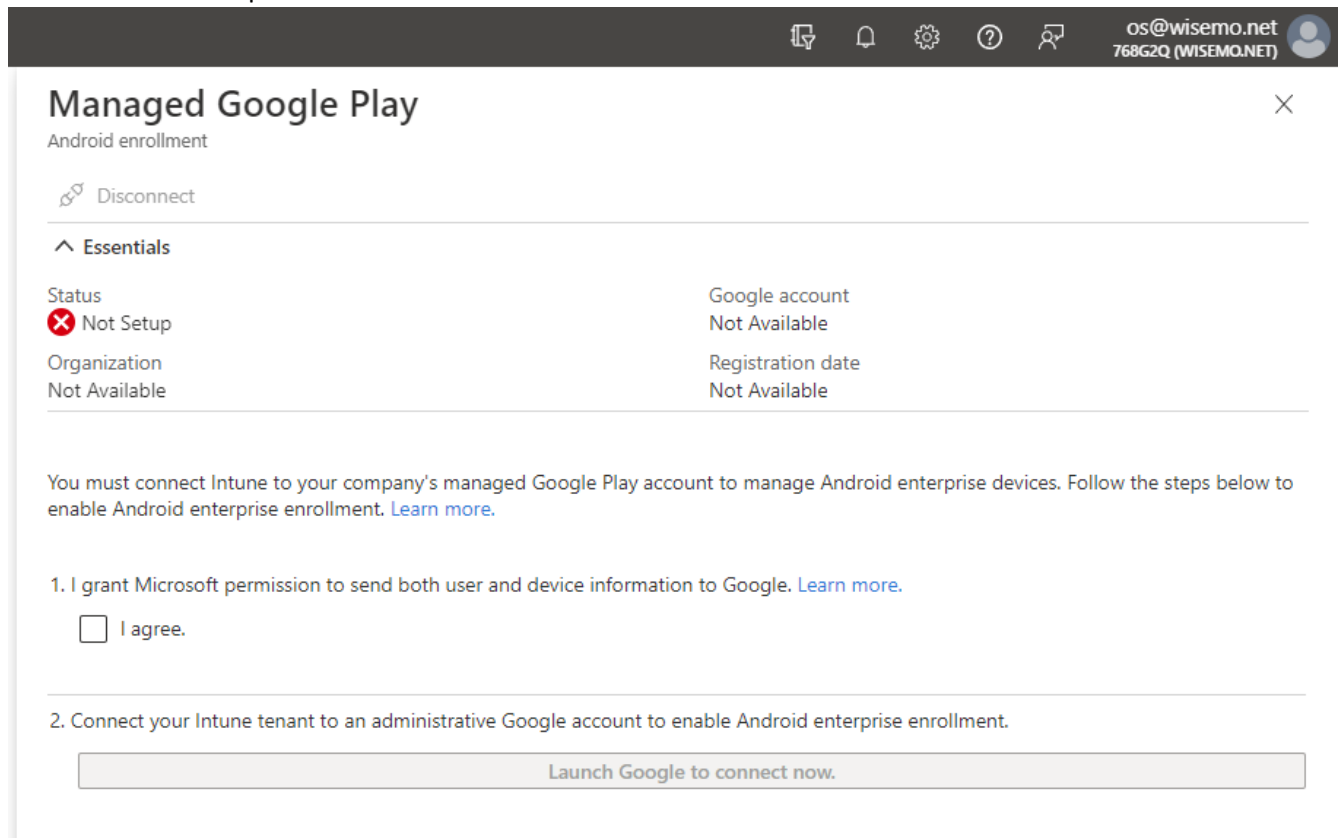
---

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

- Click **Link your managed Google Play account to Intune**



- If Intune isn't already linked to your managed Google Play account, you will see the follow screen with the status "Not setup":



- Check the **I agree** box and click **Launch Google to connect now**.
- Sign in with the managed Google Play account you want to use.
- If you get an error saying "This enterprise is already enrolled with another EMM" it's probably because the Google account is used with another EMM product. Do the following to delete this registration:
  - Sign in to the Google Account at <https://play.google.com/work/adminsettings>
  - Click on "Admin Settings" on the left (if not already selected)
  - Click on the three dots next to "Organization Information"
  - Select "Delete Organization"

*This delete action cannot be undone, so if you are not sure whether you still need this registration, please use a different managed Google Play account.*

7. When you are signed in with your Google account, Click the **Get Started** button and follow the instructions.
8. Go back to Intune and now you should see that you are successfully linked up

#### ^ Essentials

Status

✓ Setup

Organization

WiseMo

Google account

GooglePlay@WiseMo.com

Registration date

9/6/2022, 2:22:36 PM

## 4.2 Enroll a device in Intune

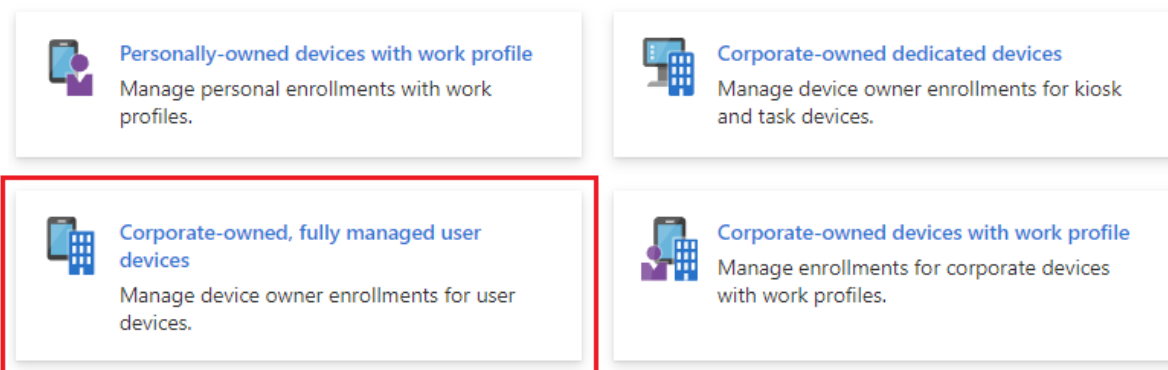
There are several methods to enroll a device into Intune – see paragraph 3.2 *Android Enrollment*. Each method depends on the device's ownership (personal or corporate), device operating system, and management requirements.

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


It is outside the scope of this document to describe how to enroll in the various enrollment methods available, so only enrollment with a token of “Corporate-owned, fully managed user devices” (what Google calls Corporately Owned, Business Only (COBO)) 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, follow the steps in paragraph 4.1 *Link managed Google Play to Intune*.
3. Then select **Corporate-owned, fully managed user devices** under **Enrollment profiles**.



4. A pane will open to the right. Click **Create profile**.
5. Specify a name and optionally a description and click **Next**.
6. Then Click **Create**.
7. Now click the profile you just created in the list and the Select **Token**:



## Android enrollment profile | Token


Android enrollment

«
⊘ Revoke token
↓ Export

ⓘ Overview

**Manage**

☰ Properties

 **Token**


### Android enrollment profile

Use this token or QR code to enroll devices. [Learn more.](#)

Token creation date  
10/17/23, 12:21 PM

Token  
XV[REDACTED]QLX

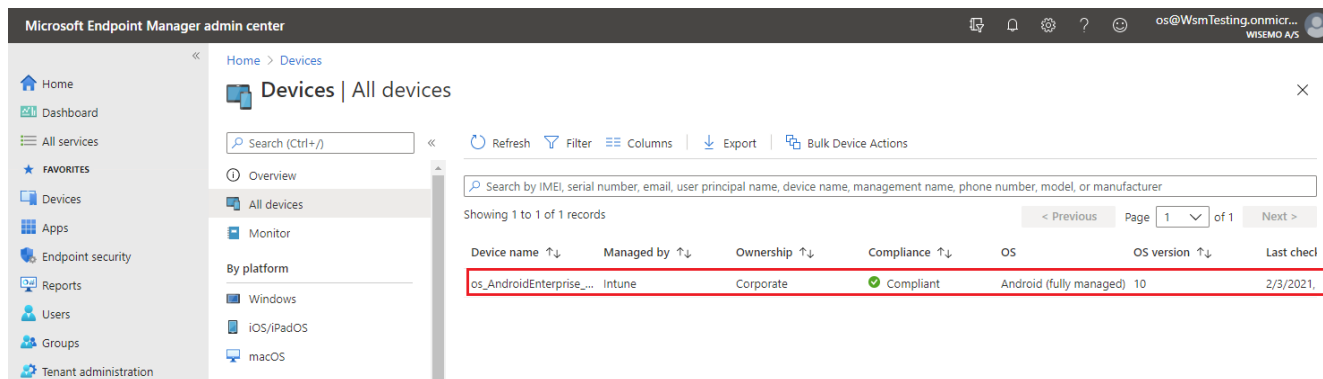
Token as QR code



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 Wi-Fi, and then choose **Next**.
4. If asked, don't copy configuration from another device. 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 the enrollment. When asked to log in to Microsoft, login with an Intune user account.
9. Install the required apps and follow the instructions.

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 name in **Management name**:

## os\_AndroidEnterprise\_2/3/2021\_2:51 PM | Properties

The screenshot shows the 'Properties' page for the device. The left sidebar has 'Properties' selected. The main area contains several fields:

- Device name:** os\_AndroidEnterprise\_2/3/2021\_2:51 PM
- Management name \*:** Android Device 1 (highlighted with a red box)
- Device category:** Unassigned
- Device ownership:** Corporate

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](#).

### 4.3 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 **+ 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 Members selected** to add members to this group. A pane opens to the right. Click the **Devices** tab and 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:

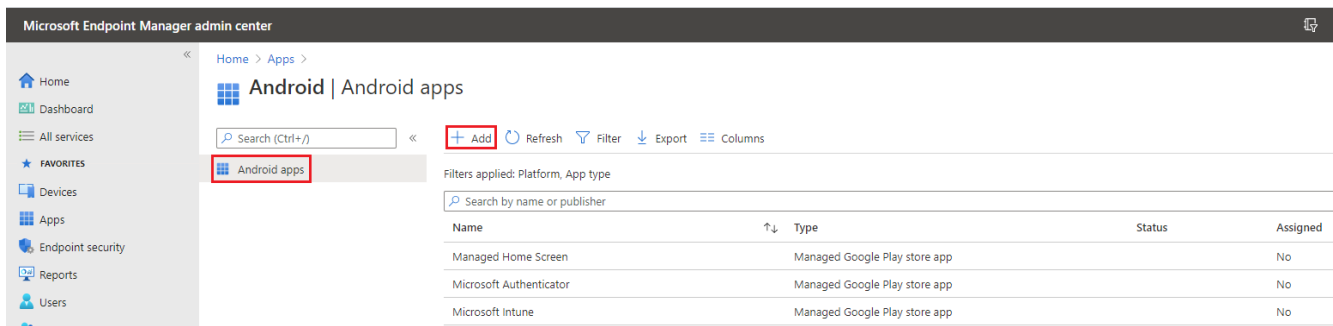


Please note that only the first 50 items are shown so you might have to search to limit the number of items shown. When you have found the device, click **Select** and then **Create** to create the group that we will use in the next paragraph.

### 4.4 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.

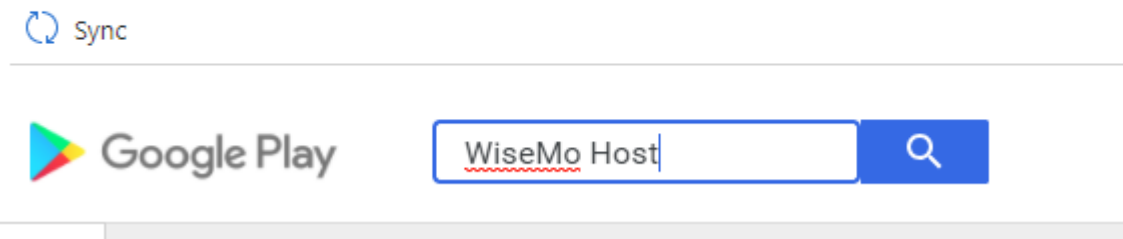
From the navigation pane, select **Apps > Android** to display an overview of Android apps and their status. Even in a fresh Intune installation, you will see some Microsoft apps:



To add an app, do the following:

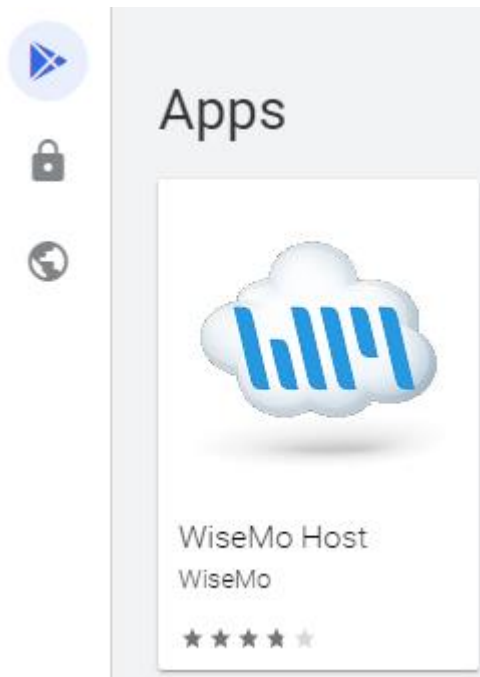
1. Click **Add** from the menu.
2. In the **Select app type** pane, under the available **Store app** types, select **Managed Google Play app**.
3. Click **Select**.
4. Intune will open Google Play. In the search field, write **WiseMo Host** and click the search symbol:

## Managed Google Play



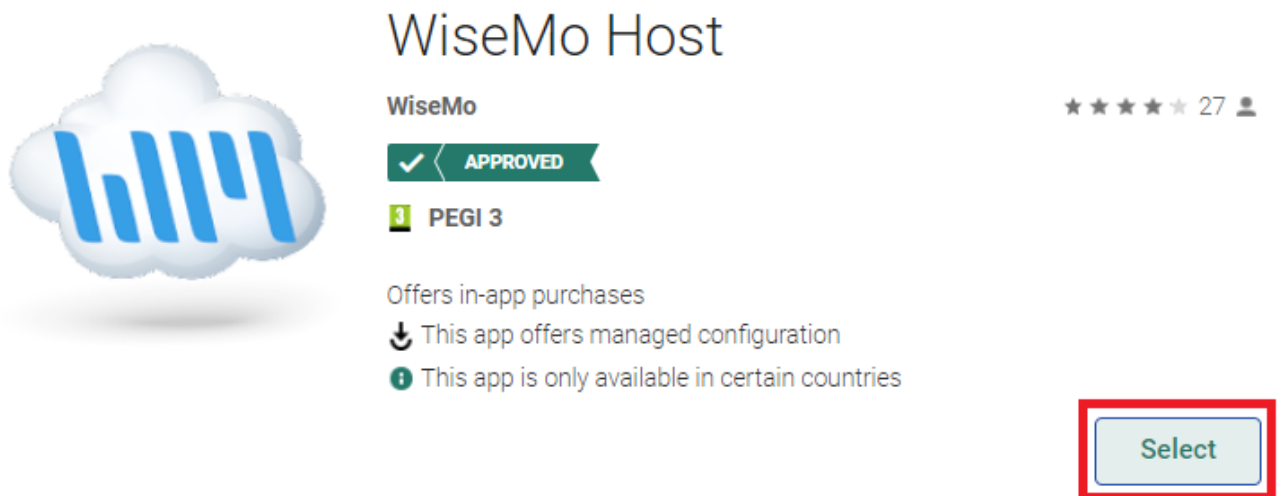
5. Select the Host app by clicking on it:






You will now go through a number of approval steps.

6. In the window that appeared click the **Select** button so it highlights like indicated:



While it is highlighted, click the Sync button in the upper left corner  Sync

You will return to the **Android Apps** page.

7. Click the Refresh button in the top several times until you see the “WiseMo Host” in the list.

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

Search (Ctrl+/)

«

+

Add

Refresh

Filter

Export

Columns


Android apps

Filters applied: Platform, App type

Search by name or publisher

Name	↑↓	Type	Status	Assigned
WiseMo Host		Managed Google Play store app		No


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

 **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 ①

Group mode Group

No assignments

[+ Add group](#) ①
 [+ Add all users](#) ①
 [+ Add all devices](#) ①

### Available for enrolled devices ①

Group mode Group

No assignments

[+ Add group](#) ①
 [+ Add all users](#) ①

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

## Select groups



Azure AD groups



Android devices

Selected

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 (and it may take some time), the WiseMo Host app should appear on the Android device(s) you added to the **Android devices** group.

### 4.5 Managed configuration and app permissions

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

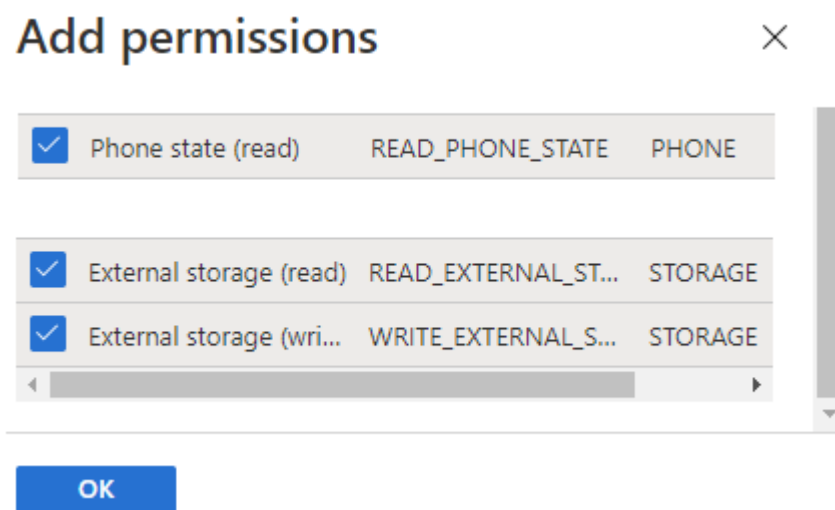
The other topic for this paragraph is predefined app permissions because it is configured in the same place as configuration settings. By predefineding app permissions, the user is alleviated from accepting permissions prompts from the app on the device. Unfortunately, it is not all permissions that can be predefined.

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

1. Choose the **Apps > App configuration policies** and select **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.

+Add				
Permission ↑↓	Permission state ↑↓	Permission name ↑↓	Permission group ↑↓	
Phone state (read)	Auto grant	READ_PHONE_STATE	PHONE	...
External storage (read)	Prompt	READ_EXTERNAL_STORAGE	STORAGE	...
External storage (write)	Auto grant	WRITE_EXTERNAL_STORAGE	STORAGE	...
Configuration Settings	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 too<sup>2</sup> 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.

<sup>2</sup> The Intune configuration design doesn't support configuration types such as the Bundle.

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 consist of a key and a 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 Shared 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 is which. Therefore it's a good idea to use the Intune variables, see tokens that can be used here: [Supported variables for configuration values](#). To use Intune's **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](#))

[Make password non-displayable](#)

[Change Password](#)

Click Show next to **Password** to see the password. Fill out 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 licensed. A Host can either be 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": {
```

```

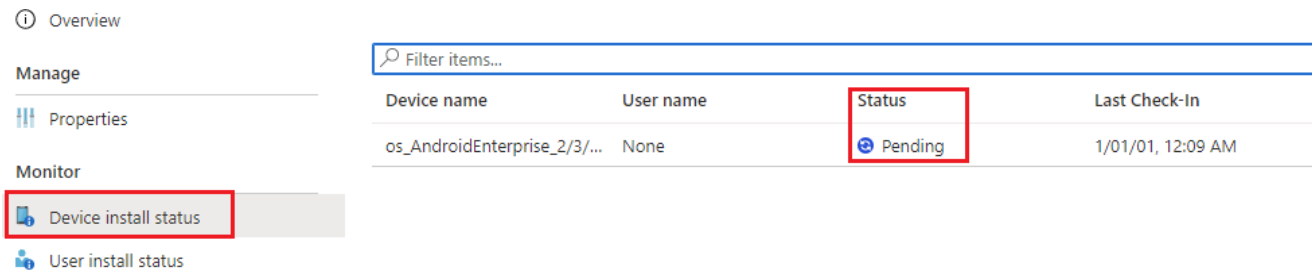
"managedProperty": [
  {
    "key": "hostLicenseMode",
    "valueString": "license_mode_mycloud"
  },
  {
    "key": "hostLicenseKey",
    "valueString": ""
  }
]
}

```

15. When you are done, click **Next** to display the **Assignments** page.
16. On the **Assignment** page click **Add groups** and select the **Android devices** group 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).

## 4.6 Completing device installation

In paragraph 2.0 *Overview of WiseMo Host* we discussed the various methods for capturing the device screen and simulating input in order to provide full remote control of a device. To finalize the setup for your devices you must complete one of the following paragraphs – or perhaps multiple paragraphs for devices requiring different methods.

## 4.7 A note about device restrictions and blocking screen capture

It is possible to create **Device restriction** policies in **Devices > Configuration policies** that will block screen capture, e.g.:

Profile name ↑	Platform	Profile type
<a href="#">Prevent remote control</a>	Android Enterprise	Device restrictions

Please refer to Microsoft's documentation about [Device restrictions for Android Enterprise](#) and about [Device Restrictions for Samsung Knox](#):

- **Screen capture: Block** prevents screenshots or screen captures on the device in the work profile. It also prevents the content from being shown on display devices that don't have a secure video output. When set to **Not configured** (default), Intune doesn't change or update this setting. By default, the OS might allow getting screenshots.
- **Screen capture (Samsung Knox only): Block** prevents screenshots. When set to **Not configured** (default), Intune doesn't change or update this setting. By default, the OS might let users capture the screen contents as an image.

As you are reading this document you will probably want full remote control to work and it would defeat the purpose to have policies preventing screen capture - so be sure that screen capture isn't blocked for devices you want to remote control.

## 5.0 Deployment to a device using an Add-on component

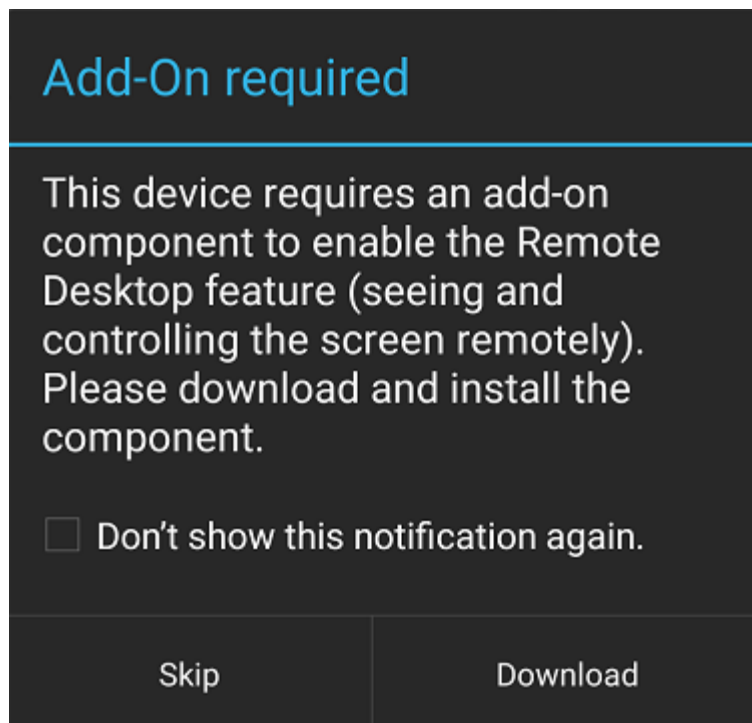
The add-on technique uses an intermediate component (the add-on) to bridge access from the WiseMo Host App to the Android operating system. The Add-on is an extra app that needs to be installed on the device and hence the Add-on app is deployed in the same way as the Host app is deployed.

### 5.1 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.

## 5.2 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 4.4 *Add apps to Intune*. There are no permissions or managed configuration that should be set for the Add-on app.

In paragraph 4.3 *Intune groups* we created the group **Android Devices**. If you have only one type of Android devices you can assign 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.

## 6.0 Deployment to a device using the Universal Add-on component

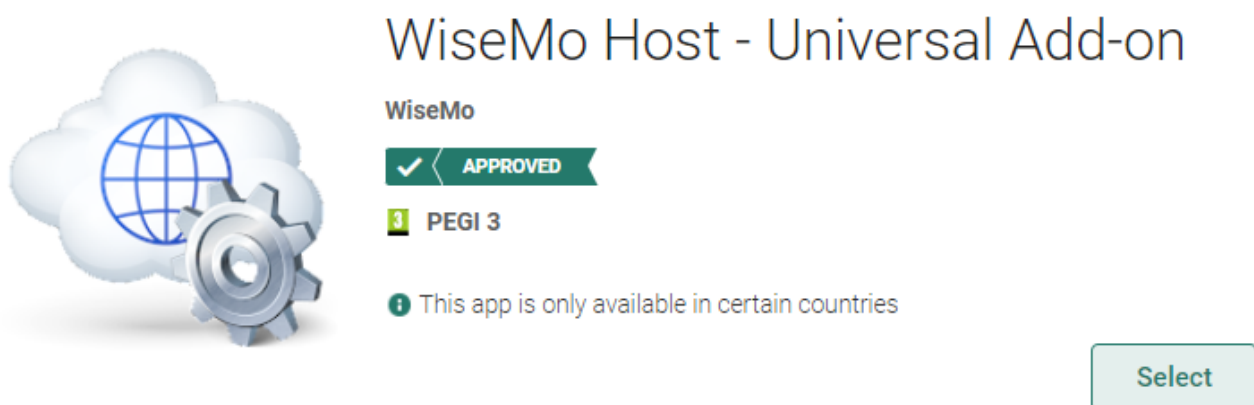
The Universal add-on is an intermediate app that needs to be installed to simulate input and to some extent, capture the screen. Screen capture is supported in a combination of Android's built-in screen capture and the Universal Add-on's screen capture.

The Universal Add-on uses Android's accessibility service to simulate input. Touch input is fully supported while injection of keyboard input is partly supported (depending on app and text input control). The Android onscreen keyboard can be opened and used remotely.

The Universal 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. The universal Add-on is available via Google Play but can be sideloaded by downloading it from WiseMo. Sideloaded an Accessibility app is not supported by Android from version 13.

### 6.1 Deploying the Universal Add-on

Add the Universal Add-on app in the same way as you did in paragraph 4.4 *Add apps to Intune*. When Managed Google Play opens, search for "WiseMo Universal Add-on" and select it. Press **Sync** and locate the Universal Add-on app in the list.



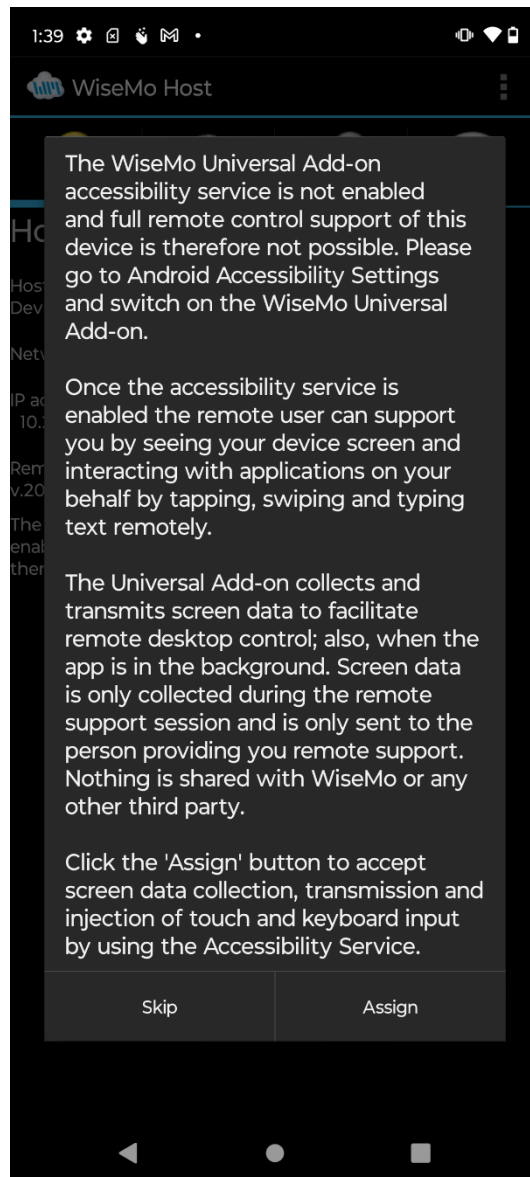
There are no permissions or managed configuration that should be set for the Add-on app.

In paragraph 4.3 *Intune groups* we created the group **Android Devices**. If you have only one type of Android devices you can assign the Universal 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 Universal Add-on**. Assign the Universal Add-on app to this group and add the member devices that should use the Universal Add-on to this group. After some time, the Universal Add-on app will be deployed to the devices.

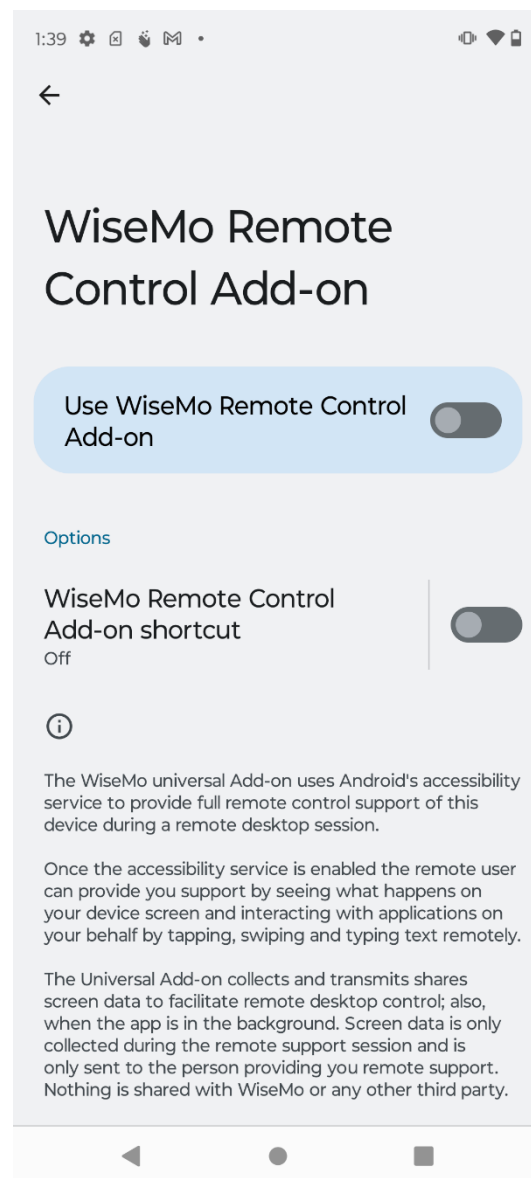
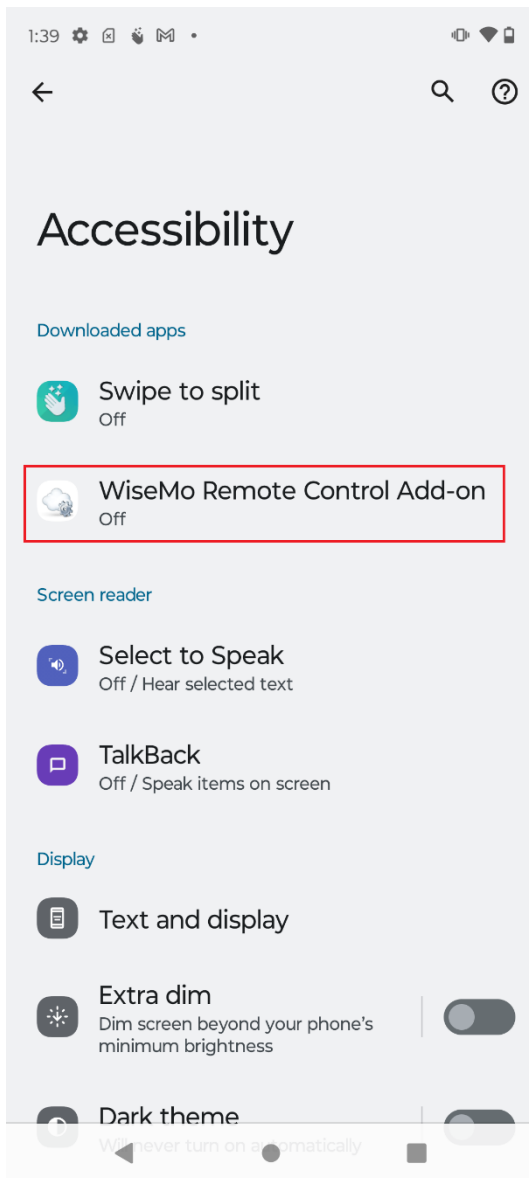
## 6.2 Enabling the Universal Add-on

Unlike the manufacturer specific Add-on, it is necessary with a manual step on the device to enable the Universal Add-on in Accessibility Services.

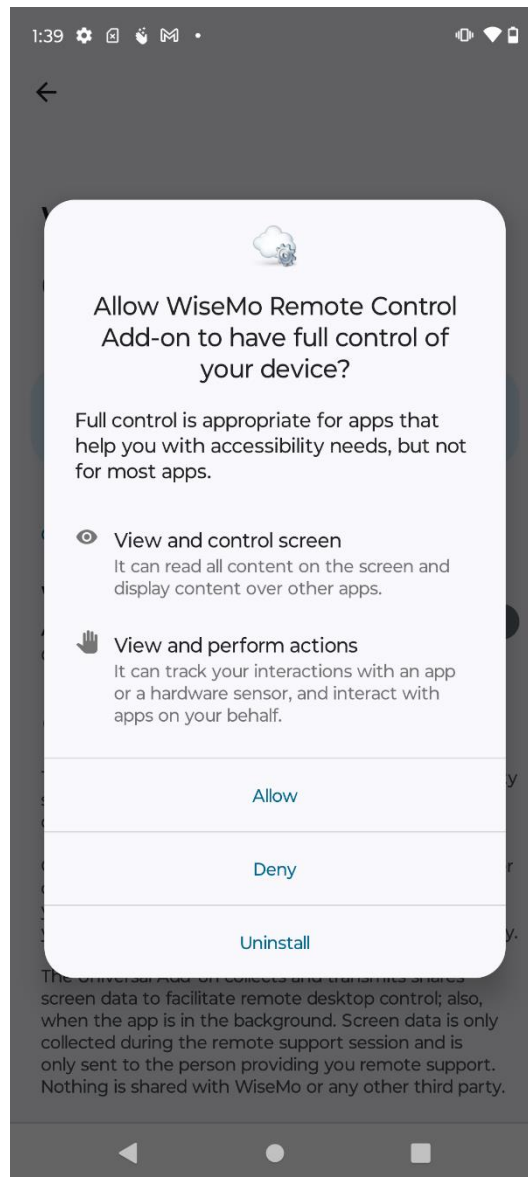
When the WiseMo Host is launched for the first time, it will show the following prompt:



Click **Assign** to enable the Universal Add-on in Accessibility settings, tap the **WiseMo Remote Control Add-on**, tap the switch on:



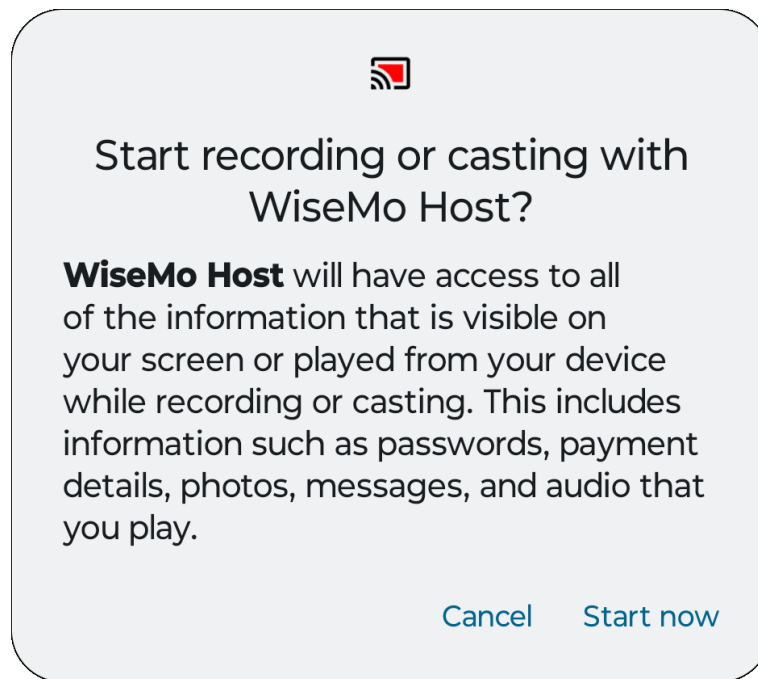
and finally tap **Allow**:



Click the back arrow several times to return to the Host. The Host is now ready to be remote controlled with the Universal Add-on.

### 6.3 Remote desktop controlling with the Universal Add-on

When the Host app is started manually, Android will show a prompt to allow the Host app to record the device screen.



When this prompt is shown, you should click **Start now**. If you click **Cancel** or if the Host app is start automatically (for example when the device starts up or via its API) this prompt will be postponed till after a remote desktop control session has been started. It is hence possible with the Accessibility services functionality to see this screen with its slower capture functionality.

To allow the Host to show this screen even when the Host is in the background, the Host should be given the permission **Display over other apps**. The Host will prompt for this permission on first start, otherwise see Host menu ≡ > **Settings > App permissions**.

## 7.0 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 EMM features specific to Samsung devices.

### 7.1 Add-on method

For Android 9<sup>3</sup> and older there's also an Add-on available and the method in *5.0 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

---

<sup>3</sup> With some limitation the Add-on might also work on newer Android versions. Make sure you use the latest WiseMo Host app.

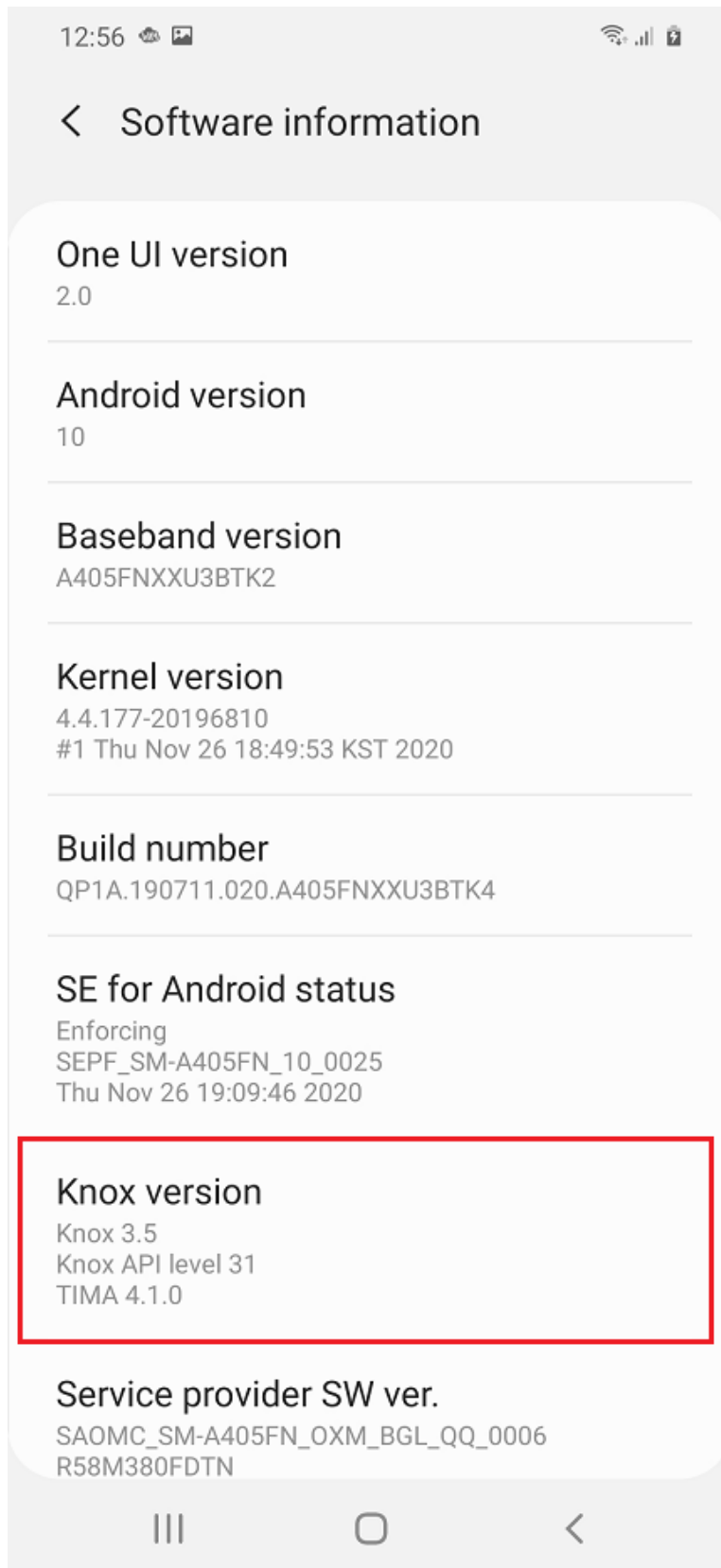
user interaction on the device on first run or the Add-on can be used on Samsung devices where the Knox API isn't available (see <https://www.samsungknox.com/en/knox-platform/supported-devices>).

## 7.2 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 link <https://www.samsungknox.com/en/knox-platform/supported-devices> specifies which devices supports 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:





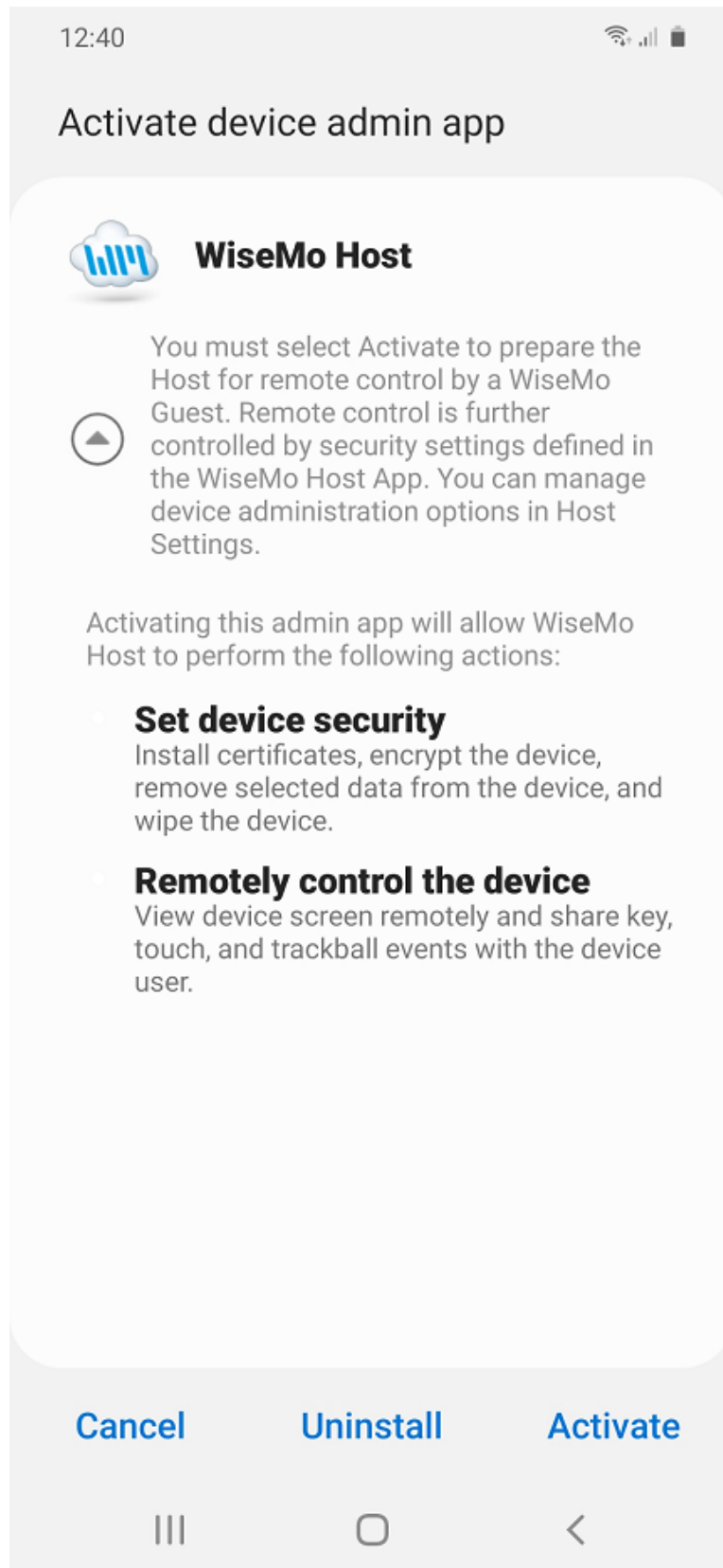
A Samsung device can be enrolled in Intune as described in paragraph 4.2 *Enroll a device in Intune*. A Samsung device can also be enrolled using Samsung Knox Mobile Enrollment (KME). Using Intune with Samsung KME, you can enroll large numbers of company-owned Android devices when end users turn on their devices for the first time and connect to a Wi-Fi 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:



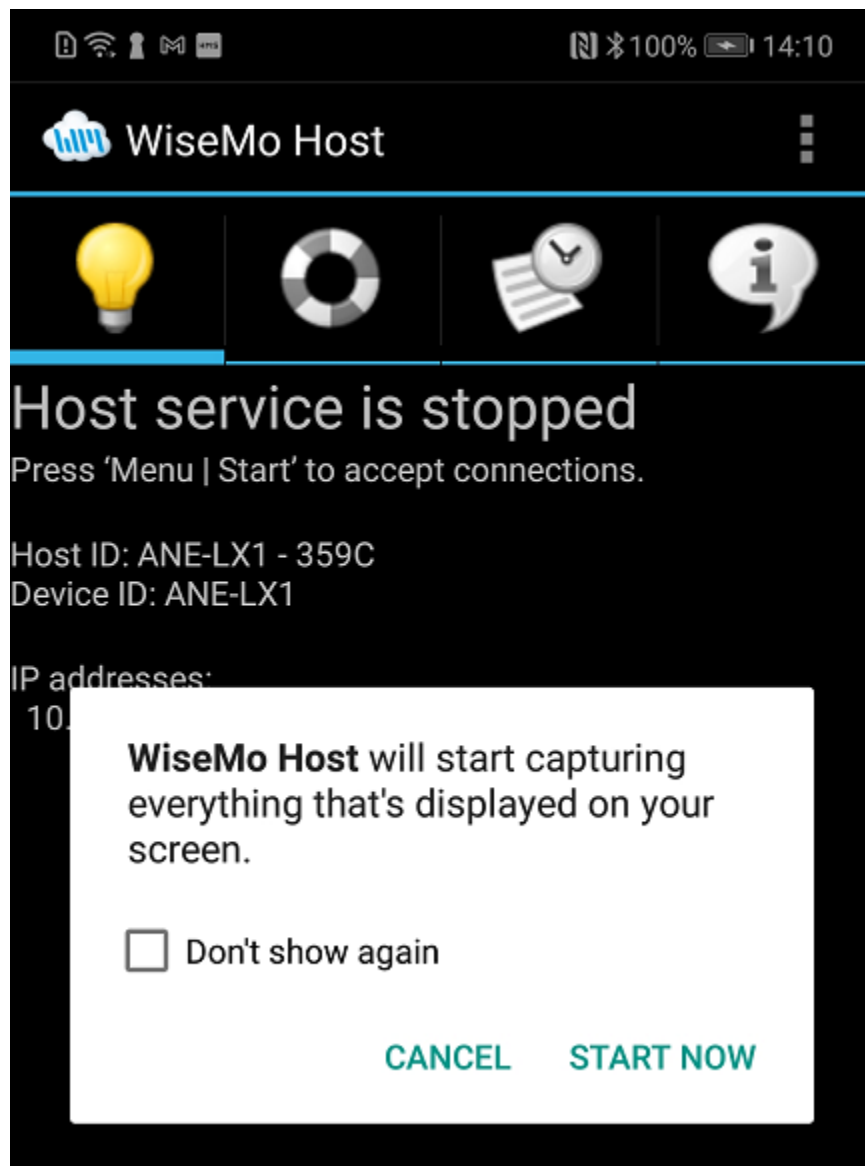
Click Activate and the Host is ready to be remote controlled with the Managed configuration that was assigned in paragraph 4.5 *Managed configuration and app permissions*.

## 8.0 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 a last resort.

The WiseMo Host is deployed and configured like it is described in chapter 4.0 *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 pre-assigned in Intune or any other EMM 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.

## 9.0 Deploy the WiseMo Host to a Zebra device

The WiseMo Android Host (version 18 and above) fully supports remote control of Zebra Android mobile and handheld devices if the device has MX 8.3 (Mobility Extensions) or greater is installed. If a device uses an older MX version, please check 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](#). This document also describes how to find the MX version.

### 9.1 Enroll a Zebra in Intune

Zebra devices can be enrolled in Intune as:

- **Android Enterprise** devices, see paragraph 9.4 *Create a configuration profile in Intune – for Android Enterprise devices*
- **Android device administrator** devices, see paragraph 9.5 *Create a profile in Intune – for Android device administrator devices*

### 9.2 Method and configuration files depend on Android version

Configuration of a Zebra device to allow remote desktop control depends on the Android and MX versions.

The following table defines 3 target groups and list the components and files you need.

	Android 8 - 9	Android 10-11	Android 11+
MX version required	MX version 8.3+	MX version 10+	MX version 11.5+
Zebra OEM Config app	Legacy Zebra OEMConfig	Legacy Zebra OEMConfig	Zebra OEMConfig Powered by MX
Suggested Android Zebra Device group name	Android Zebra Devices (8-9)	Android Zebra Devices (10-11)	Android Zebra Devices (11+)
Suggested Configuration profile name	Zebra Android Enterprise configuration for WiseMo Host (8-9)	Zebra Android Enterprise configuration for WiseMo Host (10)	Zebra Android Enterprise configuration for WiseMo Host (11+)
Configuration method	JSON	JSON	JSON
JSON configuration	<a href="#">Download file</a>	<a href="#">Download file</a>	<a href="#">Download file</a>

*Zebra configuration table*

Please note that there are two different Zebra OEMConfig apps. A legacy app for Android 8-11 and new app for Android 11 and newer. For Android 11 both apps can be used with the correct JSON file – so download the applicable JSON file – there are 3 different JSON files.

The table will be referenced in the following.

### 9.3 Create groups for your Zebra device and assign devices

In the following you'll need one or more groups to assign the Zebra configuration policies to. It is necessary to create a group for each Android target group in the *Zebra configuration table* above.

Follow the procedure described in 4.3 *Intune groups* to create a new group:

1. Go to **groups > All groups** and select the **New Group**.
2. Give it a name; for example, the suggested name in the *Zebra configuration table* above and enter a description.
3. Click **No members selected**.
4. Click the **Devices** filter button and check the Zebra devices that should belong to this group.
5. When all devices have been checked, click **Select**.
6. Finalize, and create the group.

Repeat this for each target group in the *Zebra configuration table* above.

### 9.4 Create a configuration profile in Intune – for Android Enterprise devices

This paragraph shows you how to configure Zebra Mobility Extensions (MX) on Zebra devices enrolled as Android Enterprise devices.

It requires quite a lot of configuring in Intune but hang on.

First, we need to add the Zebra OEMConfig app to Intune like we did for the WiseMo Host app in paragraph 4.4 *Add apps to Intune*.

From the navigation pane, select **Apps > Android** and then do the following<sup>4</sup>:

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 "Zebra OEMConfig" and find the Zebra OEM Config app as specified in the *Zebra configuration table* above:

---


<sup>4</sup> It is also possible to follow the instruction [here](#). The package name of the app that should be added is com.zebra.oemconfig.common




## Zebra OEMConfig Powered by MX

Zebra Technologies

 PEGI 3

 This app offers managed configuration

 This app is only available in certain countries

Select

5. Click **Select** and then click the **Sync** button in the upper left corner.
6. You will return to the list of Android apps. Click refresh in the toolbar until you see the app.
7. Select the app and click **Properties**.
8. Click **Edit** next to Assignments
9. In the Required section click **+Add group**, find and select the Android Zebra Device group as suggested in the *Zebra configuration table* above.
10. Click the **Select** button.
11. **Click review and save** and finally **Save**.
12. After some time, the app will have been deployed to the Zebra devices, please refer to **Device Install Status**.

Repeat this for both Zebra OEM Config apps if necessary.

At this point we have created one or more groups for the Zebra devices, assigned our Zebra devices to these groups and we have assigned the Zebra OEM Config apps to the groups. We are now ready to create a specific Zebra configuration profile that will allow the WiseMo Host app to capture the screen and emulate keyboard and touch events.

In Intune, create a device configuration profile:

1. Go to **Devices > Configuration profiles** and select **Create profile**.
2. Enter the following properties:
  - **Platform:** Select **Android Enterprise**.
  - **Profile:** Select **OEM config**.



## Create a profile



Platform

Android Enterprise



Profile type

OEMConfig



And click **Create**.

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

- **Name:** Enter a descriptive name for the new profile, for example as suggested in the *Zebra configuration table* above.
- **Description:** Enter a description for the profile. This setting is optional, but recommended.

### Create profile ...

OEMConfig

1 Basics

2 Configuration settings

3 Scope tags

4 Assignments

5 Review + create

Name \* ⓘ

Zebra Android Enterprise configuration for WiseMo Host (11+)



Description ⓘ

Zebra Android Enterprise configuration on Android 11+ devices for WiseMo Host

OEMConfig app \*

Select an OEMConfig app

4. Click **Select an OEMConfig app** and select the Zebra OEMConfig app as specified in the *Zebra configuration table* above.

Associated app

×

✓ ⓘ

App name	Publisher
Zebra OEMConfig powered by ...	Zebra Technologies

5. Click the **Select** button.
6. Click the **Next** button.
7. In Configuration settings select **JSON editor**:

[All services](#) > [Devices | Configuration profiles](#) >

## Create profile ...

OEMConfig

✓ Basics
2 Configuration settings
3 Scope tags
4 Assignments
5 Review + create

Configure settings with


Configuration designer

Configuration designer

JSON editor

Settings

 Locate

 Zebra OEMConfig powered by MX

**Zebra OEMConfig powered by MX**

Manage Zebra's custom MX features & settings through your EMM/UEM.

Transaction Steps


①

[Configure](#)

8. Delete everything in the edit window that appears.
9. Download the WiseMo Host JSON settings file as specified in the *Zebra configuration table* above. Open the file you downloaded in Windows Notepad. Copy all text – don't start formatting it to make it look nicer – and paste it in to the edit window.
10. There should be no errors like "Invalid OEMConfig settings" – if there are, you probably accidentally made some changes to the JSON text.
11. Click the **Next** button.
12. Configure Scope Tags if it makes sense for you and click **Next**.
13. In Assignments, click **Add groups** and select the Android Zebra Devices group we created previously for the target you are configuring.
14. Click the **Select** button and then the **Next** button.
15. Review the profile and finally, click the **Create** button.

16. Click Refresh until the new profile appears in the list.
17. Click the profile to open it and select **Device Status** in the menu to the left:

[Home](#) > [Devices | Configuration profiles](#) > [Zebra Android Enterprise configuration for WiseMo Host \(11+\)](#)



## Zebra Android Enterprise configuration for WiseMo Host (11+) | Device status

Device configuration profile

This experience will be improved in an upcoming Intune update. [Learn more about this change.](#)

Refresh Export Columns

Overview

Overview

Manage

Properties

Monitor

**Device status**

User status

Device	User Principal Name	Deployment status
Zebra ET40	os@wisemo.net	Pending

18. The Deployment status will be Pending until the configuration is deployed.
19. After some time, the status should change to Succeeded:

 Succeeded

20. If the status is Failed, open the OEM Config app on the device to inspect the result. It will give you a hint about what is wrong, e.g.



In this case, we have deployed a configuration profile that contains configuration for an unsupported MX version.

21. When the configuration profile is successfully deployed, the WiseMo Host should launch automatically. If not. Launch the WiseMo once on the device once.
22. If your device doesn't support MX version 10.0 or newer, Android will show the following screen:

**WiseMo Host** will start capturing everything that's displayed on your screen.

☐ Don't show again

**CANCEL**   **START NOW**

Check **Don't show again** and then the **START NOW** button.

23. Now the WiseMo Host should be configured and provide full remote control support.

Repeat these steps for each target group in the *Zebra configuration table* above.

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

It may be relevant to refer to this [article](#) by Microsoft.

In Intune, create a device configuration profile:

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

**Platform:** Select **Android device administrator**.

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

## Create a profile



Platform

Android device administrator



Profile type

MX profile (Zebra only)



Select **Create**.

3. 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

1 Basics

2 Configuration settings

3 Assignments

4 Review + create

Name \*

Zebra configuration for WiseMo Host



Description

Zebra configuration for WiseMo Host input simulation.



Select **Next**.

4. Download the WiseMo configuration file [here](#) for devices supporting MX version 10 or newer, otherwise download this file [here](#) for MX version 8.3-9.x.
5. In **Configuration settings** > **Choose a valid Zebra MX XML file**, and specify WiseMo configuration file you just downloaded and paste in the unchanged text.

## 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 (\*\*\*\*).

6. 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**.

7. In **Assignments**, click **Add groups** and select the group, **Android Zebra devices** you defined above. Select **Next**.
8. 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 EMM tools. Managed configuration is supported from version 18 of the Android Host. Some configuration options require a newer version of the Android Host.

The basis for Managed Configuration is a JSON file that can either be modified directly or via a built-in UI in the EMM 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 Shared 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": "macDirectConnectionDomain",
      "valueString": "[myCloud]"
    },
    {
```

```

    "key": "myCloudProfile",
    "valueBundle": {
      "managedProperty": [
        {
          "key": "myCloudUrl",
          "valueString": "http://mycloud.wisemo.com/cm"
        },
        {
          "key": "myCloudAccount",
          "valueString": ""
        },
        {
          "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
},
{
    "key": "mntPwdSection",
    "valueBundle": {
        "managedProperty": [
            {
                "key": "mntPwd",
                "valueString": ""
            },
            {
                "key": "mntPwdProtectConfiguration",
                "valueBool": true
            },
            {
                "key": "mntPwdProtectActions",
                "valueBool": true
            }
        ]
    }
},
{
    "key": "userPrompts",
    "valueBundle": {
        "managedProperty": [
            {
                "key": "showAccessAllFilesPrompt",
                "valueBool": false
            }
        ]
    }
}
}

```

```

    ]
}

```

Table of keys and values:

Key	Type	Values	Comment
authMode	valueString (choice)	shared_password username_and_password mycloud_device_access_control	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).  Supported from version 20.0: mycloud_device_access_control
sharedPassword	valueString		For authMode = shared_password  By default, no password
sharedPasswordConfirmAccess	valueBool (choice)	true false	For authMode = shared_password
macDirectConnectionDomain	valueString		For authMode = mycloud_device_access_control  myCloud domain to authenticate users connected to Host with one of the direct communication profile (TCP or UDP)  Supported from version 20.0
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 to myCloud and go to Settings > Connection and see the settings for the Default connection account.  This is not your myCloud user account password!
hostNamingMode	valueString (choice)	naming_mode_computername naming_mode_enter_or_leave_blank naming_mode_alternate	

		<code>naming_mode_imei_or_serial_number</code>	
<code>hostNameSpecific</code>	<code>valueString</code>		The name when hostNamingMode is <code>naming_mode_enter_or_leave_blank</code>
<code>hostLicenseMode</code>	<code>valueString (choice)</code>	<code>license_mode_dont_change</code> <code>license_mode_mycloud</code> <code>license_mode_key</code>	Configure license mode. For <code>license_mode_mycloud</code> a valid myCloud configuration ( <code>myCloudDomain</code> , <code>myCloudAccount</code> and <code>myCloudPassword</code> ) must be defined. For <code>license_mode_key</code> the <code>hostLicenseKey</code> must be defined.  Supported from version 18.0 (build 2021078)
<code>hostLicenseKey</code>	<code>valueString</code>		Set WiseMo license key for <code>license_mode_key</code> .  Supported from version 18.0 (build 2021078)
<code>stopHostIfNoActivity</code>	<code>valueInteger</code>		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)
<code>mntPwd</code>	<code>valueString</code>		Maintenance password. Empty value indicates that no password is assigned and user can edit settings and start/stop service  Supported from version 20.0
<code>mntPwdProtectConfiguration</code>	<code>valueBool</code>	<code>true</code>  <code>false</code>	When this value is "false" user can open Settings UI without prompt for password even if maintenance password is assigned.  Supported from version 20.0
<code>mntPwdProtectAction</code>	<code>valueBool</code>	<code>true</code>  <code>false</code>	When this value is "true" user must enter the maintenance password to start/stop the Host service and for other actions.  Supported from version 20.0 (build 2025040)
<code>showAccessAllFilesPrompt</code>	<code>valueBool</code>	<code>true</code>  <code>false</code>	If set to true, the Host will, if necessary, show a prompt asking for 'All files access'.  When using Managed Configuration and for EMM tools like Intune and Clyd, this prompt will by default be

			<p>suppressed to allow a Host deployment that won't require manual user actions.</p> <p>Supported from version 20.0 (build 2023003).</p>
automaticWakeupMode	valueBool	<p>true</p> <p>false</p>	<p>false: (default) automatic wakeup is disabled.</p> <p>true: automatic wakeup mode: accept incoming connection without the Host service being started using myCloud profile with Push Notifications enabled. The Host will also accept incoming connections in Paused state.</p> <p>Supported from version 20.0 (build 2024348).</p>

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

The latest Managed Configuration template can be downloaded [here](#)