Android Host advanced installation and configuration

1.0 Overview

This document describes features for advanced configuration and installation of the Android Host. Specifically the document describes how to add files to the APK and advanced configuration options. Normal user configuration is described in the Android Host user guide.

1.1 Prerequisites

The document assumes knowledge of Android APKs and the WiseMo Host in general.

The ability to install additional files with the Host was introduced in version 10.5 build 2015036.

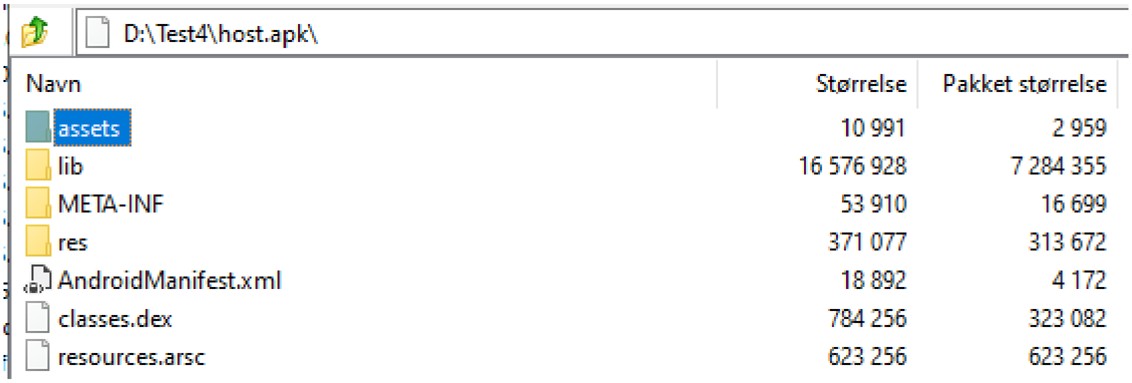
The option features are introduced over time and it is recommended to use the latest Host.

The WiseMo Host Manager enables configuration of an Android Host connected to a Windows PC. The WiseMo Host Manager can be downloaded here. Download and run the MSI installation. As for using the Manager, please refer to Android Host guide (pdf)**.**

2.0 Adding files to the Host APK

An Android Host APK can be preconfigured with license, configuration and options files.

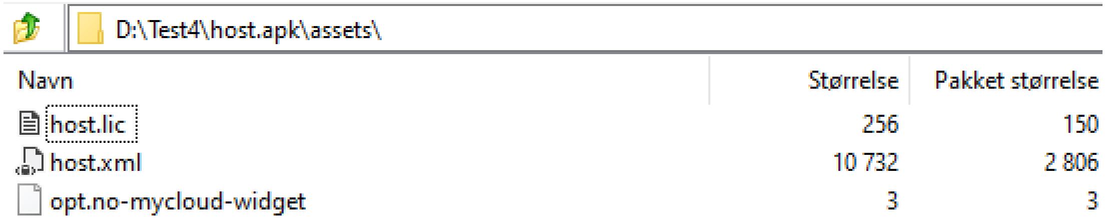
Open the APK with for example 7-Zip and create a folder called "assets":



D:\Test4\host.apk\

Navn	Størrelse	Pakket størrelse
assets	10 991	2 959
lib	16 576 928	7 284 355
META-INF	53 910	16 699
res	371 077	313 672
AndroidManifest.xml	18 892	4 172
classes.dex	784 256	323 082
resources.arsc	623 256	623 256

Copy the files that should be installed to the assets folder:



D:\Test4\host.apk\assets\

Navn	Størrelse	Pakket størrelse
host.lic	256	150
host.xml	10 732	2 806
opt.no-mycloud-widget	3	3

The Host configuration file must be named “host.xml” while the Host license file must be named “host.lic”.

The files in the “assets” folder are installed to the /sdcard/Android/data/com.wisemo.host.v10/files/WsmHost[i] folder on first run. The assets files will overwrite existing files, but only on first application run.

The modified Host APK must be resigned and zipaligned (“zipalign -v 4 <infile> <outfile>”, http://developer.android.com/tools/help/zipalign.html).

3.0 Host option files

The Host can be configured for advanced behavior by placing certain files with specific names in the /sdcard/Android/data/com.wisemo.host.v10/files/WsmHost folder[i]. Option files starts with “opt.”and it is the name of the file that determines the function, not the content of the file (the file size is zero).

The option files can be put in to the APK as described above, they can be manually copied or created via Windows with the “adb” tool, for example:

```
adb shell touch /sdcard/Android/data/com.wisemo.host.v10/files/WsmHost/opt.verbosity.high
```

The example will create “opt.verbosity.high” on the device connected to the Windows PC. The “adb” tool is installed with the WiseMo Host Manager (by default installed to “C:\Program Files (x86)\WiseMo\WiseMo RSM\Remote Desktop Host Manager”). For more information, please refer to paragraph “5.0 ADB and USB debugging”.

The specific option files are described below.

3.1 Miscellaneous option files

- "opt.verbosity.high" - put verbose mode log entries in logcat in release builds (high verbosity mode)
- "opt.no-rcbridge-ui" - if Host should not ask about installing add-on on start
- "opt.no-sony-rc" - if Sony RC API should not be used on this device.
- "opt.no-zebra-injection" - if Zebra injection API should not be used on this device.
- "opt.rotate180" - if captured image should be sent 180 degree rotated to Guest
- "opt.no-mediascreenshot" - do not use media projection screen capture on this device
- "opt.no-glscreenshot" - do not use GL screen capture

- "opt.force-use-rootft" - try using root-mode file transfer on device with API Level >= 23 (where root fs is disabled by default because it does not work)
- "opt.no-rootft" - disable root file transfer, used on devices with API level < 23.
- "opt.no-crashhandler"- disable crash handler registration (posix signal handler, java unhandled exception handler). Crash information can be saved to journal and received from customer.
- "opt.test-crafted-parcel" - When flag is set, Host works like if it works with old version of add-on that does not have BridgeMessage.FEATURE_FALLBACK_ANDROID_9 and BridgeMessage.FEATURE_SURFACE_ANDROID_10. For test of crafted messages even if we have add-on 2.7/3.1+. Crafted package is used only with RcBridge or root.
- "opt.disable-crafted-parcel" - when flag is set, Host do not use the crafted message for add-ons without BridgeMessage.FEATURE_FALLBACK_ANDROID_9 and FEATURE_SURFACE_ANDROID_10. In this case Host on Android 10 and 11 would ask for Media Projection.
- "opt.test-platform-cert" - file with a value inside that means the name of platform certificate (for example "lg3"). Host would behave like if it was launched on given platform, which is useful for testing (e.g. if we don't have physical device and we would like to verify if behavior is correct).
- "opt.no-mycloud-widget" - prevent "add to mycloud" widget. Requested when preparing with host.xml/lic in assets.

3.2 Option files for Journal

Journal is an out of proc Support Log program with some advanced features (like file rotation). By default in release mode journal files are saved in the app's private folder and can be only obtained with the "Support Log" UI in the Host. These are the options to configure the journal:

- "opt.journal.disable" - journal is disabled.
- "opt.journal.full" - Host saves the journal files in the Host's sdcard folder (\Android\data\com.wisemo.host.v10\files\journal\), so it can be extracted by a MDM tool with /sdcard access.

3.3 Option files related to Samsung

Flags to control the usage of KNOX and the device adminstrator role:

- "opt.no-knox-device-admin-prompt" - if Host should activate KNOX license without assigning Host as device admin (internal ref.: #133033)
- "opt.no-device-admin" - if Host should not ask user for device admin UI on start.
- "opt.no-esdk-rc" - if any Samsung RC API (new KNOX and legacy EDM) should not be used on this device.
- "opt.no-samsung-legacy" - if Host should not attempt to use a legacy Samsung EDM API
- "opt.no-samsung-knox" - if Host should not use Samsung KNOX API (new) on this device

Additional flags to control how height and witdh are used to capture the screen:

- "opt.disable-smt515-workaround" - when running on SM-T515 (android.os.Build.MODEL=="SM-T515") do not use the implemented workaround (internal ref: #132774)
- "opt.knox-no-screensize-fix" - force Host to not invoke method that fixes a problem with swapped height and width (use on Android 9 and above to tell the Host to work as before the fix (internal ref: #133553))
- "opt.knox-use-screensize-fix" or "opt.use-smt515-workaround"- use Samsung SM-T515 workaround on another Samsung devices, (internal ref: #132774). In this mode Host attempt to capture screen using a state machine.
- "opt.knox-set-screensize-reported" – Use reported screen sizes, if screen capture fails it will attempt to use the width and height switched (transposed mode).
- "opt.knox-set-screensize-transposed" - use the width and height switched (transposed mode).

4.0 Generating a logcat file

For advanced troubleshooting WiseMo might need a logcat file. The log cat file can be generated like this:

1. Enable USB debugging on the device. Please refer to paragraph “5.0 ADB and USB debugging”.
2. Connect the device to a Windows PC and open a command prompt
3. If you don’t have the ADB program we install a copy with the WiseMo Host Manager; downloaded here. The ADB program can be found in: C:\Program Files (x86)\WiseMo\WiseMo RSM\Remote Desktop Host Manager\
4. Create the option file /sdcard/Android/data/com.wisemo.host.v10/files/WsmHost/opt.verbosity.high to activate high verbosity logcat (e.g. can be done with a command: “adb shell touch /sdcard/Android/data/com.wisemo.host.v10/files/WsmHost/opt.verbosity.high”)
   Skip this step if the Host is not installed.
5. Save logcat with “adb logcat -v threadtime -d > %TEMP%\hostlog.txt”
6. The logcat is saved to %TEMP%\hostlog.txt. To go to the folder, issue the following command “cd %TEMP%”.

5.0 ADB and USB debugging

Android Debug Bridge (ADB) is a versatile command-line tool that lets you communicate with a device connected to your computer with a USB cable. The adb command facilitates a variety of device actions, such as installing and debugging apps, and it provides access to a Unix shell that you can use to run a variety of commands on a device.

Before tools like ADB can be used, USB debugging must be enabled. USB debugging enables communication between the Android device and a computer connected via USB.

Enabling USB debugging is a two-step process:

1. Enable developer options. The screen is hidden by default.

2. Enable USB debugging.

On Android 4.1 and lower, Developer options are available by default. On Android 4.2 and higher, Developer options must be enabled to see and the Developer options screen. To enable developer options, tap the Build Number option 7 times. This option can be found in one of the following locations, depending on Android version:

- Android 9 (API level 28) and higher: Settings > About Phone > Build Number
- Android 8.0.0 (API level 26) and Android 8.1.0 (API level 26): Settings > System > About Phone > Build Number
- Android 7.1 (API level 25) and lower: Settings > About Phone > Build Number

To enable USB debugging, toggle the USB debugging option on/off in the Developer Options menu. This option can be found in one of the following locations, depending on the Android version:

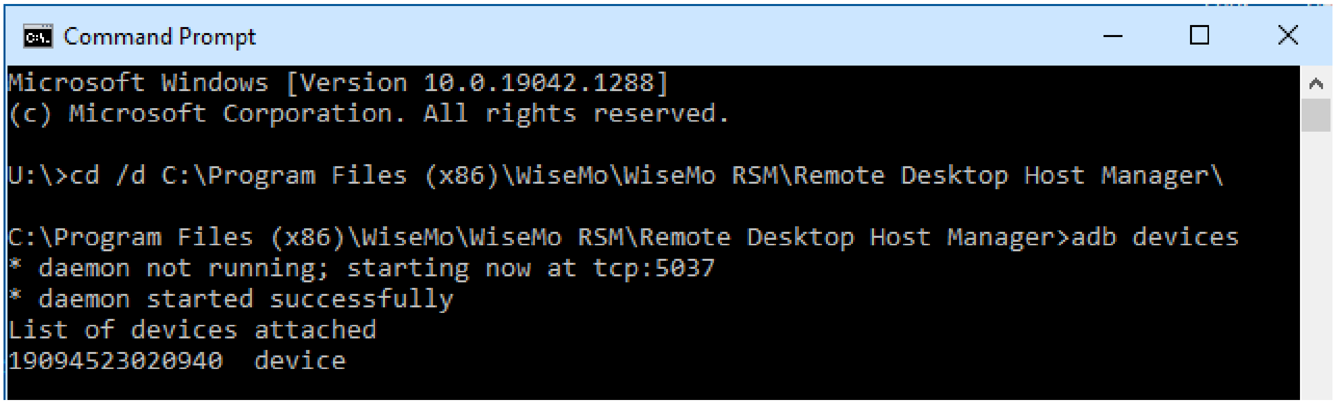
- Android 9 (API level 28) and higher: Settings > System > Advanced > Developer Options > USB debugging
- Android 8.0.0 (API level 26) and Android 8.1.0 (API level 26): Settings > System > Developer Options > USB debugging
- Android 7.1 (API level 25) and lower: Settings > Developer Options > USB debugging

Once USB debugging is enabled, you are able to use debugging tools like ADB. If you don't have the ADB program, a copy is installed with the WiseMo Host Manager. The ADB program can be found in: C:\Program Files (x86)\WiseMo\WiseMo RSM\Remote Desktop Host Manager\

To test whether USB debugging is configured correctly, open a command prompt, go to a folder with ADB.exe and write:

```
adb devices
```

and adb will list the devices connected to the computer with debugging enabled:



```
Microsoft Windows [Version 10.0.19042.1288]
(c) Microsoft Corporation. All rights reserved.

U:\>cd /d C:\Program Files (x86)\WiseMo\WiseMo RSM\Remote Desktop Host Manager\

C:\Program Files (x86)\WiseMo\WiseMo RSM\Remote Desktop Host Manager>adb devices
* daemon not running; starting now at tcp:5037
* daemon started successfully
List of devices attached
19094523020940  device
```

[i] Before version 18.0 build 2021-294, the folder /sdcard/WsmHost is used.