navigation
- Main page
- Forums

rootzwiki
- Follow Us on Twitter
- RootzWiki Store
- Contact US

handsets
- All Manufacturers
- Acer
- Geeks Phone
- HTC
- Huawei
- LG
- Motorola
- Samsung
- Sony Ericsson

tablets
- All Manufacturers
- Advent
- Archos
- Asus
- Barnes and Noble
- Dell

# How to install and use ADB

**Contents** [hide]

1 What is ADB?
2 Installation
  2.1 What You Need
  2.2 Installing Packages
    2.2.1 Windows
    2.2.2 Mac
    2.2.3 Linux
  2.3 Prepping the Android Device to Accept ADB
3 Using ADB
  3.1 The Basics
    3.1.1 Installing Apps
    3.1.2 Uninstall Apps
    3.1.3 Copying Files To and From The Device
    3.1.4 Rebooting
  3.2 Advanced
    3.2.1 Remounting /system for Read/Write
    3.2.2 Shell Access
    3.2.3 Getting Logs
4 Terms

## What is ADB? [edit]

ADB stands for the android debugging bridge and is used for testing and debugging purposes by developers.

However, we like to get more out of our devices, and its a great way to fix things.

Knowing adb can mean the difference between a paperweight and a working phone.

## Installation [edit]

- Elocity
- HTC
- Huawei
- Motorola
- Nothion Ink
- Samsung
- Viewsonic

dictionary

- A-F
- G-L
- M-S
- T-Z
- 0-9

search

Go Search

google adsense


## Installation [edit]

Please Note that we're assuming this is your first time going through installation of the Android SDK so we're talking about the current version (which is currently on Version 10).

## What You Need [edit]

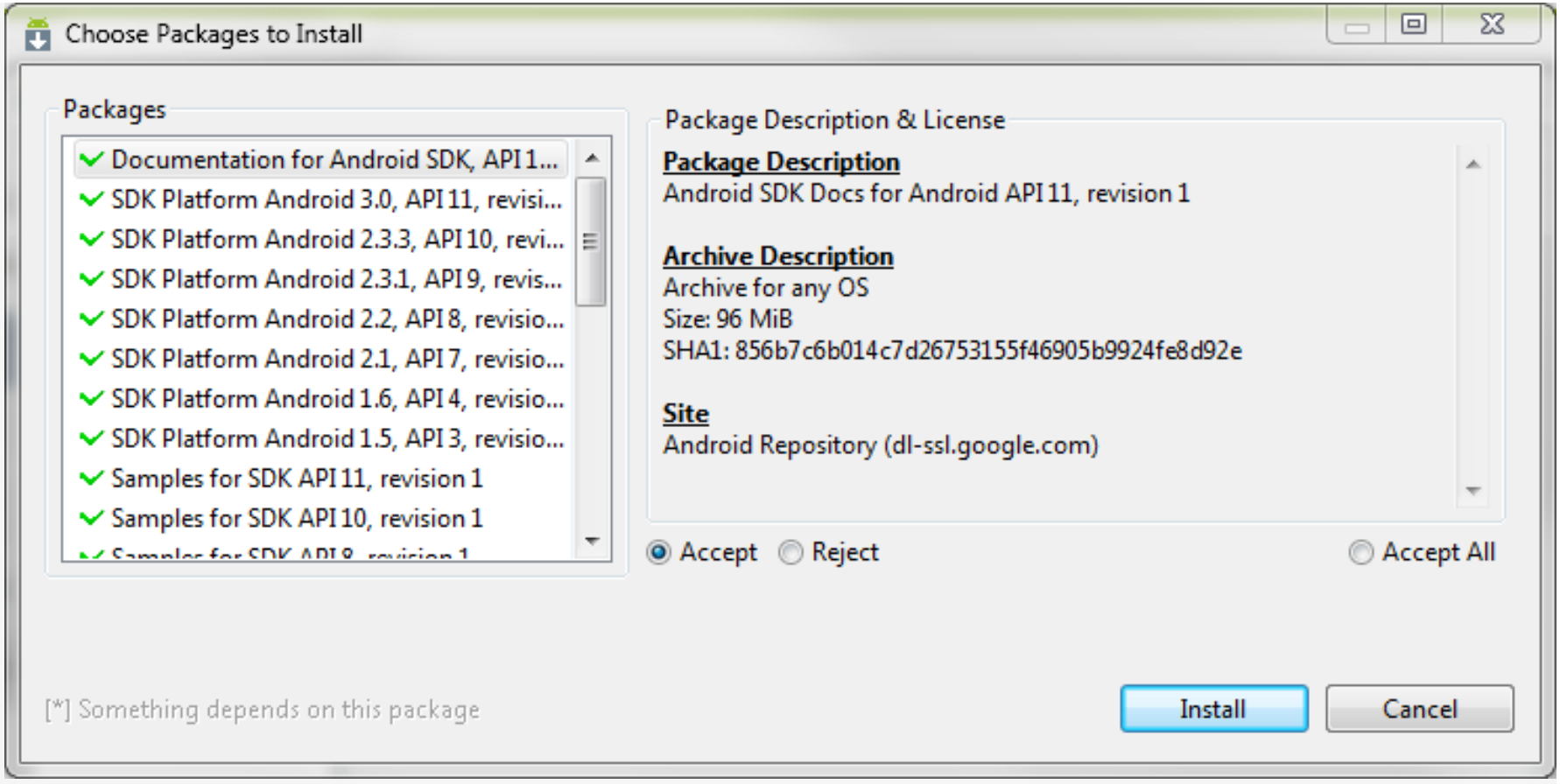
Sun/Oracle Java Development Kit (JDK)
Android Software Development Kit (SDK)
USB Drivers for your Android device

Note to Windows users: We recommend that you download the Zip version of the Android SDK so we don't have to copy long paths

## Installing Packages [edit]

### Windows [edit]

If you haven't already, run through the installation for the JDK (Java Development Kit).
Open the Zip file and you'll see a folder titled **android-sdk-windows**, copy that folder to an easy to remember location (we recommend copying it to the C drive).
Open the folder you just extracted (C:\android-sdk-windows) and run **SDK Manager.exe**.
You'll see the Package manager (as seen below):




At this point I normally just choose the **Accept All** radio button and just download everything that's on that list (which includes the sources for developing on Android 1.5, 1.6, 2.1-update1, 2.2, 2.3.1, 2.3.3, and 3.0) but the primary package that we want to download is the **Android SDK Tools, revision 10** and **Android SDK Platform-tools, revision 3), once installation is complete and you restart ADB you're pretty much done but lets go a step further.**

1) Click on the Start Menu and right click on *My Computer* and choose *Properties*.
2a) **Windows Vista/7 Users Only:** On the left side, click on *Advanced system settings*
2b) **Windows XP Users Only:** Click on the tab that says *Advanced*
3) Click on *Environment Variables*
4) Under **System variables**, look for *Path* and double click on it
This will open up another box for editing the Path variable.
5) In the end of the **Variable value**, add the following:

```
;C:\android-sdk-windows\tools;C:\android-sdk-windows\platform-tools
```

- Special pages
- Printable version
- Permanent link


(Note: this is assuming you have installed the Android SDK in the C:\android-sdk-windows path, if you had installed it elsewhere, please update to match, space does not matter but make sure you sperate each new path with a semi-colon (;))
6) Keep clicking OK until your out.

What you just did was set up a "path" for windows to look into so when you're using adb and/or fastboot, you don't have to be in told folders at all.

To test to make sure everything is working, open a command prompt and type **adb help** and **fastboot help**. If you get help information back then your paths are setup correctly and you're ready to start using ADB and fastboot.

### Mac [edit]

[Todo]

### Linux [edit]

[Todo]

## Prepping the Android Device to Accept ADB [edit]

Now that we got the Android Development Kit tools installed, it's time to do some prep work for you to be able to commutate with your Android device.

First off, install the USB drivers (this can normally be done by simply plugging your Android device into your computer via the USB cable and set the USB mode to Mass Storage or PC Mode or by downloading the drivers off the manufacturer's web site). We're not going to go into great details on here, just know that your computer must be able to at least see the phone before we move on.

After the USB drivers has been install, on your Android device, go to:
Menu (usually there's a soft key you press to open up the menu)
Tap **Settings**
Tap **Applications**
Tap **Development**
Place a check mark next to **Enable USB Debugging**

One quick note: If you get device not found/connected, please reboot your phone. DJ05 has a quirk in it where ADBD randomly crashes on boot.A reboot will fix this.

To make sure you're able to talk to your Android device via ADB, open up a command prompt or thermal and type:

```
adb devices
```

If you're able to see a serial number then it's talking to your phone.

## Using ADB [edit]

### The Basics [edit]

#### Installing Apps [edit]

First thing you should know is that all Android Apps ends with a file extension of **.apk**, try to install any other apps that doesn't end in **.apk** will result in an error.

Why would you want to install an app using ADB? Simply because you don't need to worry about having to turning Unknown Sources for starters and you don't have to click additional buttons on your phone. This is known as a side install, allows you to bypass the market and not worry about putting the APK file onto your SD Card and looking for it and then installing it.

Basic command is simply:

```
adb install filename.apk
```

It'll install the package and will tell you any errors in the process.

If you want to install an app but lock it (keep it from running) right off the bat, then you can use the following command:

```
adb install -l filename.apk
```

And there are times when you might just want to reinstall a package, to do this use the following command:

```
adb install -r filename.apk
```

And for those whose on Android 2.2 and above who wishes to install to the SD Card instead of to the internal storage can use this command:

```
adb install -s filename.apk
```

All the switches above are interchangeable as well, meaning if you, say, want to install an application to your SD Card and

lock the application, you can do the following:

```
adb install -l -s filename.apk
```

*Do I need to be rooted to use this?* No, you do not need to be rooted to install packages via ADB

### Uninstall Apps [edit]

Uninstalling apps can be done through ADB as well. To uninstall an app simply use the following command:

```
adb uninstall packagename
```

This will completely remove the package including data and cache directories, but what if you want to keep the data and cache directories? Look no further then this command here:

```
 adb uninstall -k packagename
```

Please note that you MUST match the name of the package you wish to uninstall (including the cases, e.g. YouTube)

### Copying Files To and From The Device [edit]

I, personally, leave my USB Mode on my Droid X to Charge Only and use ADB to copy files to and from my device (Scsa20), it really that simple as using:

```
adb push filename /sdcard/whatever/
```

For copying a file to your device and:

```
adb pull /sdcard/whatever/filename.whatever
```

For copying a file from your phone to whatever folder your in in your command prompt or thermal. Don't want to copy to the current directory you're in on your computer but don't want to change directories? No problem, use:

```
adb pull /sdcard/whatever/filename.whatever C:\Directory
```

(this is assuming you're on a windows machine, if you're on a Mac or Linux based computer, the command is the same but

instead of using **C:**, it's more likely something like **/home/user/** (or wherever you wish to put it to))

Please note that in order for you to *push* (copy to) a file anywhere but to the /sdcard/ directory (your SD Card) you'll need to make sure you have write access (see the Advanced section for more details).

### Rebooting [edit]

You might be working with your Android device and you make a change so now you need to reboot the device for the changes to take effect. One way is to shut down your device and powering it back up but since your phone is connected to your computer and you've been using ADB already, why not use ADB again to reboot the phone? It's simple as using this command:

```
adb reboot
```

This will cause your device to reboot normally, but now lets say you want to reboot into the recovery to work on something (or to format your phone). You can use the following command:

```
adb reboot recovery
```

In retrospect, if you need to boot into the bootloader for any reason (like flashing your Android device back to stock which you do sometimes need to be in the bootloader), you can run either:

```
adb reboot bootloader
```

or

```
adb reboot-bootloader
```

## Advanced [edit]

ADB is so advanced that the basics isn't enough for what you need to do, so the more advanced stuff is here.

Please note that most advanced options here requires that you have root access. If you don't have root access then a lot of the commands will fail.

### Remounting /system for Read/Write [edit]

There's plenty of times where you might need to write to the /system directory of your Android device. An easy way to make sure you have read/write access the fast and easy way is to use the following command:

```
adb remount
```

### Shell Access [edit]

The Android OS is based off of Unix so you do have shell access. To access the shell from your computer, simply use the following command:

```
adb shell
```

More then likely you'll be presented with a dollar sign (**$**), this means your in standard user account. If you're rooted, you can now type **su** followed by enter and you should see a pound sign (#).

If you go into the shell and you're taken straight to the pound sign (#), then more then likely you have a ROM or a mod installed that will take you to super user access

If you want, if there's a command you wish to run in the shell but don't wish to waste your time accessing the shell to run that one command, you can simply use:

```
adb shell command
```

This will run whatever command after shell without having to actually access the shell.

### Getting Logs [edit]

If you wish to see what's going on, why your phone randomly rebooted, etc., you can pull the logs by using this command:

```
adb logcat
```

This will allow you to view the log and see what's going on.

## Terms [edit]

ADB = Android Debug Bridge

ADBD = Android Debug Bridge Daemon
APK = Android Package

Written by CaptainKrtek at XDA Developers, derived from written tutorial by Adrynalyne.