# Tegra 250 Development Kit
# Android Setup Experience

Version 20101117

# Contents

# Welcome to Tegra

So you have your NVIDIA Tegra 250 Developer Kit!  Now you just want to develop.  Here's how!

This guide is a fast track to get you from a boxed developer kit to compiling and running your first sample on the Android OS using a Windows XP or Windows 7 host machine.

These instructions cover the setup of a "Bare board" Tegra devkit, pictured below:



**"Bare Board" Devkit**

# STEP 1: Gather Additional Hardware

## Gather the following:

- Windows based PC with 1 USB port free; ideally with a 2GHz CPU and 1GB RAM
- USB cable – Type-A to Mini-B
- Cable to connect the external display and the devkit
- SD card or USB "thumb" drive is recommended for additional storage (not needed for this guide)
- External display supporting VGA (15-pin D-Sub), HDMI, or DVI (via HDMI-to-DVI adapter)
- USB mouse
- USB keyboard

# STEP 2:  Unboxing your Devkit

Unpack the components

- Devkit main board (contains the NVIDIA Tegra chip and several connectors)
- 15V power adapter (USA 120V mains power connection)
- Threaded WiFi antenna
- Expansion board (contains serial port, LEDs, and 3 buttons)

# STEP 3: Setup the Connections

Make the basic connections:

- **Expansion board** – Connect the two ribbon cables of the expansion board to the matching pair of headers on the devkit main board's front edge. Take care to ensure that all of the pins align.

- **Display connection** – Connect your VGA, HDMI, or DVI-D (via an HDMI-to-DVI adapter) display to the appropriate port on the devkit main board's back edge. Connect only one display.

- **WiFi Antenna** – The threaded end of the WiFi antenna should be firmly screwed onto the WiFi antenna connector on the rear edge of the devkit main board.

- **Keyboard and mouse** – Connect the USB keyboard and mouse to the two stacked USB jacks. If the mouse and keyboard are power hungry devices a powered hub is recommended.

- **External storage** – If you have a SD card, insert it into the SD card slot on the left edge of the devkit main board. If you have a USB "thumb" drive, insert it into the USB port on the right side of the devkit main board.

- **Power** – Place the "ACOK" switch, on the front right edge of the board, to "BATT". Connect the supplied 15V power supply to the power jack on the rear edge of the devkit main board.

# STEP 4: Install the Android Support Pack

## Download the Android OS Support Pack Installer here:

http://developer.nvidia.com/tegra/downloads

Download the latest Android Froyo OS pack Windows installer (MSI file) from this page.

## Run the Installer MSI:

1) [ Click "Next" ]

2) Decide to accept the EULA

3) [ Select "Complete" ] – to install the entire support pack.

4) [ Click "Install" ]

5) [ Click "Finish" ]

# STEP 5: Flash the Devkit with the Android OS

**Place the DevKit into Recovery Mode:**

1) Press and hold "ForceRecovery" button on the attached expansion board.

2) Press the "ONKEY*" button until the power LEDs (near the center of the main board) light up.

3) Release the "ForceRecovery" button after 1-2 seconds.

## Install the Recovery Mode driver:

1) Connect the USB cable from the free USB port (Type-A) on your PC to the USB port (Mini-B) on the left-rear corner of the devkit.  Referred to as "Development Cable".

2) Install the Recovery Mode driver if asked.  Use the "install from a specific location" option in the "New Device Wizard" dialog box.  It is located here in the directory referred to by:

Start Menu: All Programs: NVIDIA Corporation: Tegra 250 Android Platform: Version <XXXXXXXX>: Platform Directory

In the usbpcdriver subdirectory of this platform directory.

This driver has not been submitted for Windows Logo testing, you may have to click "Continue Anyway" to install the driver.  Note that the driver can take up to 5 minutes to install.

---

**Note:**  Windows 7 may not indicate the new device's presence at times.  If it does not:

1) [ Press and hold the Windows Key ]

2) [ Press the Break key ]

3) [ Release the Windows Key ]

4) [ Select "Device Manager" ] – on the left hand pane of the "System Information" dialog

5) Unrecognized devices will be shown as opened groupings when the "Device Manager" window initially opens.  One of these devices should be the devkit.  Select it and install the USB driver listed above.

---

If you receive an error message while installing the driver, see the appendix at the end of this document on *Troubleshooting the Recovery Mode Driver*.

## Flash your OS:

1) Open an Explorer window to the Android OS support pack you installed previously. It has the OS image in it. It should be found here:

    Start Menu: All Programs: NVIDIA Corporation: Tegra 250 Android Platform: Version <XXXXXXXX>: Platform Directory

2) Select one of two batch files, depending on your display device. See the chart below.

| Display Type | Flash Batch Script |
|---|---|
| CRT, VGA | nvflash_1gb_crt.bat |
| HDMI, DVI (via adapter) | nvflash_1gb_hdmi.bat |
| If built-in "LVDS" LCD is supplied (not common) | nvflash_1gb_lcd.bat |

3) Double click to run the appropriate .bat file. The flashing process will begin immediately. At the end of a successful flash, remove the USB cable connection to the PC. If you do not, Windows will ask for another driver, please cancel out of the "New Device Wizard". We will install this driver later. The last few lines of a successful flash will look something like this:

```
sending file: bootloader.bin
\ 903580/903580 bytes sent
bootloader.bin sent successfully
sending file: boot.img
| 2529280/2529280 bytes sent
boot.img sent successfully
sending file: system.img
| 79590720/79590720 bytes sent
system.img sent successfully
Press enter to continue:
```

4) The device should have rebooted to the desired video out mode with the Android desktop. If it did not, stop here and see the appendix at the end of this document on *Troubleshooting the Recovery Mode Driver*.

# STEP 6: Install the Java Development Kit (JDK)

## Download the Java SE Development Kit:

Use your search engine to find "java se development kit", or use the link:

http://www.oracle.com/technetwork/java/javase/downloads/index.html

Download the JDK for your PC platform, likely JDK 6 (most recent release as of the time of this document was 22)

## Run the installer:

When you reach the "Custom Setup" page in the installer (the one that allows you to select the desired location and components):

1)  [ Click "Change. . ." ] - next to "Install to:".

2)  Install the JDK in a pathname *without spaces*, as spaces in the path of the JDK can cause problems later in your development.  We will recommend and reference "C:\Java\jdk" for the remainder of this document.

3)  Accept default install packages and continue.

4)  If prompted to install the JRE, change the default install directory to a directory without spaces as above. (e.g. "C:\Java\jre6")

# STEP 7: Install the Android Software Development Kit (SDK)

## Download the Android SDK Installer here:

http://developer.android.com/sdk/index.html

Download the zipfile containing the latest version for your platform (as of the time of this document that was version 7).

## Setup the SDK:

1) Extract the zip file to an appropriate SDK directory of your choosing. For this guide we'll use `C:\Android`. So once extracted, the SDK will be contained in: `C:\Android\android-sdk-windows`.

2) Run "`SDK Manager.exe`" contained in `C:\Android\android-sdk-windows`. It may also be called "`SDK Setup.exe`", but will always be in `C:\Android\android-sdk-windows`.

> **Note:**  If during the install an error occurs similar to, "HTTPS SSL error.  You might want to force download through HTTP in the settings," do the following steps:
>
> 1) [ Click "Settings" ]
>
> 2) [ Select "Force https://... Sources to be fetched using http://..." ]
>
> 3)  Return to appropriate step.

3) The "Choose Packages to Install" dialog should appear.

4) [ Select "Accept All" ]

5) [ Click "Install" ]

6) Wait for the download/install to finish and close the dialog.

7) Close the "Android SDK and AVD Manager".

# STEP 8: Make Android Debug Bridge (ADB) Modifications

**Modifying the Android ADB driver:**

1) Edit the file:

   `C:\Android\android-sdk-windows\usb_driver\android_winusb.inf`

2) Add the following 3 lines in the section "`[Google.NTx86]`":

```
;NVIDIA Tegra
%SingleAdbInterface%    = USB_Install, USB\VID_0955&PID_7000
%CompositeAdbInterface% = USB_Install, USB\VID_0955&PID_7100&MI_01
```

3) Add those same 3 lines in the section "`[Google.NTamd64]`".

4) Save the file.

# STEP 9: Make ADB Connection to your Host PC

## Make the connection:

1) Connect the Development Cable.

2) If asked to install a driver, install using the "install from a specific location" option in the "New Device Wizard" dialog box. It is located here:

```
C:\Android\android-sdk-windows\usb_driver
```

This driver has not been submitted for Windows Logo testing; you may have to click "Continue Anyway" to install the driver.

---

**Note:** Windows 7 may not bring up a "New Device Wizard", in this case:

1) [ Press and hold the Windows Key ]

2) [ Press the Break key ]

3) [ Release the Windows Key ]

4) [ Select "Device Manager" ] – on the left hand pane of the "System Information" dialog

5) In the Device Manager, open the "Universal Serial Bus controllers".

6) If you see, "Android Debug Bridge Interface" or "Android Composite ADB Interface" or something similar, continue to the next step, otherwise move to the next section.

7) Right-click on the Debug entry from above, select "Update Driver Software . . .".

8) Select the "Browse" option, select the "Let me pick" option.

9) Follow all the next steps, as Windows will not always install the driver properly.

10) [ Click "Have Disk . . ." ]

11) [ "Browse . . ." ] - to `C:\Android\android-sdk-windows\usb_driver`.

12) [ Click "OK" ]

13) [ Click "Next" ]

14) You should now see "Android Phone" in your Device Manager.

---

## Check the connection:

1) Open a Windows command prompt.

   a.  [ Click *Start -> Run…* ]

   b.  Enter "cmd" in the "Open" field.

   c.  [ Click "OK" ]

2) Change to the Android tools directory by entering (later, we will add this directory to the global path):

```
cd /d C:\Android\android-sdk-windows\tools
```

3) Make sure adb lists the Tegra device by entering the following 2 commands:

```
adb kill-server
adb devices
```

4) The output from "adb devices" should look similar to (device id is unique per device):

```
List of devices attached
040372421601A00D        device
```

5) If there is no Tegra device listed, you must stop here and see the Appendix *Troubleshooting the ADB Connection*.

# STEP 10: Install Cygwin

## Download the Cygwin setup:

The installer may be found online at http://www.cygwin.com/, currently:

http://www.cygwin.com/setup.exe

## Run the setup app:

1) Select "Install from Internet".

2) Set the root directory (we will use `C:\Cygwin`).

3) Select your desired local package directory.

4) Select your connection (Direct Connection recommended).

5) Select your download site.

6) Wait for package listing download.

7) Make sure under the "*Devel*" tree that "*make: The GNU version of the 'make' utility*" is set to install the latest version, "*3.81-2*" as of this writing, and not set to "*Skip*".

8) In addition, select to install the "*Python*" subtree and *ncurses* under "*Utils*" (for later use with debugging scripts).

9) Click "Next" as required to begin the download and install.

10) Wait for the packages to download/install.  It takes awhile.

11) Click "Finish".

# STEP 11: Install the Android Native Development Kit (NDK)

## Download the Android NDK setup:

http://developer.android.com/sdk/ndk/index.html

Download the NDK zipfile for the Windows platform.  As of the time of the document, the version was "r4b", and the filename was thus `android-ndk-r4b-windows.zip`.

## Install and setup the NDK:

Extract the zip file to an appropriate NDK directory of your choosing.  For ease we'll use the same directory we installed the SDK in: `C:\Android`.  So once extracted, the NDK will be contained in `C:\Android\android-ndk-r4b`.

(Versions of the NDK prior to 4 required additional setup scripts, but these are no longer required).

# STEP 12: Install and Setup the Eclipse IDE

## Download the Eclipse IDE for Java Developers setup:

http://www.eclipse.org/downloads/

Look for the download for "Eclipse IDE for Java EE Developers" and download the associated zipfile.

## Install Eclipse:

1) Extract the zip file to an appropriate Eclipse directory of your choosing. For ease we'll use the same directory we installed the SDK in: `C:\Android`. So once extracted, Eclipse will be contained in `C:\Android\eclipse`.

2) Start a "Cygwin Bash Shell" from the Cygwin shortcut in your start button.

3) From within the bash shell, run eclipse:

   `/cygdrive/c/Android/eclipse/eclipse.exe &`

4) If the "Workspace Launcher" dialog comes up, choose a directory for your projects. For ease we'll use `C:\Android\workspace` and select this as your default.

> **Note:** It is recommended for best behavior of the tools that Eclipse is always run from within a Cygwin bash shell!

## Download/Install the Android Development Tools (ADT) Plugin:

1) Select *Help -> Install New Software*.

2) [ Click "Add . . ." ] – to add a new site.

   a. Enter "Android Plugin" in the "Name" field.

   b. Enter "`https://dl-ssl.google.com/android/eclipse/`" in the "Location" field.

3) [ Click "OK" ]

4) [ Check "Developer Tools"under "Name" ] – to install the development tools.

5) [ Click "Next" ] - to begin install

6) [ Click "Next" ] – to accept the ADT items.

7) [ Click "I accept the terms of the license agreement" and then "Finish" ] - to accept the license agreement.

8) Wait until the download finishes, you may have to accept some of the downloads.

9) [ Click "Yes" ] - if a dialog asks if you want to restart, otherwise restart Eclipse manually.

## Configure ADT Plugin:

1) Select *Window -> Preferences*.

2) [ Select "Android" ] - from the left panel.

3) If prompted that the SDK location is not set, [ Click "OK" ] to acknowledge.

4) Set the "SDK Location" by browsing to the Android SDK path install tree. (e.g. "C:\Android\android-sdk-windows")

5) [ Click "Apply" ]

6) [ Click "OK" ]


## Install the C/C++ Development Tools (CDT) Plugin:

1) Select *Help -> Install New Software*.

2) In the entry box "Work with", enter: "http://download.eclipse.org/tools/cdt/releases/helios".

   If you try to do this using "add", you may receive an error that this is a duplicate site. Simply enter the URL in the initial text entry.

3) [ Click "OK" ]

4) [ Check "CDT Main Features" ] – to install the main features.

5) [ Click "Next" ]

6) [ Click "Next" ] – to accept the CDT items.

7) [ Click "Finish" ] – to accept the license agreement.

8) Wait until the download finishes, you may have to accept some of the downloads.

9) [ Click "Yes" ] - if a dialog asks if you want to restart, otherwise restart Eclipse manually.

# STEP 13: Setup Environment Variables

**Enter new Environment Variables:**

1) In Windows, right click on *My Computer -> Properties -> Advanced -> Environment Variables*.

2) [ Click "New" ] - in the "User variables for . . ." section.

   a. Enter "NDKROOT" in the "Variable name" edit box.

   b. Enter your NDK installation path in the "Variable value" edit box. (e.g. "C:\Android\android-ndk-r4b")

3) [ Click "OK" ]

4) [ Click "New" ] - in the "User variables for . . ." section.

   a. Enter "CYGWIN_HOME" in the "Variable name" edit box.

   b. Enter your Cygwin installation path  in the "Variable value" edit box. (e.g. "C:\Cygwin")

5) [ Click "OK" ]

6) If you have a "PATH" variable highlight it and click "Edit", otherwise click "New" in the "User variables for . . ." section.

   a. "PATH" should be entered in the "Variable name" edit box.

   b. Append the JDK bin directory (e.g. "C:\Java\jdk\bin") and the Android tools directory (e.g. "C:\Android\android-sdk-windows\tools") in the "Variable value" edit box.

> **Note:** After changing these values, close all existing Cygwin Bash Shells and re-open them to pull the new values!

# STEP 14: Compile the Sample Application

## Import the OpenGL ES 2.0 (vertex/fragment shaders) sample:

> **Note:** It is recommended for best behavior of the tools that Eclipse is always run from within a Cygwin Bash Shell!

1) In Eclipse, the first time it is launched, a "Welcome" screen is presented. Before continuing on, make sure you select "Workbench" (the curled arrow icon).

2) Open the Android Project Wizard by selecting, *File -> New -> Project -> Android -> Android Project -> Next*. A new dialog, "New Android Project," should appear.

3) [ Enter "hello-gl2" ] - in "Project name".

4) [ Select "Create project from existing source" ]

5) "Browse..." to "\samples\hello-gl2" inside of NDKROOT to set the "Location".

6) Make sure the most current Android SDK is selected under "Build Target", as of this writing "Android 2.2".

7) [ Click "Finish" ]


## Configure the NDK library to compile within Eclipse:

1) In Eclipse, prepare a C/C++ project by going to *File -> New -> Other*.

   a. Open the "C/C++" tree and select "Convert to a C/C++ Project" in the "New" dialog.

   b. [ Click "Next" ]

   c. [ Select "hello-gl2" ] - project in the "Candidates for conversion" panel.

   d. Make sure "Makefile project" and "—Other Toolchain –" are selected in the lower two panels.

   e. [ Click "Finish" ]

2) [ Click "Yes" ] - if the "Open Associated Perspective" dialog appears.

3) Right-click on "hello-gl2" and select "Properties" in the "Project Explorer" tab.

4) [ Select "Environment" ] - in the left panel, it's under the "C/C++ Build" group.

   a. [ Click "Add . . ." ]

   b. "Name" should be "PATH" and "Value" should be "${CYGWIN_HOME}\bin".

c.   [ Click "OK" ] - in the "New variable" dialog.

5)   [ Double-click "C/C++ Build" ] - in the left panel of the properties dialog.

6)   In the "Builder Settings" tab:

   a.   Change the default Build Command ("`make`") to be:

   ```
   bash ${NDKROOT}/ndk-build
   ```

   b.   Set the "Build directory" to be:

   ```
   ${workspace_loc:/hello-gl2/jni}
   ```

   c.   [ Click "Apply" ]

7)   [ Click "OK" ] – to exit the properties window

## Compile the application:

1)   In Eclipse, compile by going to *Project -> Build All*.

2)   If you only get warnings in the "Problems" tab on the bottom panel, move on to the next step.

3)   If you get the error, "*The project cannot be built until build path errors are resolved*":

   a.   Close Eclipse.

   b.   Reopen Eclipse, which should automatically reopen your project.

   c.   Restart at step 1.

4)   If you got a different error, don't proceed until it is solved.  Check here for help:

   http://developer.android.com/intl/de/sdk/ndk/index.html

# STEP 15: Run the Sample Application

1) In the Eclipse "Project Explorer" panel, right-click on the "hello-gl2" project.

2) Select *Run As -> Android Application*.  This should automatically install and launch the app on your Tegra 250 developer kit.  It will be located in the Android slider panel as "GL2JNI".

Running the app will draw a green triangle on a background which is pulsing from black to white.  Note that this app cannot be run in the emulator; it does not support OpenGL ES 2.0 hardware emulation yet.

# Summing Up

You now have run the entire course from: setting up your kit; to compiling an Android app which uses the Android SDK, NDK, and OpenGL ES 2.0 with shaders; to deploying the new application.  This exercise should give you a quick start to using the Android SDK and Dalvik, writing (or porting) your own native C/C++ code, and using OpenGL ES.  It is *highly recommended* you at least browse the following pages and documents:

Google's Hello World walk-through:
http://developer.android.com/intl/de/resources/tutorials/hello-world.html

Google Android – Developing in Eclipse guide:
http://developer.android.com/intl/de/guide/developing/eclipse-adt.html

Android NDK Overview:
http://developer.android.com/intl/de/sdk/ndk/index.html#overview

The Android home link for future reference:
http://developer.android.com/intl/de/index.html

Tegra developer site (forums, SDKs, documentation, and news):
http://developer.nvidia.com/tegra/

# Appendix: Troubleshooting the Recovery Mode Driver

## Flashing Produces Device Not Found

If running the flash bat file gives the output similar to:

```
C:\Program Files (x86)\NVIDIA Corporation\tegra_froyo_20101105>"nvflash.exe" --bct
flash.bct --setbct --bl bootloader.bin --configfile flash.cfg --odmdata 0x30000011
--create --go
Nvflash started
USB device not foundPress enter to continue:
```

Then try the following:

1) Ensure that the device is connected to the PC via USB

2) Ensure that the device is on and in recovery mode

Open the Device Manager and check under "Universal Serial Bus controllers". You should see:

"NVIDIA USB Boot-recovery driver for mobile devices"

If this device is missing or has an error marker next to it, install or reinstall the driver via the method listed in the following section *Manually Install the Driver*.

## Flashing Halts without Completing

If the flashing process produces the following output similar to:

```
C:\Program Files (x86)\NVIDIA Corporation\tegra_froyo_20101105>"nvflash.exe" --b
ct flash.bct --setbct --bl bootloader.bin --configfile flash.cfg --odmdata 0x300
00011 --create --go
Nvflash started
rcm version 0X20001
System Information:
   chip name: t20
   chip id: 0x20 major: 1 minor: 2
   chip sku: 0x8
   chip uid: 0x0808104842204617
   macrovision: disabled
   hdcp: enabled
   sbk burned: false
   dk burned: false
```

```
    boot device: emmc
    operating mode: 3
    device config strap: 0
    device config fuse: 0
    sdram config strap: 0

sending file: flash.bct
- 4080/4080 bytes sent
flash.bct sent successfully
odm data: 0x30000011
downloading bootloader -- load address: 0x108000 entry point: 0x108000
sending file: bootloader.bin
| 933404/933404 bytes sent
bootloader.bin sent successfully
waiting for bootloader to initialize
```

And then blocks for an extended period of time (more that ~20 seconds), try rebooting the device into recovery mode again and retry the flashing procedure. If this does not succeed, it is possible that you may have a development kit that is not supported by the publicly-available OS images. Please contact TegraDev@nvidia.com for a new OS image for your device.

## Driver Will Not Install

If the driver will not install or produces a "Code 10" error, try manually forcing a removal of the driver:

1) Locate and download the "usbdeview.exe" utility (note that there is a 64 bit version for Windows Vista 64 bit and Windows 7 64 bit) -- we generally recommend using http://download.cnet.com or other "safe" download sites.

2) Install and run usbdeview.exe (for Windows Vista and Windows 7 (32 and 64 bit) you need to right-click on usbdeview.exe and choose "Run as Administrator...").

3) From the list of devices, select all devices named "NVIDIA USB Boot-recovery driver for Mobile devices", and click the Uninstall button in the toolbar.

4) Unplug the USB cable from the Tegra development kit and plug it back in.

5) Place the device in recovery mode again.

6) Follow the Hardware Wizard and reinstall the USB driver (see next section).

Next, follow the *Manually Install the Driver* section that follows.

## Manually Install the Driver

Connect the device and place it in recovery mode again. When the device wizard shows up:

1) "Can Windows connect to Windows Update..."

   a.   Windows does not need to connect to Windows Update

   b.   [ Click "Next" ]

2) "What do you want the wizard to do?"

   a.   "Install from a list or specific location (Advanced)"

   b.   [ Click "Next" ]

3) Select "Don't Search. I will choose the driver to install."

   a.   [ Click "Next" ]

4) [ Click "Have Disk..." ]

5) Browse to where your modified .inf file is, and pick that .inf file specifically.  For example, browse to:

   `C:\Program Files\NVIDIA Corporation\tegra_froyo_20101105\usbpcdriver`

   and select:

   `NvidiaUsb.inf`

   | | |
   |---|---|
   | **Note:** | You must select the "Don't Search" and "Have Disk…" options otherwise Windows will likely select an older driver that is already installed even though it also sees the updated driver you are pointing it at. |

# Appendix: Troubleshooting the ADB Connection

If you cannot make an ADB connection, check the following troubleshooting recommendations.

## Signs that Windows is using the wrong driver

- The device is listed under "Universal Serial Bus controllers" rather than under "Android Phone" or "ADB Interface".
- The Device Manager lists the device name as "Android Debug Bridge Interface" instead of "Android Composite ADB Interface" or "Android ADB Interface".

## Driver Will Not Install

If the driver will not install or produces a "Code 10" error, try manually forcing a removal of the driver:

1) Locate and download the "usbdeview.exe" utility (note that there is a 64 bit version for Windows Vista 64 bit and Windows 7 64 bit) -- we generally recommend using http://download.cnet.com or other "safe" download sites.

2) Install and run usbdeview.exe (for Windows Vista and Windows 7 (32 and 64 bit) you need to right-click on usbdeview.exe and choose "Run as Administrator...").

3) From the list of devices, select all devices named "Android Device" and "NVIDIA Harmony", and click the Uninstall button in the toolbar.

4) Unplug the USB cable from the Tegra development kit and plug it back in.

5) Follow the Hardware Wizard and reinstall the USB driver (see next section).

Next, follow the *Manually Install the Driver* section that follows.

## Manually Install the Driver

Connect the device and place it in recovery mode again. When the device wizard shows up:

1) "Can Windows connect to Windows Update..."

   a. Windows does not need to connect to Windows Update

   b. [ Click "Next" ]

2) "What do you want the wizard to do?"

   a. "Install from a list or specific location (Advanced)"

   b. [ Click "Next" ]

3) Select "Don't Search. I will choose the driver to install."

   a.　[ Click "Next" ]

4) [ Click "Have Disk..." ]

5) Browse to where your modified .inf file is, and pick that .inf file specifically.  For example, browse to:

   `C:\Program Files\NVIDIA Corporation\tegra_froyo_20101105\usbpcdriver`

   and select:

   `NvidiaUsb.inf`

   > **Note:**　You must select the "Don't Search" and "Have Disk…" options otherwise Windows will likely select an older driver that is already installed even though it also sees the updated driver you are pointing it at.

## Remove NVIDIA Recovery Mode Driver

If problems persist, try deleting the NVIDIA Recovery Mode driver from your hard drive (e.g. the files in `C:\Program Files\NVIDIA Corporation\android_tegra_250\usbpcdriver\`) before reinstalling the Android Device USB driver.  Some developers have reported that this solves the problem with adb.

NVIDIA Corporation
2701 San Tomas Expressway
Santa Clara, CA 95050
www.nvidia.com