

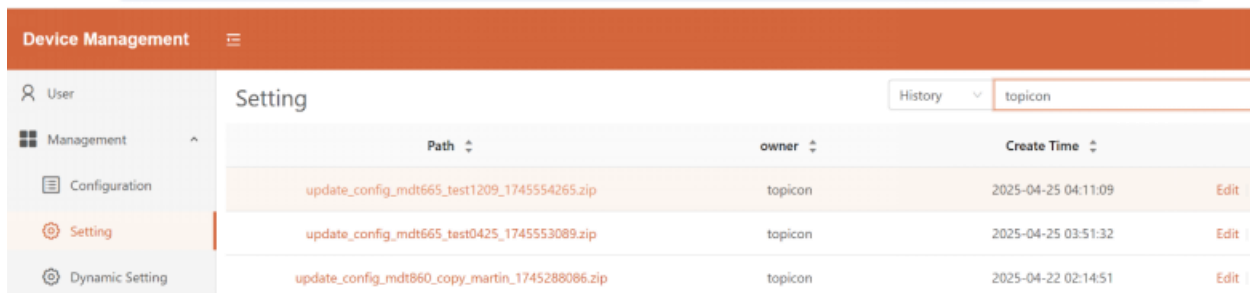
# Quick Start Guide for Kiosk Mode

## 1 Introduction

This document describes how to set up the PaceBlade products for kiosk mode, in which only one client application is displayed, and the end user usually has limited access to the device settings.

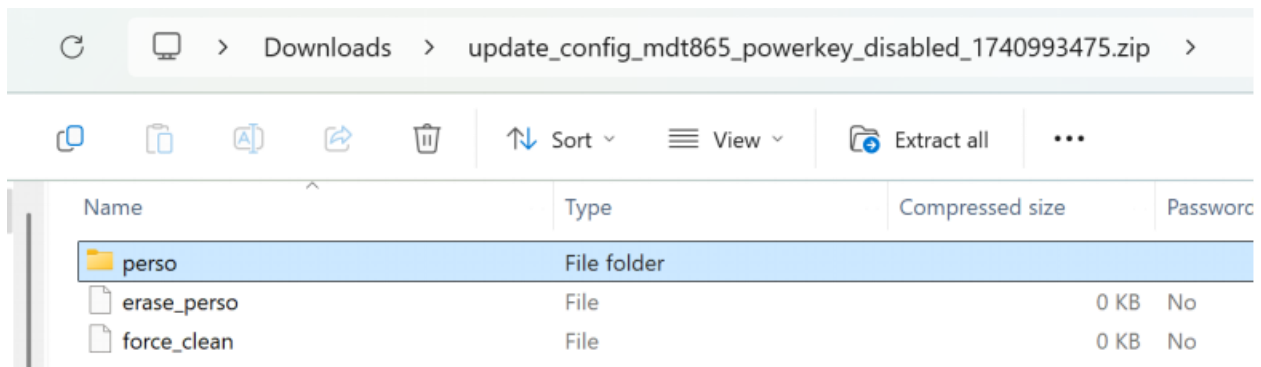
## 2 Kiosk mode demo

For section 2/3, it is an explanation for the kiosk mode perso folder using sd card. If you are using remote control server, after you made config file, you can also download the perso folder by clicking the update\_config\_xxx\_xxxx.zip



Path	owner	Create Time	
update_config_mdt665_test1209_1745554265.zip	topicon	2025-04-25 04:11:09	Edit
update_config_mdt665_test0425_1745553089.zip	topicon	2025-04-25 03:51:32	Edit
update_config_mdt860_copy_martin_1745288086.zip	topicon	2025-04-22 02:14:51	Edit

After downloaded the update\_config\_xxx\_xxxx.zip, it is same structure below (explanation in section 2/3)



Name	Type	Compressed size	Password
perso	File folder		
erase_perso	File	0 KB	No
force_clean	File	0 KB	No

### 2.1 Installing DemoKiosk

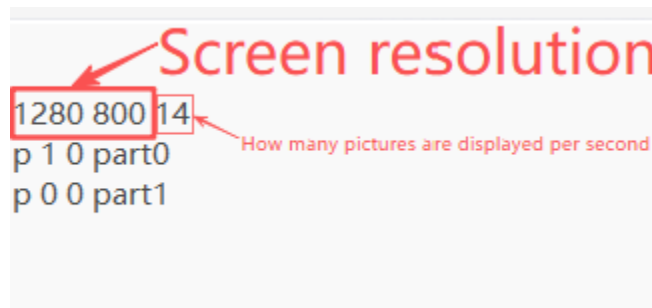
- Unzip the archive perso.zip at the root of a micro SD card. A directory “/perso” is created

- copy a firmware image at the root of the micro SD card
- insert this card in the device and click “Yes” when the firmware upgrade popup appears.
- Wait until the firmware upgrade is complete. The device will then restart, displaying the new splash screen and boot animation, and will start the DemoKiosk application directly.

## 2.2 Content of the perso.zip demo

- bootanimation.zip => the new boot animation, in Android format, needs to use the linux system computer to zip the bootanimation file.(use this command: `~/bootanimation zip -r -0 bootanimation.zip part0 part1 desc.txt`)

The image format can be jpg or png, and the image file name should use lowercase letters.



- DemoKiosk.apk => an Android package to install within the system. This one is configured in order to replace the default Android launcher (ie kiosk mode)
- Application=> install the application you need.
- Application overlay=>It is a special function for system package applications (for example, setting.apk).
- setting.txt => additional settings for the device. They will be discussed in section 3.2.

## 3 Setup kiosk mode

The device is configured in kiosk mode during firmware upgrade.

A directory called “perso” needs to be created at the root of a microSD card, and will contain customer specific data and settings.

As usual, the firmware image needs also to be copied at the root of this microSD.

### 3.1 “perso” directory content

- An optional boot animation, which must be called “bootanimation.zip”, in Android startup animation format.

- Default wallpaper, upload the file in image format to the device manager server (Setting->upload files->Default wallpaper).

If you update the wallpaper via SD card, You need to change the file name of the wallpaper to "default\_wallpaper" (no extension).

- Any number of Android packages (which must end with ".apk".) These packages will be copied on the device and added to its default package list.
- A text file called "setting.txt", which contains customer specific configurations and which will be described below.
- An xml file called "vendor\_apn.xml", which contains an additional APNs list. It will be discussed in section 3.3
- a directory called "sdcard", which contains any files and directory to be copied in the device's shared storage (so-called "internal SD") . ([Only available for Android 9 firmware.](#))
- an optional empty file called "selinux\_permissive" (resp. "selinux\_enforcing") to disable (resp. enable) Selinux enforcement on the device.

Selinux is a security framework which is used on Android version > 5.0

It will, among other things, prevent application code to access and modify the device file systems, even in the case this code gets super-user access permissions.

When Selinux is set to permissive mode through perso, security violations will be reported, but not blocked.

This option is intended for legacy applications which used super-user access in older Android versions.

After Selinux has been set to permissive mode, it will remain this way even in case of factory reset.

Restore enforcing mode must be done explicitly, with the "selinux\_enforcing" option file.

## 3.2 "setting.txt" file content

This text file contains a number of keys, one per line, which will be recorded in the device and modify its behavior.

### Keys which change the behavior

- **restrict\_access**

This key disables access to the settings from the top status bar. Only BRIGHTNESS, ROTATION, CHARGE and SCREEN SHOT are still accessible

All other settings and applications of the device can only be accessed through the client application (or through the default home application if no client is provided)

- **disable\_status\_panel**

This key disables the top sliding menu. Neither quick settings nor notifications will be accessible to the end-user.

- **restrict\_top\_bar** (Does not exists on Android9 +, replaced by “disable\_status\_panel”)

When the application *is in full screen mode*, this key disables access to the top bar by swiping the finger on the top of the screen.

The purpose is the same as “restrict\_access”, i.e. to limit the end-user access to the device, but “restrict\_top\_bar” is intended for full-screen applications which want to completely disable the top status bar.

- **sd\_launcher**

If this key is present, inserting a microSD card with an empty file called system\_mode (no extension) at the root of it will launch the default android desktop.

This is typically used when the kiosk application does not allow access to the settings, or if the distributor wants to have access to the browser, file manager, or manually install and remove applications.

- **global\_action\_disabled**

If this key is present, long press on the power button will no longer call the global action menu (power off and reset.)

This is typically used when the distributor wants to prohibit the end user from switching the device off.

- **long\_home\_intent=value**

(example : long\_home\_intent=com.android.settings/.Settings)

This key redefined the behavior of the long press on the HOME button. If defined, long press on HOME will start the activity whose component name is *value*.

- **short\_home\_intent=value**

(example:short\_home\_intent=com.android.settings/.Settings)

This key redefined the behavior of the short press on the HOME button. If defined, short press on HOME will start the activity whose component name is value.

- **short\_recent\_intent**

Example: `hk.topicon.intent.action.RECENT_PRESSED`. If this key is set, a press on APP\_SWITCH will send a broadcast intent whose action is given by the value (the behavior of the APP\_SWITCH key is not otherwise modified ; `recent_key_disabled` can be used to disable it.)

- **navbar\_disabled (only used on pre-Android 7 devices)**

If this key is present, the navigation bar will be disabled (only meaningful for devices with navigation bars, the 7" device does not have one.)

- **low\_bat\_warn\_disabled**

If this key is present, the low battery warning pop-up is disabled. The device will still shutdown when the battery level becomes critical.

- **volume\_keys\_disabled**

If this key is present, the volume buttons are disabled.

- **home\_key\_disables**

If this key is present, the home button is disabled.

- **recent\_key\_disabled**

If this key is present, the recent applications button is disabled.

- **report\_user\_activity=value (example: `report_user_activity=5`)**

If this key is present and different from 0, a broadcast message "`hk.topicon.utility.USER_ACTIVITY`" will be sent each time a user activity is detected on the device (touchpanel or key press.)

The value is the minimum period, in seconds, between messages. For instance a value of 5 means one message every 5 seconds, even if there is permanent user activity in the meantime.

- **cam\_tap\_pwr\_disabled**

If this key is present, double tap on the power button will not launch the camera (for Android 7+).

- **tethering\_disabled**

If this key is present, the Hotspot & tethering menu will be removed from the Settings menu

- **system\_launcher**

If this key is present, a key press combination on the BACK button will launch the default android desktop. The combination is LONG PRESS (more than 2 seconds) - SHORT PRESS - LONG PRESS (more than 2 seconds.) This is typically used when the kiosk application does not allow access to the settings, or if the distributor wants to have access to the browser, file manager, or manually install and remove applications. It works on Android 9.0 firmware version >=2.1.5

- **webrtc\_allowed**

This key bypasses user authorization for remote screen cases. If set, screen cast will be enabled directly without prompting the device user for confirmation.

When the webrt\_allowed key is set and the TestScreenCast.apk is installed via remote server, screen cast user consent will be bypassed for all applications (no confirmation popup dialog will be displayed, as they requested.) (e.g. TeamViewer Host application)

#### [TestScreenCast.apk](#)

- **restrict\_nav\_bar**

When the application is in immersive mode (navigation bar hidden), this key disables un-hiding the navigation bar by swiping the finger on the top or the bottom of the screen. The purpose is the same as “restrict\_access”, ie to limit the end-user access to the device, but “restrict\_nav\_bar” is intended for a kiosk application which wants to completely disable the navigation buttons.

- **restrict\_addon=value (example: restrict\_addon=wifi)**

When restrict\_access is selected, these keys define a comma separated list of function tiles that should still be accessible for the end user. The tiles must be a combination of:

location,wifi,airplane,bt,cast,cell,dnd

- **adb\_tcp\_port**

exemple : 5556. When this key is set, adb will listen to the specified tcp port number (it will still also listen on USB)

- **ethernet\_metered**

When this key is set, the ethernet connection will be managed as metered (paid), with limited background traffic (same as a cellular connection)

## Keys which change the default configuration

IMPORTANT: These keys change the default value of the settings, not the current value. Current value will be set to default at factory reset. A factory reset can be done at the same time as firmware upgrade by adding an empty file called "force\_clean" at the root of the microSD card.

- **timezone=value (example : timezone=Europe/Berlin)**

This key sets the default time zone.

- **language=value (example : language=it)**

This key sets the default language. The language code must follow the ISO 639 standard.

- **country=value (example : country=GB)**

This key sets the default country. The country code must follow the 2 letters ISO 3166 standard.

- **auto\_time\_zone=value (example : auto\_time\_zone=1)**

This key enables or disables the auto time zone feature (which allows the device to get the current time zone from the GSM network.) When this key is not present, the auto time zone is enabled.

- **accelerometer\_rotation=value (example : accelerometer\_rotation=1)**

This key enables or disables automatic screen rotation. When this key is not present, screen rotation is enabled.

- **dataroaming=value (example : dataroaming=false)**

This key enables or disables modem data roaming. When this key is not present, data roaming is disabled.

- **mobile\_data\_enabled=value (example : mobile\_data\_enabled=false)**

This key defines if mobile data is enabled or disabled by default. When this key is not present, mobile data is enabled.

- **adb\_default\_state=value (example : adb\_default\_state=0)**

Set the default status of the debugger service, enabled or disabled. This only changes the *default* status, the debugger can be manually enabled or disabled in the settings menu.

If this case is not present, the debugger is enabled by default.

- **screen\_auto\_brightness=value (example : screen\_auto\_brightness=0)**

Set the default status of the automatic screen brightness, enabled or disabled. This only changes the *default* status.

- **screen\_off\_timeout=value (the value is in milliseconds)**

Set the default duration of the screen timeout. The display will go off if no user action was done during the specified timeout (and if no wake lock is held). This only changes the default value. For never screen off, please input 2147483647. (representing  $2^{31}-1$ )

- **ntp\_server=value (example : ntp\_server=pool.ntp.org)**

Set the address of the NTP server (Network Time protocol.) By default it is set to "2.android.pool.ntp.org"

- **time\_12\_24=value (value is 12 or 24, example : time\_12\_24=24)**

Set the default time format, 12 or 24 hours.

- **usb\_default\_config=value**

(accepted values are "adb" and "mtp,adb", example : usb\_default\_config=adb)

Set the device's default USB mode. "mtp,adb" mode means mass storage access from PC and Android Debug Bridge are both enabled, "adb" means mass storage access is disabled.

- **install\_non\_market**

This key allows or forbids the installation of non-market applications

## Special purpose keys

The following keys have very specific purposes and behaviors, and should be avoided in normal operation. Please contact us for confirmation if you think you need one of those.

- **ntp\_disabled**

This key disables the device's Network Time Protocol client. It won't get the current time from the network. This is mainly intended for users who want to control every network access of the device.

- **nfc\_default**

This key enables or disables Nfc. When this key is not present, Nfc is enabled by default.

- **avin\_default\_camera**

Avinput default camera setting

- **font\_scale=value (example: font\_scale=0.85,1.0,1.15,1.3)**



Set default font size, choose the font size options "Small, Default, Large, Largest" on the device management server. If set via SD card, it can set font\_scale=value ( Small=0.85, Default=1.0, Large=1.15, Largest=1.3).

- **agps\_disabled**

This key disables support for Assisted Gps. The Gps engine will no longer attempt to download ephemeris from the network, and will do a standard fix instead. This is mainly intended for users who want to control every network access of the device.

- **portal\_detect\_disabled**

This key disables the auto-detection of web access portals (the kind that are used in the airport and hotel, for instance.) This is mainly intended for users who want to control every network access of the device.

- **data\_call\_wakelock (No longer useful on Android 12 device, since newer models of MTK devices can sleep with modem connected)**

This key disables sleep mode when modem data is connected. Only the display will be switched off.

WARNING : battery life when the screen is off will be considerably reduced. For instance on the mdt720 with 4300mA/h battery it will fall from more than one week to 15 hours.

- **time\_permission\_disabled**

If this key is set, no permission is needed to change the system time through AlarmManager.setTime().

(Normally SET\_TIME permission is needed, and it is a privileged permission only granted to system or signed applications.)

- **bt\_dev\_class=value (example : bt\_dev\_class=0x5A,0x01,0x14)**

This key overrides the Bluetooth class of the device. The syntax is bt\_dev\_class=<service class>,<major class>,<minor class>, where service, major and minor class are 1-byte hexadecimal numbers prefixed with 0x and capitalized.

- **can\_block\_system\_notif**

This key allows system notifications to be blocked in the same way as application notifications.

Notification blocker APP

The notification blocker must be installed *through perso*.

[NotificationBlocker\\_android\\_12.apk](#)

[NotificationBlocker\\_android\\_9.apk](#)

- **key\_map=value (example : key\_map=A,B,BACK,ENDCALL,HOME)**

This option defines the mapping of the mechanical keyboard of the device, if any.

The first button of the mapping is the left of the device, then the second next, and so on.

Model	Customizable button					
	1	2	3	4	5	6
MDT540/ MDT740	Power button					
MDT661/ MDT665	Left button	volume+	volume-	Power button		
MDT860	F1 (First button on the left)	F2	F3	F4	F5	Power button
MDT865/ MDT1065	5 (First button on the left)	4	3	volume+	volume-	Power button

For the barcode scanner model, key\_map=F is defined as the scan button. This option is only available for our scanner demo app.

Example below:

MDT540/MDT740

key\_map= BACK (in this case, you can change the power button to back function)

MDT860

key\_map= F1, F2, F3, ENDCALL, HOME, BACK

(in this case, first 3 button name is F1/2/3 which can use for your customized apk if needed, otherwise it is no function, then 4<sup>th</sup> button is end call, 5<sup>th</sup> button is HOME and power button will become back function )

MDT865/MDT1065

key\_map= F, 4, 3, ENDCALL, HOME, BACK


(in this case, first button will become scanner button, then next 2 button you can use for your apk if you defined it as 4 and 3, otherwise it will have 4/3 input (see in keyboard), 4<sup>th</sup> button is end call, 5<sup>th</sup> button is HOME and power button will become back function.

## Key Map Helper

You can define the key by Key Map Helper. There are four ways to choose from.

- 1) Select a special key: VOLUME\_UP, VOLUME\_DOWN, APP\_SWITCH, HOME, BACK, POWER.
- 2) Choose Application: Choose the APK from your PC that you want to activate with a key, you need to install this APK on your device.
- 3) Choose From Device: Choose the APK from your device that you want to activate with a key.
- 4) Manually enter the key name that you want to define.

P3:  Select a special key:  Launch an application, support .apk:

 avincamera\_for\_ahd\_2...

The name of the mapping entries must be one of the key labels defined in [Annex 1](#).

- **battery\_low\_wear (Only available for Android 9)**

If this key is set the battery will be in storage mode : it will only be recharged below 3.6V, and the device will power off after one minute out of charger

- **call\_ring\_disabled**

If this key is set, incoming call ringing it disabled

- **force\_headset**

If this key is set, the wired headset detection will be forced to ON.

- **tether\_ethernet\_default**

If this key is set, Ethernet tethering will be enabled automatically when an Ethernet interface is available.

- **status\_bar\_disabled (Currently, this feature is only available for MDT760 model)**

This key disables the status bar, with permanent immersive mode and no swipe for the transient bar.

- **force\_bt\_discoverable ( (It works on Android 12 firmware version >=1.8.9)**

If this key is set and Bluetooth is enabled, the device will always be in discoverable mode, even outside of the pairing menu

- **need\_gms\_compliant**

This key is only taken into account in Gms type firmware. It is needed to be fully compliant with CTS, and it will disable non-CTS hardware, like video input

- **avin\_replace\_camera**

If this key is set and the device supports it, an external video channel will be used instead of the back camera. On mdt880, by default 3 external video channels and a back camera are available. When this key is set, 4 external video channels are available but the back camera is disabled.

- **force\_pin\_code=value (example: force\_pin\_code=1234)**

Define a code PIN that will be automatically used with the SIM card (if the SIM card does not need a pin code it will be ignored)

- **default\_mock\_location**

example : hk.topicon.mocklocservice. When this key is set, the mock location will be enabled and the mock location package name will be set to the specified value.

- **model=value (example :model=xyz7)**

This option overrides the default ro.build.model property of the device, which is used for several user-visible labels, like the name reported on the PC connection and the Bluetooth name. Use it with care, as changing the model name may affect previously established pairing/registration made by application software.

- **config\_ethernet=value**

(example:ip=192.168.132.151/24 gateway=192.168.132.1 dns=8.8.8.8)

This key configures a static ip address on the Ethernet interface, for devices that support it. If not set, DHCP will be used.

- **remote\_server**

example : ssl://myserver.com:8883 .\*need to specify tcp/ssl\*

This key defines the address and port on which to control device management server (by Docker)

If the default value of this field changes, the device will not reconnect to this remote upgrade server.

The default remote server IP is "ssl://fw.topicon.hk"

- **eth\_video\_hub**

If this key is set, Ethernet USB interface, if present, will be used for the video hub.

- **tether\_ethernet\_addr (example: 192.168.66.1/24 )**

This feature is only available for MDT865 and MDT1065 firmware version  $\geq 1.4.3$ .

This key configures the DHCP address range when ethernet tethering is used.

- **tether\_wifi\_addr (It works on Android 12 firmware version  $\geq 1.8.1$ )**

This key configures the dhcp address range when wifi tethering is used.Example: 192.168.66.1/24

- **tether\_dhcp\_option (example: max\_addr=23 lease\_time=86400)**

This key setup is an extra option for the Dhcp server, if set it is used for all tethering modes (WiFi, Ethernet and USB). Two options are currently supported : max\_addr: the maximum number of addresses that will be attributed (starting from 0). For instance if max\_addr=23 and the Dhcp server address is 10.0.67.1/24, only addresses between 10.0.67.2 and 10.0.67.23 will be distributed to clients. lease\_time: Dhcp lease time, in seconds. The minimum (and default) value is 3600.

- **Brand**

example : NewBrand. This option overrides the ro.product.brand property of the device.

BUT, the application only reads it the first time, when it is installed. I.e. if you change the perso after that, you need to reinstall the application

- **manufacturer**

example : NewManufacturer. This option overrides the ro.product.manufacturer property of the device.


**Get setting list from device (It works on Android 12 firmware version  $\geq 1.9.0$ )**

Click “Choose from device” to select the device that you want to get the settings from.


Global Settings, System Settings and Secure Settings can be obtained.


You can apply these settings to your device by upgrading the config file.


#### get setting list from device

 Choose from device

IMEI:350930740104124:

 Global Settings ×

 System Settings ×

 Secure Settings ×

- **“vendor\_apn.xml” file content**

It is an xml file which contains a list of APNs to be added to the device's default set.

**IMPORTANT** : this list is taken into account only after factory reset, or after pressing Settings > Wireless & network > More > Cellular networks > Access Point Names > menu (3 vertical dots) > Reset to default.

This file must be compliant with Android apn.xml version 8. Please find a template below.

```
<?xml version="1.0" encoding="utf-8"?>
<apns version="8">
  <apn carrier="Example" mcc="202" mnc="01" apn="Internet" user="user" password="pass"
type="default,supl" />
  <apn carrier="Example MMS" mcc="202" mnc="01" apn="Mms" mmsc="101.102.103.103:8000"
mmsproxy="100.100.100.100" mmsport="8080" type="mms" />
</apns>
```

Further examples can be seen in the device's default APNs list, located in /system/etc/apns-conf.xml

The current implementation does not check for duplicate, so please make sure no apn in vendor\_apn.xml are already defined in /system/etc/apns-conf.xml

- **WiFi config**

-It is an xml file which contains a list of Wifi access points that can be uploaded on the remote upgrade server, in Management > Setting - Edit > upload files -> Wifi Config, or it can be put in the /perso directory of the upgrade SD card.

-If wifi\_default.xml is provided through perso, wifi will be on by default (note that this is only the default value, i.e. after a factory reset.)

- in both case it must be called wifi\_default.xml

- the syntax must follow the following template:

```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>

<WifiConfigStoreData>

  <int name="Version" value="1" />

  <NetworkList>

    <Network>

      <WifiConfiguration>

        <string name="SSID">&quot;topicon_hk&quot;</string>

        <string name="PreSharedKey">&quot;topicon2014&quot;</string>

      </WifiConfiguration>

    </Network>

    <Network>

      <WifiConfiguration>

        <string name="SSID">&quot;ct2020&quot;</string>

        <string name="PreSharedKey">&quot;12345678&quot;</string>

      </WifiConfiguration>

    </Network>

  </NetworkList>

</WifiConfigStoreData>
```

- you need to keep the "&quot;" historical they are used to make the difference between newer WPA wifi hotspot and older WEP ones

- note that Wifi will still be powered off by default

## 4 Configuring and Android application to be used in kiosk mode

To appear on top and be launched by default, an Android activity must react to the HOME intent.

AndroidManifest.xml should typically contains :

```
<activity android:name="com.my.package.MyKiosk"
    android:launchMode="singleInstance"
<!-- "singleTask" may be used instead of "singleInstance" -->
    android:stateNotNeeded="true" <!-- optional -->
<!-- other activity settings here -->
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.HOME" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
```

For more information about the Activity settings in kiosk mode, more information can be found here :

<http://developer.android.com/guide/topics/manifest/activity-element.html>

- **Intent Firewall**

The firewall rules need to be uploaded from the remote upgrade server, in Management > Setting - Edit > upload files > Intent firewall. The file must be called **ifw.xml**

IntentFirewall is not (yet) a public Android API, but it has been around for several years and you can find a good documentation about the file format here:

<https://carteryagemann.com/pages/android-intent-firewall.html>

```
<rules>
```

```
<!-- block access to Google Chrome's IntentDispatcher for the VIEW action -->
```

```
<activity block="true" log="false">
```

```
    <intent-filter>
```

```
        <action name="android.intent.action.VIEW" />
```

```
    </intent-filter>
```

```
    <component-filter name="com.android.chrome/com.google.android.apps.chrome.IntentDispatcher" />
```



```
</activity>
```

```
<!-- block access to the location setting menu -->
```

```
<activity block="true" log="false">
```

```
<intent-filter>
```

```
<action name="android.settings.LOCATION_SOURCE_SETTINGS" />
```

```
</intent-filter>
```

```
</activity>
```

```
</rules>
```

- **Application to disable** (It works on Android 12 firmware version  $\geq 1.9.4$ )

### Remove the Google SetupWizard

1) Install the default provisioning application below :

For Android 12: [ProvisionHide.apk](#)

For Android 14: [ProvisionHideHipri.apk](#)

It is the same application that is used in the mdt865 firmware without GMS services.

2) To disable the Google Setup wizard, you have to provide a disable\_app.xml file in the perso :  
Setting - Edit > upload files > Applications to disable

The list is a xml file in the following format :

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<disable_app>
```

```
<pkg name="com.google.android.setupwizard" />
```

```
</disable_app>
```

3) "Applications to disable" are only taken into account **after a factory reset**. So you need to enable it in your perso.

List of system packages to be disabled (from functional or user point of view it is the same as removing it.)

WARNING: if you disable a critical package the device may not even startup successfully, so use this with caution.

Note: applications marked as "coreApp" in their AndroidManifest.xml cannot be disabled (but, like Settings, they could be overridden by a perso app)

## Remove multiple apps

The list is a xml file in the following format :

```
<?xml version="1.0" encoding="utf-8"?>

<disable_app>

  <pkg name="the.first.package.to.disable" />

  <pkg name="the second packages to disable" />

</disable_app>
```

- **Disable Navigation bar** ( note that this is only the default value, i.e. after a factory reset.)
  - Remove the manual bar (3 buttons at the bottom). Install the below apk in our remote control server (Setting - Edit > upload files > Application Overlay)

[GestureNavOverlay.apk](#)

## create child setting

A child setting can only save the difference with the master setting, and it can be applied to the device that has already applied the master setting.

Only applicable to models with Android 12 and above.

## Annex 1 : list of key labels

UNKNOWN  
SOFT\_LEFT  
SOFT\_RIGHT  
HOME  
BACK  
CALL  
ENDCALL  
0  
1  
2  
3  
4  
5  
6  
7  
8  
9  
STAR  
POUND  
DPAD\_UP  
DPAD\_DOWN  
DPAD\_LEFT  
DPAD\_RIGHT  
DPAD\_CENTER  
VOLUME\_UP  
VOLUME\_DOWN  
POWER  
CAMERA  
CLEAR  
A  
B  
C  
D  
E  
F  
G  
H

I  
J  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
W  
X  
Y  
Z  
COMMA  
PERIOD  
ALT\_LEFT  
ALT\_RIGHT  
SHIFT\_LEFT  
SHIFT\_RIGHT  
TAB  
SPACE  
SYM  
EXPLORER  
ENVELOPE  
ENTER  
DEL  
GRAVE  
MINUS  
EQUALS  
LEFT\_BRACKET  
RIGHT\_BRACKET  
BACKSLASH  
SEMICOLON  
APOSTROPHE  
SLASH  
AT  
NUM

HEADSETHOOK  
FOCUS // \*Camera\* focus  
PLUS  
MENU  
NOTIFICATION  
SEARCH  
MEDIA\_PLAY\_PAUSE  
MEDIA\_STOP  
MEDIA\_NEXT  
MEDIA\_PREVIOUS  
MEDIA\_REWIND  
MEDIA\_FAST\_FORWARD  
MUTE  
PAGE\_UP  
PAGE\_DOWN  
PICTSYMBOLS  
SWITCH\_CHARSET  
BUTTON\_A  
BUTTON\_B  
BUTTON\_C  
BUTTON\_X  
BUTTON\_Y  
BUTTON\_Z  
BUTTON\_L1  
BUTTON\_R1  
BUTTON\_L2  
BUTTON\_R2  
BUTTON\_THUMBL  
BUTTON\_THUMBR  
BUTTON\_START  
BUTTON\_SELECT  
BUTTON\_MODE  
ESCAPE  
FORWARD\_DEL  
CTRL\_LEFT  
CTRL\_RIGHT  
CAPS\_LOCK  
SCROLL\_LOCK  
META\_LEFT  
META\_RIGHT  
FUNCTION  
SYSRQ

BREAK  
MOVE\_HOME  
MOVE\_END  
INSERT  
FORWARD  
MEDIA\_PLAY  
MEDIA\_PAUSE  
MEDIA\_CLOSE  
MEDIA\_EJECT  
MEDIA\_RECORD  
F1  
F2  
F3  
F4  
F5  
F6  
F7  
F8  
F9  
F10  
F11  
F12  
NUM\_LOCK  
NUMPAD\_0  
NUMPAD\_1  
NUMPAD\_2  
NUMPAD\_3  
NUMPAD\_4  
NUMPAD\_5  
NUMPAD\_6  
NUMPAD\_7  
NUMPAD\_8  
NUMPAD\_9  
NUMPAD\_DIVIDE  
NUMPAD\_MULTIPLY  
NUMPAD\_SUBTRACT  
NUMPAD\_ADD  
NUMPAD\_DOT  
NUMPAD\_COMMA  
NUMPAD\_ENTER  
NUMPAD\_EQUALS  
NUMPAD\_LEFT\_PAREN

NUMPAD\_RIGHT\_PAREN  
VOLUME\_MUTE  
INFO  
CHANNEL\_UP  
CHANNEL\_DOWN  
ZOOM\_IN  
ZOOM\_OUT  
TV  
WINDOW  
GUIDE  
DVR  
BOOKMARK  
CAPTIONS  
SETTINGS  
TV\_POWER  
TV\_INPUT  
STB\_POWER  
STB\_INPUT  
AVR\_POWER  
AVR\_INPUT  
PROG\_RED  
PROG\_GREEN  
PROG\_YELLOW  
PROG\_BLUE  
APP\_SWITCH  
BUTTON\_1  
BUTTON\_2  
BUTTON\_3  
BUTTON\_4  
BUTTON\_5  
BUTTON\_6  
BUTTON\_7  
BUTTON\_8  
BUTTON\_9  
BUTTON\_10  
BUTTON\_11  
BUTTON\_12  
BUTTON\_13  
BUTTON\_14  
BUTTON\_15  
BUTTON\_16  
LANGUAGE\_SWITCH

MANNER\_MODE  
3D\_MODE  
CONTACTS  
CALENDAR  
MUSIC  
CALCULATOR  
ZENKAKU\_HANKAKU  
EISU  
MUHENKAN  
HENKAN  
KATAKANA\_HIRAGANA  
YEN  
RO  
KANA  
ASSIST  
BRIGHTNESS\_DOWN  
BRIGHTNESS\_UP  
MEDIA\_AUDIO\_TRACK  
SLEEP  
WAKEUP  
PAIRING  
MEDIA\_TOP\_MENU  
11  
12  
LAST\_CHANNEL  
TV\_DATA\_SERVICE  
VOICE\_ASSIST  
TV\_RADIO\_SERVICE  
TV\_TELETEXT  
TV\_NUMBER\_ENTRY  
TV\_TERRESTRIAL\_ANALOG  
TV\_TERRESTRIAL\_DIGITAL  
TV\_SATELLITE  
TV\_SATELLITE\_BS  
TV\_SATELLITE\_CS  
TV\_SATELLITE\_SERVICE  
TV\_NETWORK  
TV\_ANTENNA\_CABLE  
TV\_INPUT\_HDMI\_1  
TV\_INPUT\_HDMI\_2  
TV\_INPUT\_HDMI\_3  
TV\_INPUT\_HDMI\_4



TV\_INPUT\_COMPOSITE\_1  
TV\_INPUT\_COMPOSITE\_2  
TV\_INPUT\_COMPONENT\_1  
TV\_INPUT\_COMPONENT\_2  
TV\_INPUT\_VGA\_1  
TV\_AUDIO\_DESCRIPTION  
TV\_AUDIO\_DESCRIPTION\_MIX\_UP  
TV\_AUDIO\_DESCRIPTION\_MIX\_DOWN  
TV\_ZOOM\_MODE  
TV\_CONTENTS\_MENU  
TV\_MEDIA\_CONTEXT\_MENU  
TV\_TIMER\_PROGRAMMING  
HELP  
NAVIGATE\_PREVIOUS  
NAVIGATE\_NEXT  
NAVIGATE\_IN  
NAVIGATE\_OUT  
STEM\_PRIMARY  
STEM\_1  
STEM\_2  
STEM\_3  
DPAD\_UP\_LEFT  
DPAD\_DOWN\_LEFT  
DPAD\_UP\_RIGHT  
DPAD\_DOWN\_RIGHT  
MEDIA\_SKIP\_FORWARD  
MEDIA\_SKIP\_BACKWARD  
MEDIA\_STEP\_FORWARD  
MEDIA\_STEP\_BACKWARD  
SOFT\_SLEEP  
CUT  
COPY  
PASTE