

App Install User's Guide

Android-ALL-App-Install User Guide

Apr 20, 2012.

TeleChips

Preliminary

DISCLAIMER

All information and data contained in this material are without any commitment, are not to be considered as an offer for conclusion of a contract, nor shall they be construed as to create any liability. Any new issue of this material invalidates previous issues. Product availability and delivery are exclusively subject to our respective order confirmation form; the same applies to orders based on development samples delivered. By this publication, Telechips, Inc. does not assume responsibility for patent infringements or other rights of third parties that may result from its use.

Further, Telechips, Inc. reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of Telechips, Inc.

This product is designed for general purpose, and accordingly customer be responsible for all or any of intellectual property licenses required for actual application. Telechips, Inc. does not provide any indemnification for any intellectual properties owned by third party.

Telechips, Inc. can not ensure that this application is the proper and sufficient one for any other purposes but the one explicitly expressed herein. Telechips, Inc. is not responsible for any special, indirect, incidental or consequential damage or loss whatsoever resulting from the use of this application for other purposes.

COPYRIGHT STATEMENT

Copyright in the material provided by Telechips, Inc. is owned by Telechips unless otherwise noted.

For reproduction or use of Telechips' copyright material, permission should be sought from Telechips. That permission, if given, will be subject to conditions that Telechips' name should be included and interest in the material should be acknowledged when the material is reproduced or quoted, either in whole or in part. You must not copy, adapt, publish, distribute or commercialize any contents contained in the material in any manner without the written permission of Telechips. Trade marks used in Telechips' copyright material are the property of Telechips.

Important Notice

This product may include technology owned by Microsoft Corporation and in this case it cannot be used or distributed without a license from Microsoft Licensing, GP.

For customers who use licensed Codec ICs and/or licensed codec firmware of mp3:

"Supply of this product does not convey a license nor imply any right to distribute content created with this product in revenue-generating broadcast systems (terrestrial. Satellite, cable and/or other distribution channels), streaming applications(via internet, intranets and/or other networks), other content distribution systems(pay-audio or audio-on-demand applications and the like) or on physical media(compact discs, digital versatile discs, semiconductor chips, hard drives, memory cards and the like). An independent license for such use is required. For details, please visit <http://mp3licensing.com>".

For customers who use other firmware of mp3:

"Supply of this product does not convey a license under the relevant intellectual property of Thomson and/or Fraunhofer Gesellschaft nor imply any right to use this product in any finished end user or ready-to-use final product. An independent license for such use is required. For details, please visit <http://mp3licensing.com>".

For customers who use Digital Wave DRA solution:

"Supply of this implementation of DRA technology does not convey a license nor imply any right to this implementation in any finished end-user or ready-to-use terminal product. An independent license for such use is required."

For customers who use DTS technology:

"Supply of this implementation of DTS technology does not convey a license, exhaust DTS' rights in the implementation, or imply a right under any patent, or any other industrial or intellectual property right of DTS to use, offer for sale, sell, or import such implementation in any finished end-user or ready-to-use final product. Notice is hereby provided that a license from DTS is required prior to such use."

"This product made under license to U.S. Patents 5,451,942; 5,956,674; 5,974,380; 5,978,762; 6,487,535; 6,226,616 and/or foreign counterparts."

"© 1996 – 2010 DTS, Inc."

Preliminary

Revision History

Date	Version	Description
2010-10-01	1.00	Initial release
2011-04-08	1.01	Add "how to change external storage for installing applications"

TABLE OF CONTENTS**Contents**

Revision History	3
TABLE OF CONTENTS	4
Contents	4
1 App installation on the external storage	5
2 App Install Location	5
3 Backward Compatibility	6
4 Applications That Should NOT Install on External Storage	7
5 Applications That Should Install on External Storage	8
6 How to change external storage for installing applications	9

1 App installation on the external storage

The Android platform now allows applications to request installation onto [the external storage](#), as an alternative to installation onto the device's internal memory.

		Framework		
		INTERNAL_ONLY (default state)	AUTO	PREFER_EXTERNAL (External Storage)
App(SDK 2.1)		Internal (don't move to external)	Internal (don't move to external)	External (don't move to external)
App (SDK 2.2)	internalOnly	Internal (don't move to external)	Internal (don't move to external)	Internal (don't move to external)
	Auto	Internal (can move to external)	Internal (can move to external)	Internal (can move to external)
	preferExternal	External (can move to external)	External (can move to external)	External (can move to external)

Application developers can express the preferred installation location for their applications by means of a new attribute of `<manifest>` in the manifest file, `android:installLocation`. The attribute supports three values: "internalOnly", "preferExternal", and "auto". At install time, the system checks the value of `android:installLocation` and installs the application .apk according to the preferred location, if possible. If the application has requested external installation, the system installs it into a private, encrypted partition in the external media. Once an application .apk is installed externally, the system lets the user change the storage location of the .apk and move it onto the device's internal memory if needed (and vice versa), through Manage Applications in the user settings.

By default, the system installs all applications onto the device's internal memory, except for those that explicitly request external installation. This means that the system will always install legacy applications onto internal memory, since they do not have access to the `android:installLocation` attribute. However, it is possible to configure and compile a legacy application such that it is installed internally on older versions of the platform and externally on Android 2.2 and later platforms, if necessary.

Note that requesting installation onto the device's external media is not suitable for all applications, particularly because the external media may be removable and unmounting/remounting may disrupt the user experience and system settings.

For more information about setting a preferred install location for your application, including a discussion of what types of applications should and should not request external installation, please read the [App Install Location](#) document.

2 App Install Location

Beginning with API Level 8, you can allow your application to be installed on the external storage. This is an optional feature you can declare for your application with the `android:installLocation` manifest attribute. If you do *not* declare this attribute, your application will be installed on the internal storage only and it cannot be moved to the external storage.

To allow the system to install your application on the external storage, modify your manifest file to include the `android:installLocation` attribute in the `<manifest>` element, with a value of either `"preferExternal"` or `"auto"`. For example:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    android:installLocation="preferExternal"
    ... >
```

If you declare `"preferExternal"`, you request that your application be installed on the external storage, but the system does not guarantee that your application will be installed on the external storage. If the external storage is full, the system will install it on the internal storage. The user can also move your application between the two locations.

If you declare `"auto"`, you indicate that your application may be installed on the internal storage, but you don't have a preference of install location. The system will decide where to install your application based on several factors. The user can also move your application between the two locations.

When your application is installed on the external storage:

- There is no effect on the application performance so long as the external storage is mounted on the device.
- The `.apk` file is saved on the external storage, but all private user data, databases, optimized `.dex` files, and extracted native code are saved on the internal device memory.
- The unique container in which your application is stored is encrypted with a randomly generated key that can be decrypted only by the device that originally installed it. Thus, an application installed on an SD card works for only one device.
- The user can move your application to the internal storage through the system settings.

Warning: When the user enables USB mass storage to share files with a computer or unmounts the SD card via the system settings, the external storage is unmounted from the device and all applications running on the external storage are immediately killed.

3 Backward Compatibility

The ability for your application to install on the external storage is a feature available only on devices running API Level 8 (Android 2.2) or greater. Existing applications that were built prior to API Level 8 will always install on the internal storage and cannot be moved to the external storage (even on devices with API Level 8). However, if your application is designed to support an API Level *lower than* 8, you can choose to support this feature for devices with API Level 8 or greater and still be compatible with devices using an API Level lower than 8.

To allow installation on external storage and remain compatible with versions lower than API Level 8:

1. Include the `android:installLocation` attribute with a value of `"auto"` or `"preferExternal"` in the `<manifest>` element.
2. Leave your `android:minSdkVersion` attribute as is (something *less than* "8") and be certain that your application code uses only APIs compatible with that level.

3. In order to compile your application, change your build target to API Level 8. This is necessary because older Android libraries don't understand the `android:installLocation` attribute and will not compile your application when it's present.

When your application is installed on a device with an API Level lower than 8, the `android:installLocation` attribute is ignored and the application is installed on the internal storage.

Caution: Although XML markup such as this will be ignored by older platforms, you must be careful not to use programming APIs introduced in API Level 8 while your `minSdkVersion` is less than "8", unless you perform the work necessary to provide backward compatibility in your code. For information about building backward compatibility in your application code, see the [Backward Compatibility](#) article.

4 Applications That Should NOT Install on External Storage

When the user enables USB mass storage to share files with their computer (or otherwise unmounts or removes the external storage), any application installed on the external storage and currently running is killed. The system effectively becomes unaware of the application until mass storage is disabled and the external storage is remounted on the device. Besides killing the application and making it unavailable to the user, this can break some types of applications in a more serious way. In order for your application to consistently behave as expected, you **should not** allow your application to be installed on the external storage if it uses any of the following features, due to the cited consequences when the external storage is unmounted:

Services

Your running [Service](#) will be killed and will not be restarted when external storage is remounted. You can, however, register for the [ACTION_EXTERNAL_APPLICATIONS_AVAILABLE](#) broadcast Intent, which will notify your application when applications installed on external storage have become available to the system again. At which time, you can restart your Service.

Alarm Services

Your alarms registered with [AlarmManager](#) will be cancelled. You must manually re-register any alarms when external storage is remounted.

Input Method Engines

Your [IME](#) will be replaced by the default IME. When external storage is remounted, the user can open system settings to enable your IME again.

Live Wallpapers

Your running [Live Wallpaper](#) will be replaced by the default Live Wallpaper. When external storage is remounted, the user can select your Live Wallpaper again.

Live Folders

Your [Live Folder](#) will be removed from the home screen. When external storage is remounted, the user can add your Live Folder to the home screen again.

App Widgets

Your [App Widget](#) will be removed from the home screen. When external storage is remounted, your App Widget

will *not* be available for the user to select until the system resets the home application (usually not until a system reboot).

Account Managers

Your accounts created with [AccountManager](#) will disappear until external storage is remounted.

Sync Adapters

Your [AbstractThreadedSyncAdapter](#) and all its sync functionality will not work until external storage is remounted.

Device Administrators

Your [DeviceAdminReceiver](#) and all its admin capabilities will be disabled, which can have unforeseeable consequences for the device functionality, which may persist after external storage is remounted.

Broadcast Receivers listening for "boot completed"

The system delivers the [ACTION_BOOT_COMPLETED](#) broadcast before the external storage is mounted to the device. If your application is installed on the external storage, it can never receive this broadcast.

If your application uses any of the features listed above, you **should not** allow your application to install on external storage. By default, the system *will not* allow your application to install on the external storage, so you don't need to worry about your existing applications. However, if you're certain that your application should never be installed on the external storage, then you should make this clear by declaring [android:installLocation](#) with a value of "[internalOnly](#)".

Though this does not change the default behavior, it explicitly states that your application should only be installed on the internal storage and serves as a reminder to you and other developers that this decision has been made.

5 Applications That Should Install on External Storage

In simple terms, anything that does not use the features listed in the previous section are safe when installed on external storage. Large games are more commonly the types of applications that should allow installation on external storage, because games don't typically provide additional services when inactive. When external storage becomes unavailable and a game process is killed, there should be no visible effect when the storage becomes available again and the user restarts the game (assuming that the game properly saved its state during the normal [Activity lifecycle](#)).

If your application requires several megabytes for the APK file, you should carefully consider whether to enable the application to install on the external storage so that users can preserve space on their internal storage.

6 How to change external storage for installing applications

The Telechips support several storages: nand, sdcard, and scsi.

The ICS of Telechips is able to select one of several storages for installing application.

In order to change external storage for installing applications, you need to declare "dev_asec_mount" of mount_type in the vold.fstab file. "dev_asec_mount" is added by Telechips.

The path of vold.fstab:

devices/telechips/tcc8800/vold.fstab
 devices/telechips/tcc8920/vold.fstab
 devices/telechips/tcc8920st/vold.fstab

The vold.fstab file structure:

mount_type	Label	mount_point	partition	sysfs_path
dev_asec_mount	nand	/mnt/sdcard	auto	/devices/virtual/block/ndda default
dev_mount	sdcard	/mnt/ext_sd	auto	/devices/platform/tcc-sdhc
dev_mount	usb_sda	/mnt/usb_sda1	all	/devices/platform/dwc_otg /devices/platform/tcc-ohci /devices/platform/tcc-ehci
dev_mount	usb_sdb	/mnt/usb_sdb1	all	/devices/platform/dwc_otg /devices/platform/tcc-ohci /devices/platform/tcc-ehci

Example: select sdcard storage for installing application.

In the vold.fstab file

```
dev_mount nand /mnt/sdcard auto /devices/virtual/block/ndda
dev_asec_mount sdcard /mnt/ext_sd auto /devices/platform/tcc-sdhc0.0/mmc_host/mmc0
dev_mount usb_sda /mnt/usb_sda1 all /devices/platform/dwc_otg /devices/platform/tcc-ohci /devices/platform/tcc-ehci
dev_mount usb_sdb /mnt/usb_sdb1 all /devices/platform/dwc_otg /devices/platform/tcc-ohci /devices/platform/tcc-ehci
```

Reference

<http://developer.android.com/sdk/android-2.2.html>

<http://developer.android.com/guide/appendix/install-location.html>