NVIDIA DEBUG MANAGER FOR ANDROID NDK - VERSION 9.0.0

DA-05628-001_v03 | February 2011

Installation Guide

DOCUMENT CHANGE HISTORY

DA-05628-001_v03

Version	Date	Authors	Description of Change
01			Initial release
02	01/19/2011	Sebastien	Updated ADT for 8.0.1
03	02/15/2011	Ryan	Updated ADT for 9.0.0
04	02/17/2011	Ryan	Updated Getting Started…

TABLE OF CONTENTS

SOFTWARE REQUIREMENTS

SOFTWARE COMPONENTS

The NVIDIA Debug Manager for Android NDK Eclipse plugin expects very specific components to be installed – in order to operate as expected. Please make sure that your Android application development environment is conformant with the following components versions:

- Android SDK r09(or r08)
- Android NDK r4b, Android NDK r5b
- Eclipse Classic 3.6.1 (Helios)
- Eclipse C/C++ Development Tools 7.0.1
- Android Development Tools (ADT) 9.0.0(or 8.0.1)

PRE-INSTALLATION REQUIREMENTS

Before you install the NVIDIA Debug Manager for Android NDK Eclipse plugin, please make sure that the following components and environment variables are properly installed and configured:

- Install the Android SDK r09 and the Android NDK r4b or NDK r5b. See http://developer.android.com/sdk/installing.html and http://developer.android.com/sdk/ndk/index.html for installation instructions.
- Download SDK and NDK from http://dl.google.com/android/android-sdk_r09-windows.zip and http://dl.google.com/android/ndk/android-ndk-r5b-windows.zip (or http://dl.google.com/android/ndk/android-ndk-r4b-windows.zip)
- The NDKROOT environment variable should be set to the location of the Android NDK.

- You should have a Tegra 2 board connected and available to adb. This can be verified by running two commands:

`adb devices` – this command should output at least one available device.
`adb shell ps` – this command should output the list of processes running on the Tegra 2 board.

To run adb your PATH environment variable should contain the tools subdirectory from the Android SDK.

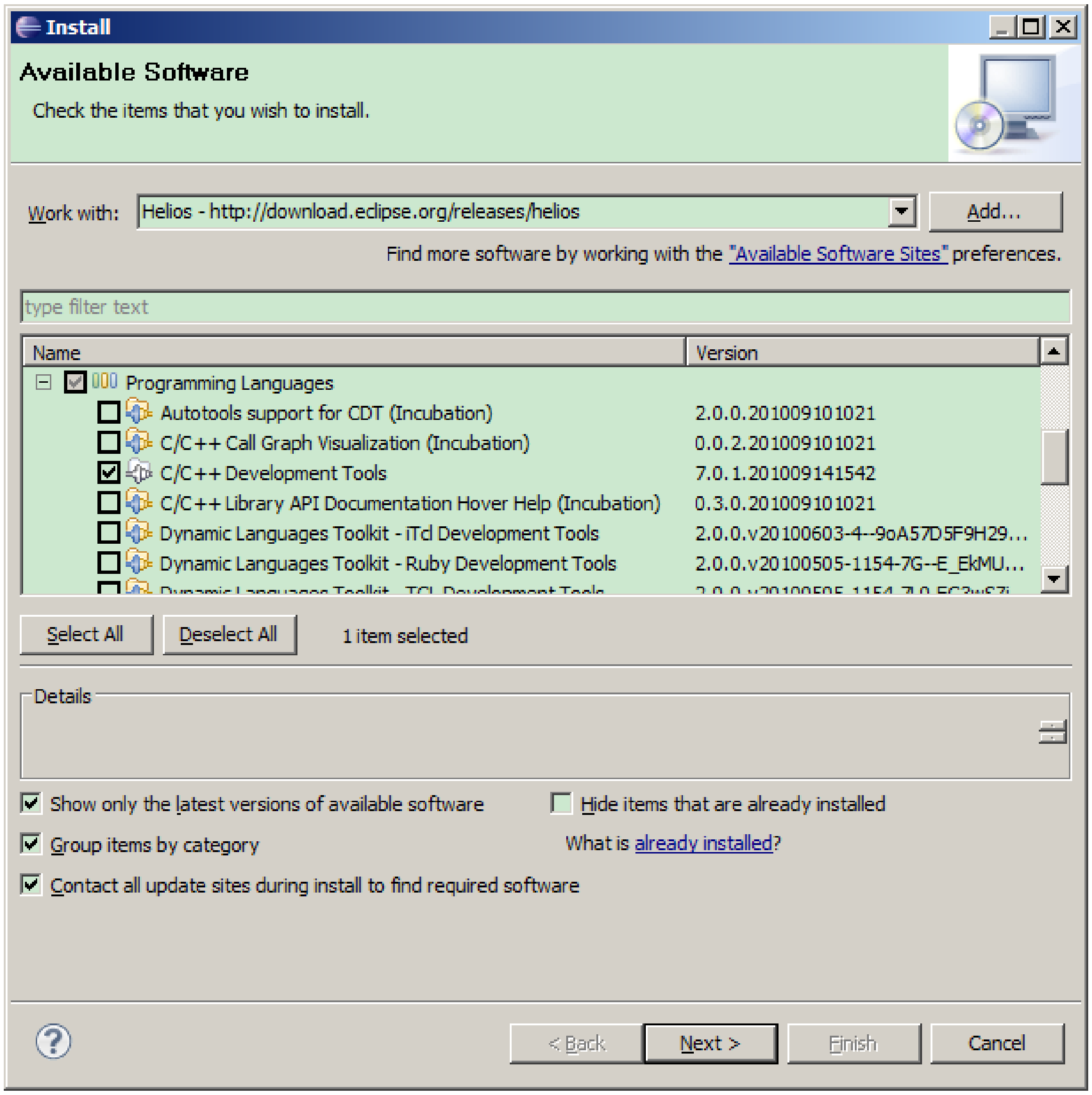
INSTALLATION

INSTALLING ECLIPSE SDK

1. Download Eclipse Classic 3.6.1 (Helios) from http://eclipse.org/downloads/
2. Unpack the downloaded archive by 7zip or any other compress tool.
3. Start Eclipse, for instance by double-clicking eclipse.exe in the “eclipse” directory which is unpacked above.
4. Select a workspace location.
5. Verify the Eclipse version number by opening Help / About Eclipse SDK. Close the dialog.

INSTALLING C/C++ DEVELOPMENT TOOLS

6. Open Help / Install New Software. This shows the “Install” dialog.
7. In the “Work with” dropdown list choose “Helios – http://download.eclipse.org/releases/helios” and wait while the list of available software is loaded.
8. From the list of available software select “Programming Languages” / “C/C++ Development Tools 7.0.1”.


Install

Available Software

Check the items that you wish to install.

Work with: Helios - http://download.eclipse.org/releases/helios

Find more software by working with the "Available Software Sites" preferences.

type filter text

Name	Version
[✓] Programming Languages	
[ ] Autotools support for CDT (Incubation)	2.0.0.201009101021
[ ] C/C++ Call Graph Visualization (Incubation)	0.0.2.201009101021
[✓] C/C++ Development Tools	7.0.1.201009141542
[ ] C/C++ Library API Documentation Hover Help (Incubation)	0.3.0.201009101021
[ ] Dynamic Languages Toolkit - iTcl Development Tools	2.0.0.v20100603-4--9oA57D5F9H29...
[ ] Dynamic Languages Toolkit - Ruby Development Tools	2.0.0.v20100505-1154-7G--E_EkMU...

1 item selected

Details

[✓] Show only the latest versions of available software
[ ] Hide items that are already installed
[✓] Group items by category
What is already installed?
[✓] Contact all update sites during install to find required software

9. Press the "Next >" button. Then press the "Next >" button again and then accept the license agreement and press the "Finish" button.
10. Wait while the software is installed and then agree to restart Eclipse by pressing "Restart Now".
11. Verify that C/C++ Development Tools 7.0.1 is installed by opening Help / About Eclipse SDK / Installation Details. You should see C/C++ Development Tools 7.0.1 in the list of installed software. Close the "Eclipse SDK Installation Details" dialog and then the "About Eclipse SDK" dialog.

INSTALLING ADT

ADT needs be installed according to http://developer.android.com/sdk/eclipse-adt.html

12. Open Help / Install New Software and press the “Add” button.
13. Type in “ADT” into the “Name” field and “https://dl-ssl.google.com/android/eclipse/” into “Location” and press “Ok”.

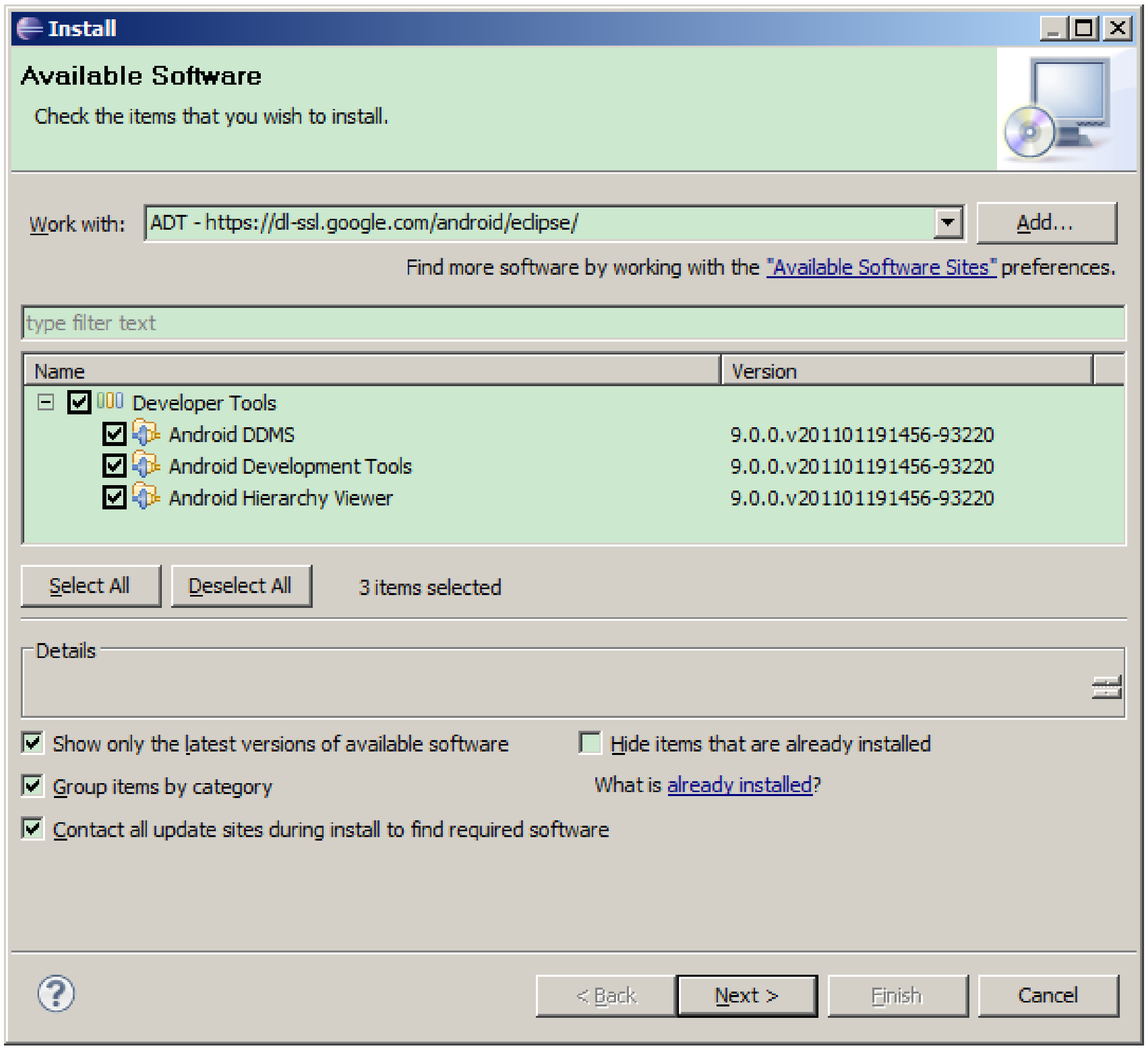
Note: if the version 9.0.0 and 8.0.1 are not available on this site, you can download an archive zip at:

http://dl.google.com/android/ADT-9.0.0.zip or

http://dl.google.com/android/ADT-8.0.1.zip

Press “Archive…” and enter the proper path of ADT-x.0.0.zip when “Add Repository”

14. Open the “Developer Tools” folder in the available software list and select “Android DDMS” and “Android Development Tools”.

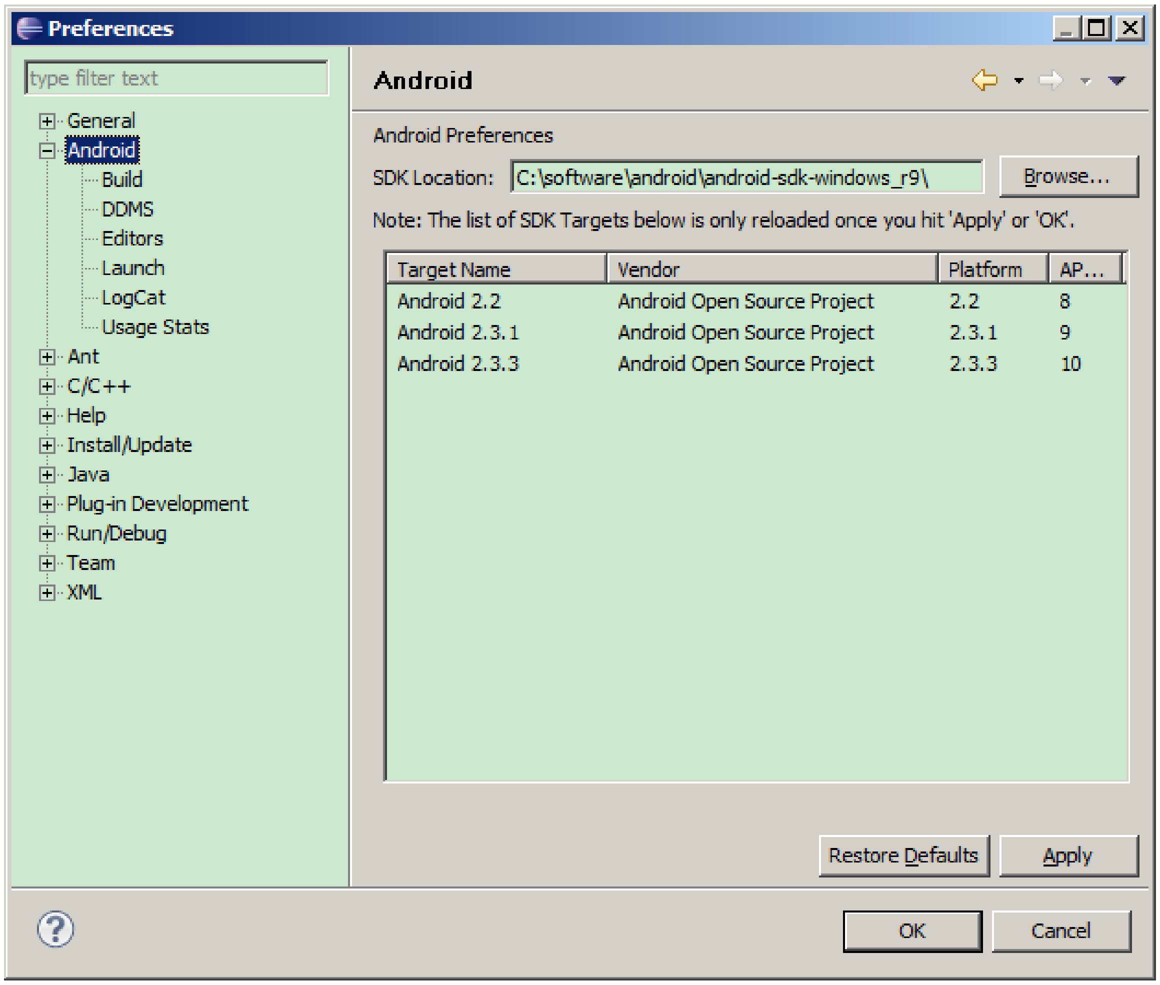

Install
Available Software
Check the items that you wish to install.
Work with:
ADT - https://dl-ssl.google.com/android/eclipse/
Add...
Find more software by working with the "Available Software Sites" preferences.
type filter text

Name	Version
[✓] Developer Tools	
[✓] Android DDMS	9.0.0.v201101191456-93220
[✓] Android Development Tools	9.0.0.v201101191456-93220
[✓] Android Hierarchy Viewer	9.0.0.v201101191456-93220

Select All
Deselect All
3 items selected
Details

- [✓] Show only the latest versions of available software
- [ ] Hide items that are already installed
- What is already installed?
- [✓] Group items by category
- [✓] Contact all update sites during install to find required software

15. Press Next, then Next again, then read and accept End User License Agreement and press the "Finish" button.
16. Press the "Ok" button on the security warning claiming that you are installing unsigned content. Wait for installation to complete, then agree to restart Eclipse by pressing "Restart Now".
17. After Eclipse restarts open "Window" / "Preferences", select "Android" on the left. Enter the correct Android SDK location on the right and press "Ok".



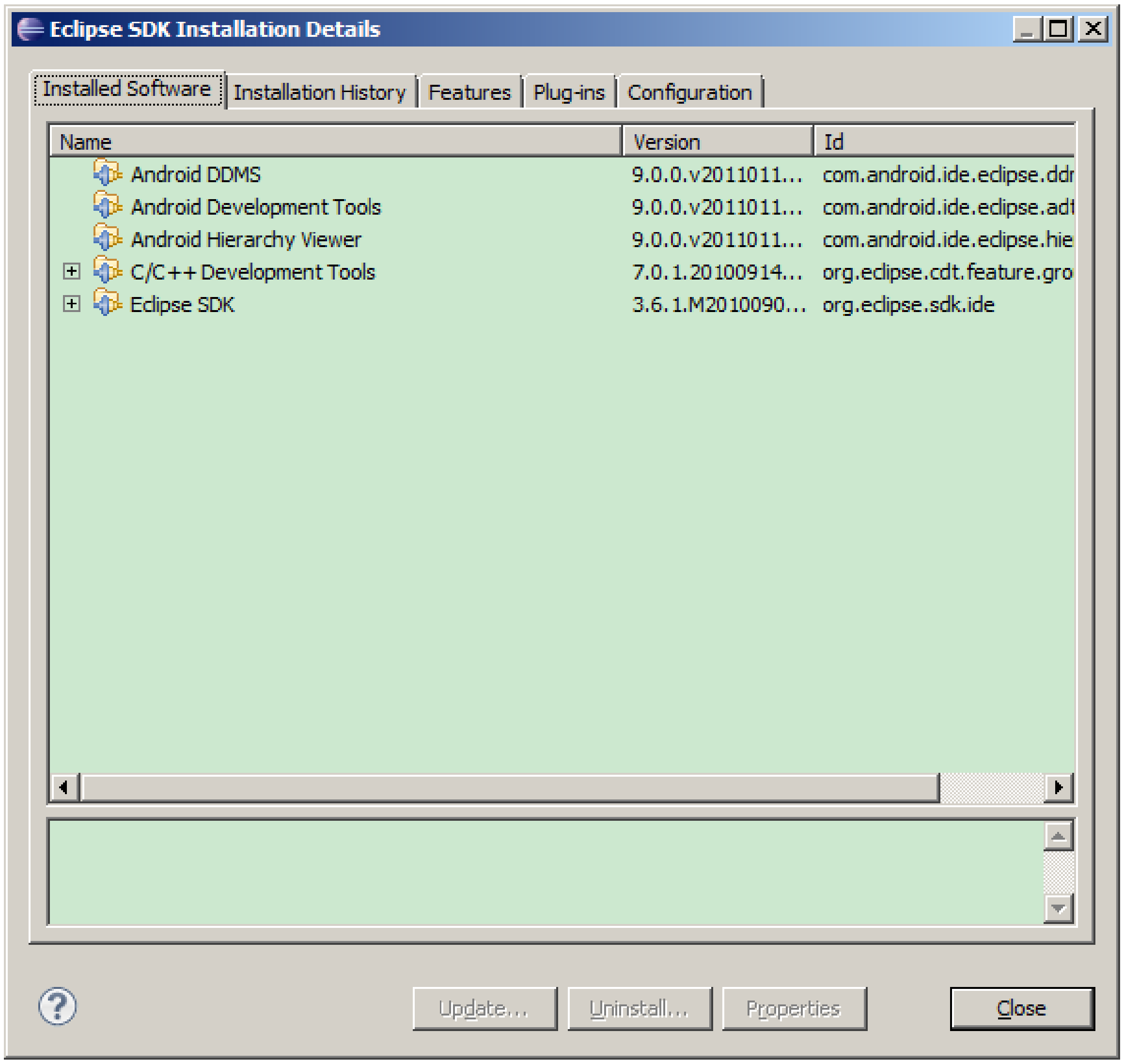
Android Preferences

SDK Location: C:\software\android\android-sdk-windows_r9\

Note: The list of SDK Targets below is only reloaded once you hit 'Apply' or 'OK'.

Target Name	Vendor	Platform	AP...
Android 2.2	Android Open Source Project	2.2	8
Android 2.3.1	Android Open Source Project	2.3.1	9
Android 2.3.3	Android Open Source Project	2.3.3	10

18. Verify that ADT 9.0.0 is installed by opening Help / About Eclipse SDK / Installation Details. You should see “Android DDMS 9.0.0” , “Android Development Tools 9.0.0” and “Android Hierarchy Viewer 9.0.0”in the list of installed software. Close the “Eclipse SDK Installation Details” dialog and then the “About Eclipse SDK” dialog.



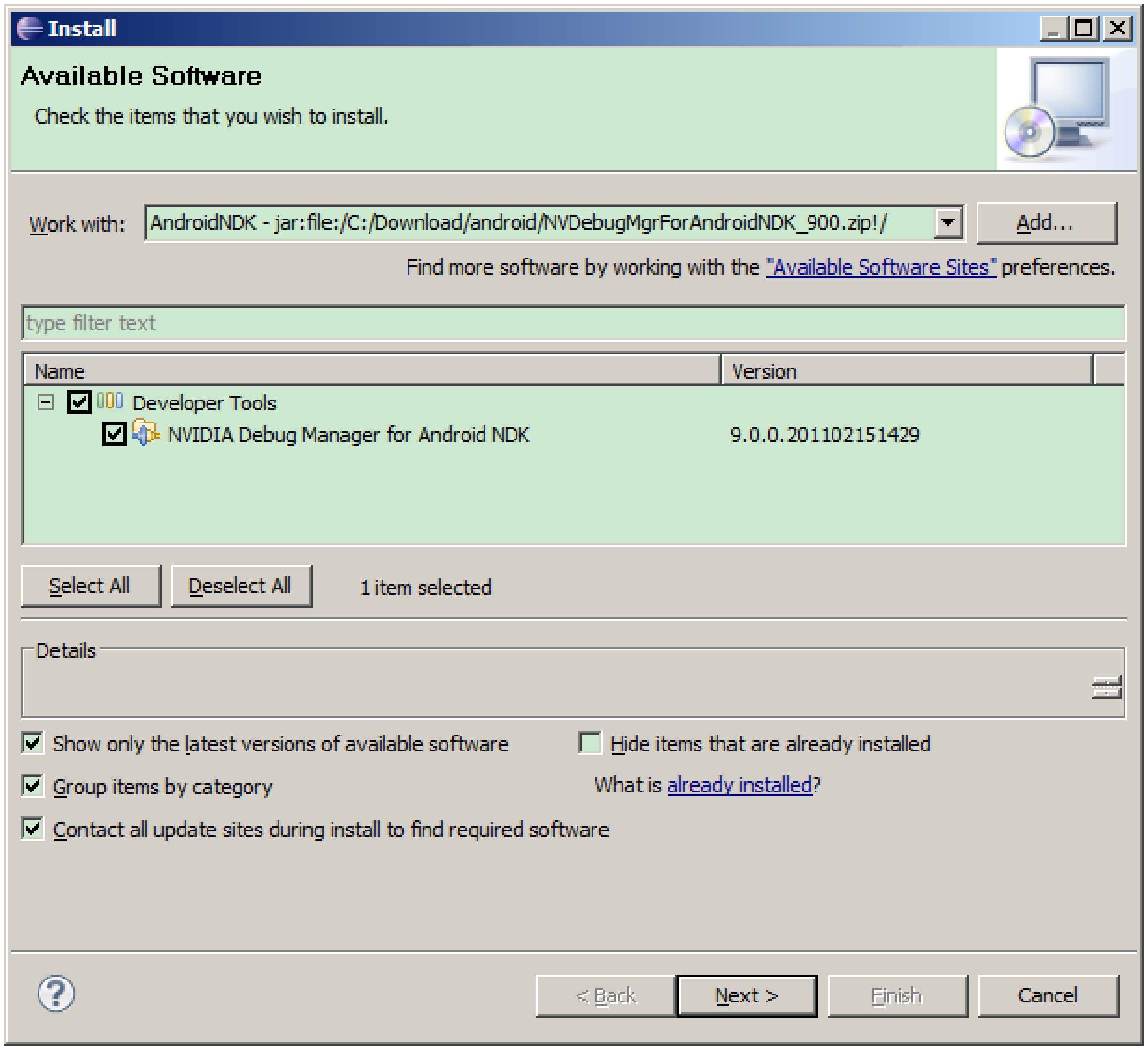
Eclipse SDK Installation Details

Installed Software | Installation History | Features | Plug-ins | Configuration

Name	Version	Id
Android DDMS	9.0.0.v2011011...	com.android.ide.eclipse.ddr
Android Development Tools	9.0.0.v2011011...	com.android.ide.eclipse.adt
Android Hierarchy Viewer	9.0.0.v2011011...	com.android.ide.eclipse.hie
C/C++ Development Tools	7.0.1.20100914...	org.eclipse.cdt.feature.gro
Eclipse SDK	3.6.1.M2010090...	org.eclipse.sdk.ide

INSTALLING NVIDIA DEBUG MANAGER FOR ANDROID NDK ECLIPSE PLUGIN

19. Open Help / Install New Software and press the "Add" button.
20. Press "Archive" and enter the location of the supplied update site archive (zip file). Then press "Ok".
21. Open the "Developer Tools" folder in the available software list and select "NVIDIA Debug Manager for Android NDK".


Install

Available Software

Check the items that you wish to install.

Work with: AndroidNDK - jar:file:/C:/Download/android/NVDebugMgrForAndroidNDK_900.zip!/

Find more software by working with the "Available Software Sites" preferences.

Name	Version
☑ Developer Tools	
☑ NVIDIA Debug Manager for Android NDK	9.0.0.201102151429

1 item selected

Details

☑ Show only the latest versions of available software

☐ Hide items that are already installed

☑ Group items by category

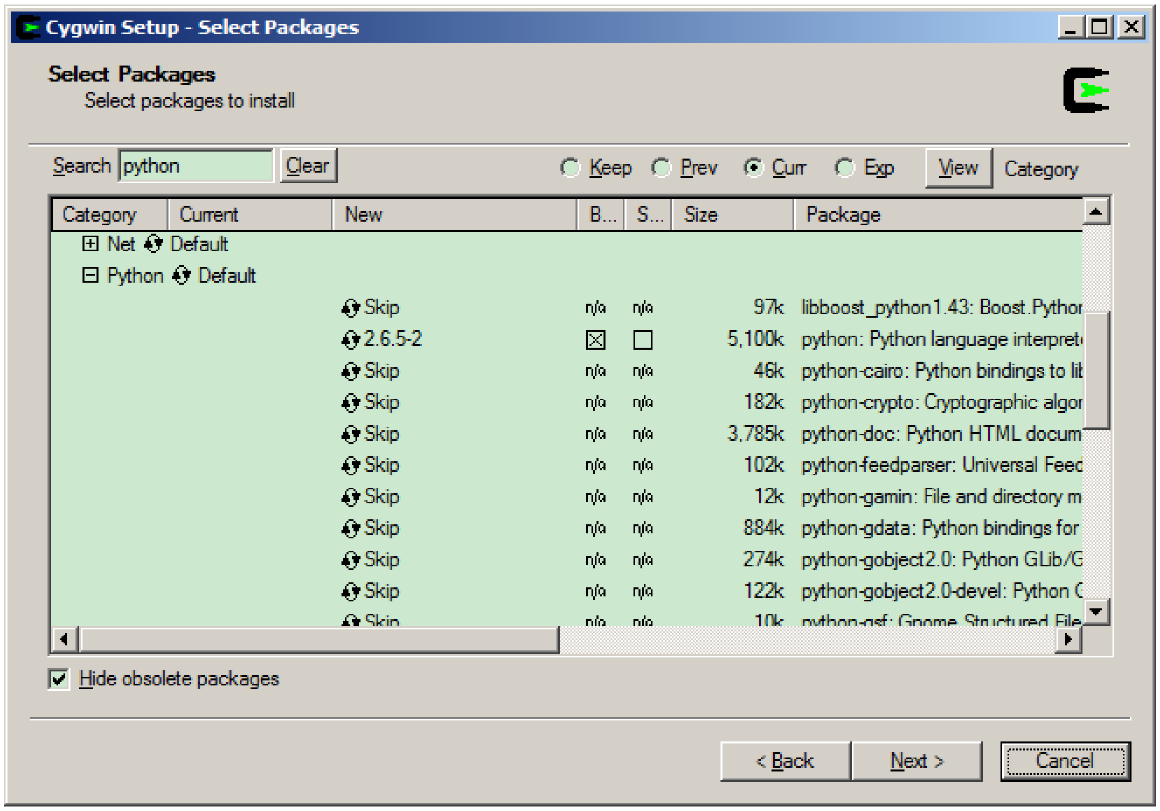
What is already installed?

☑ Contact all update sites during install to find required software

22. Press Next, then Next again, then read and accept the End User License Agreement and press “Finish”.
23. Press “Ok” button on the security warning claiming that you are installing unsigned content. Wait for installation to complete, then agree to restart Eclipse by pressing “Restart Now”.

INSTALLING CYGWIN

24. Install Cygwin and make sure Python 2.6 and Ncurses are selected to be installed.



Cygwin Setup - Select Packages

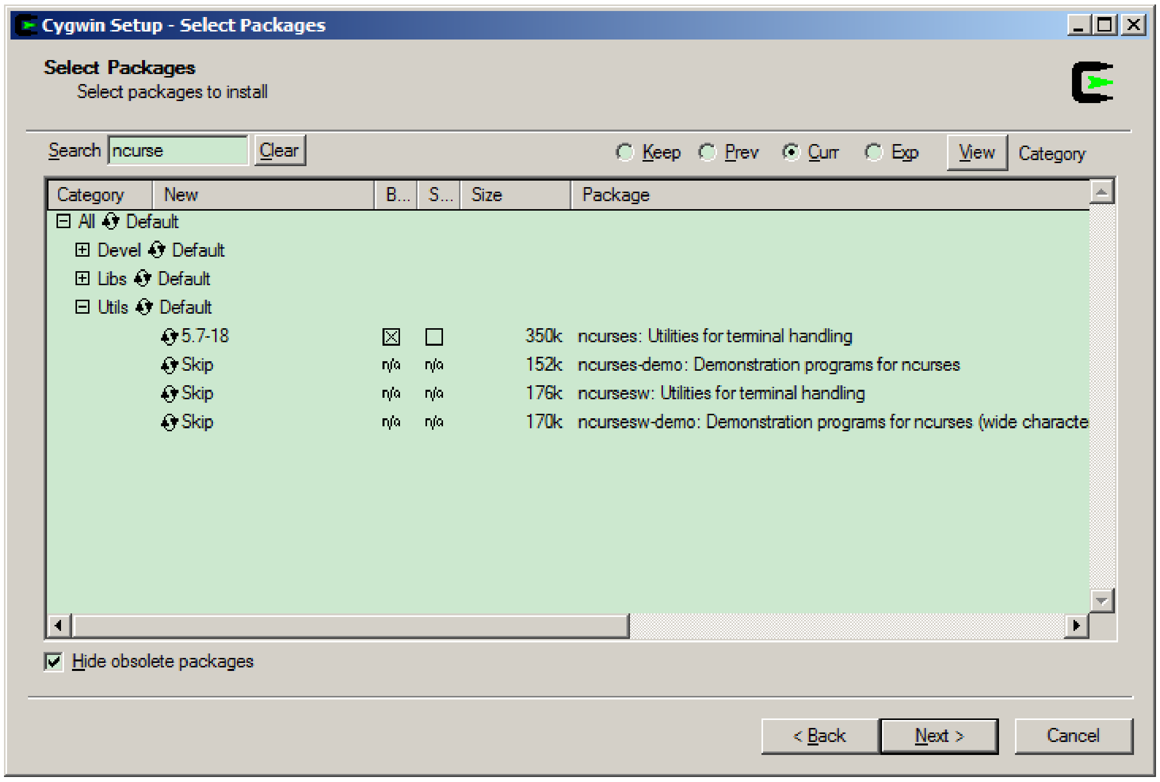
Select Packages
Select packages to install

Search python Clear ( ) Keep ( ) Prev (●) Curr ( ) Exp View Category

Category	Current	New	B...	S...	Size	Package
⊞ Net Default						
⊟ Python Default						
		Skip	n/a	n/a	97k	libboost_python1.43: Boost.Pythor
		2.6.5-2	☒	☐	5,100k	python: Python language interpret
		Skip	n/a	n/a	46k	python-cairo: Python bindings to lil
		Skip	n/a	n/a	182k	python-crypto: Cryptographic algor
		Skip	n/a	n/a	3,785k	python-doc: Python HTML docum
		Skip	n/a	n/a	102k	python-feedparser: Universal Feed
		Skip	n/a	n/a	12k	python-gamin: File and directory m
		Skip	n/a	n/a	884k	python-gdata: Python bindings for
		Skip	n/a	n/a	274k	python-gobject2.0: Python GLib/G
		Skip	n/a	n/a	122k	python-gobject2.0-devel: Python (
		Skip	n/a	n/a	10k	python-gsf: Gnome Structured File

☑ Hide obsolete packages

< Back Next > Cancel



Cygwin Setup - Select Packages

Select Packages
Select packages to install

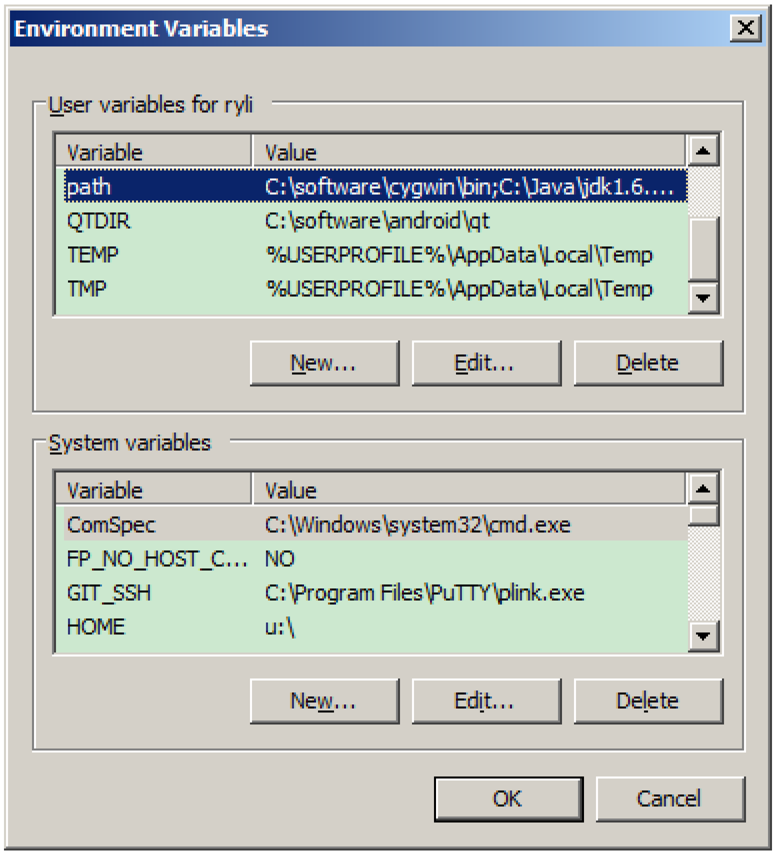
Search ncurse Clear ( ) Keep ( ) Prev (●) Curr ( ) Exp View Category

Category	New	B...	S...	Size	Package
⊟ All Default					
⊞ Devel Default					
⊞ Libs Default					
⊟ Utils Default					
	5.7-18	☒	☐	350k	ncurses: Utilities for terminal handling
	Skip	n/a	n/a	152k	ncurses-demo: Demonstration programs for ncurses
	Skip	n/a	n/a	176k	ncursesw: Utilities for terminal handling
	Skip	n/a	n/a	170k	ncursesw-demo: Demonstration programs for ncurses (wide characte

☑ Hide obsolete packages

< Back Next > Cancel

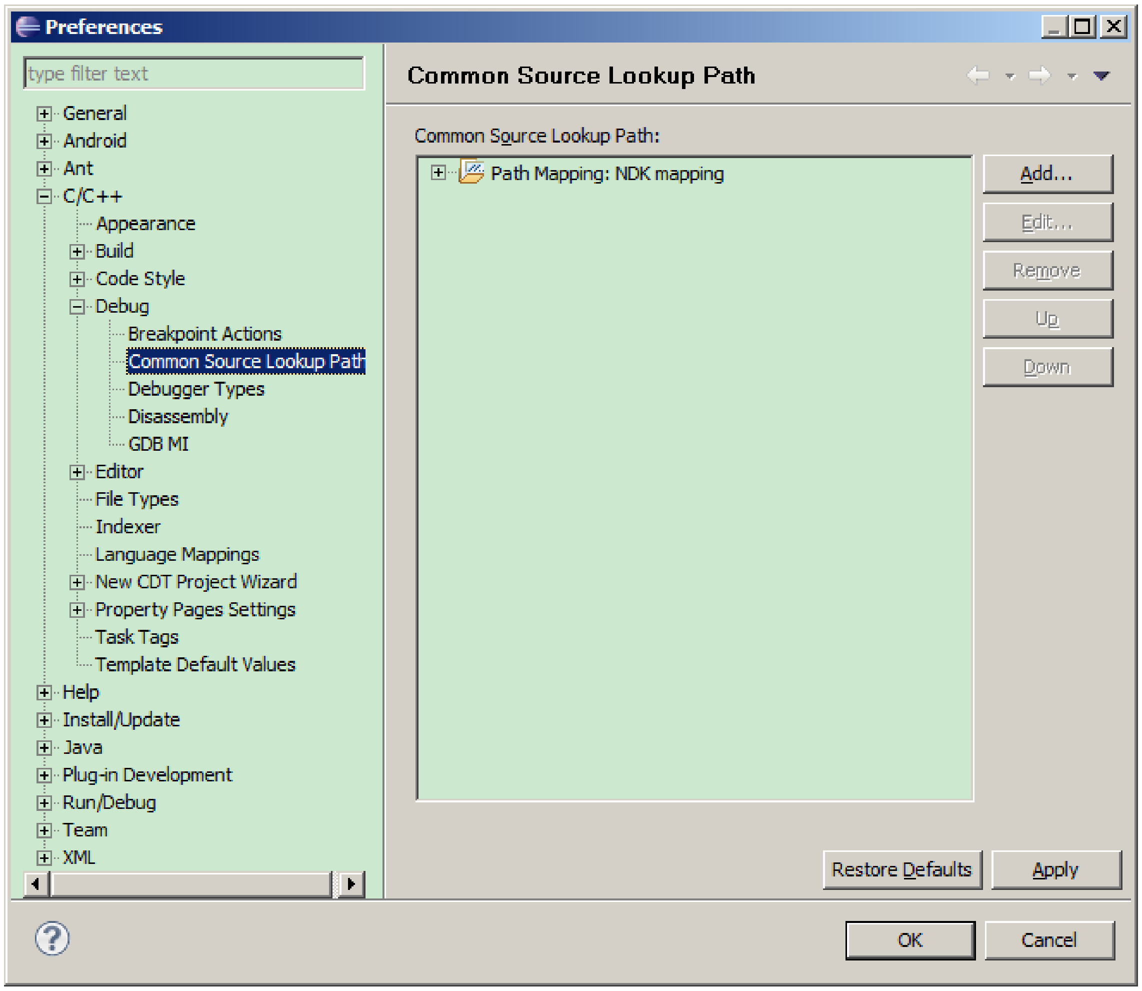
25. Add $CYGWIN_HOME/bin to your PATH environment variable.



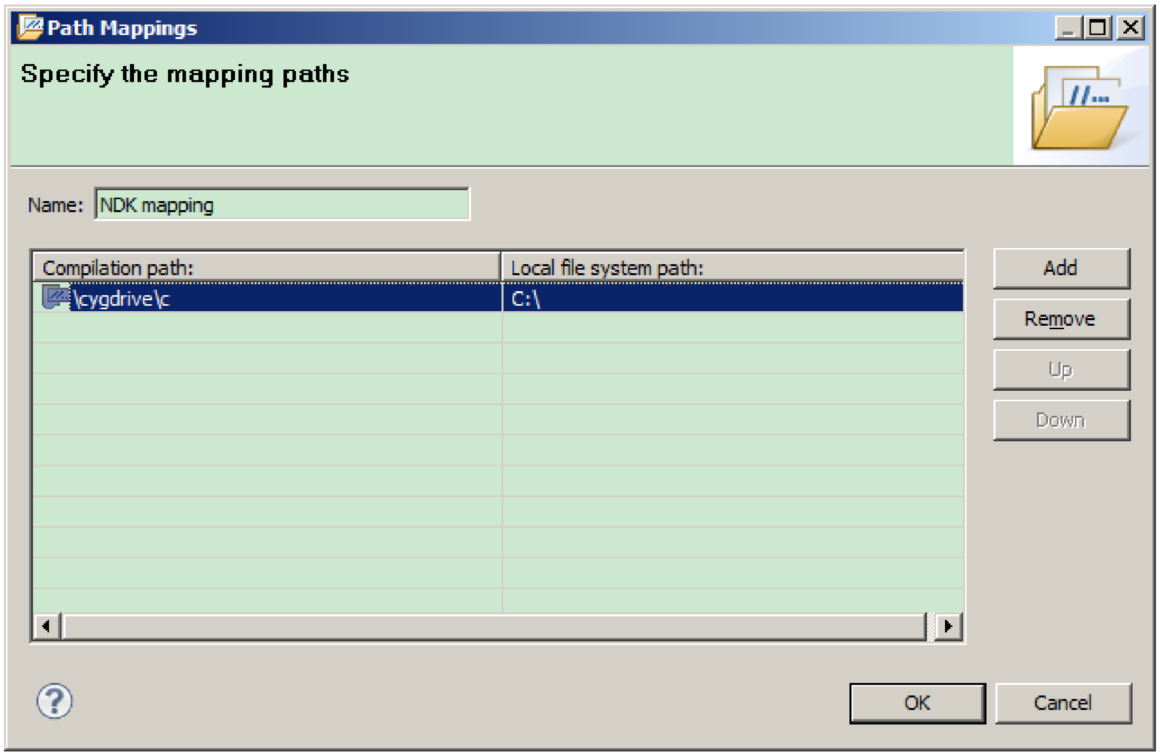
Variable	Value
path	C:\software\cygwin\bin;C:\Java\jdk1.6....
QTDIR	C:\software\android\qt
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

Variable	Value
ComSpec	C:\Windows\system32\cmd.exe
FP_NO_HOST_C...	NO
GIT_SSH	C:\Program Files\PuTTY\plink.exe
HOME	u:\

26. In Eclipse, select the menu item "Window->Preferences", select "C/C++ -> Debug -> Common Source Lookup Path", click "Add" and select "Path Mapping".



Under "Compilation path", type "\cygdrive\c" and under "Local file system path",select "C:". This ensures that gdb and Eclipse are on the same page with regards to source code. If your source code resides on other drives than C:, please add those drives too in a similar manner.



Name: NDK mapping

Compilation path:	Local file system path:
\cygdrive\c	C:\

The installation now is complete.

GETTING STARTED…

DEBUGGING TEGRA PACK SAMPLES

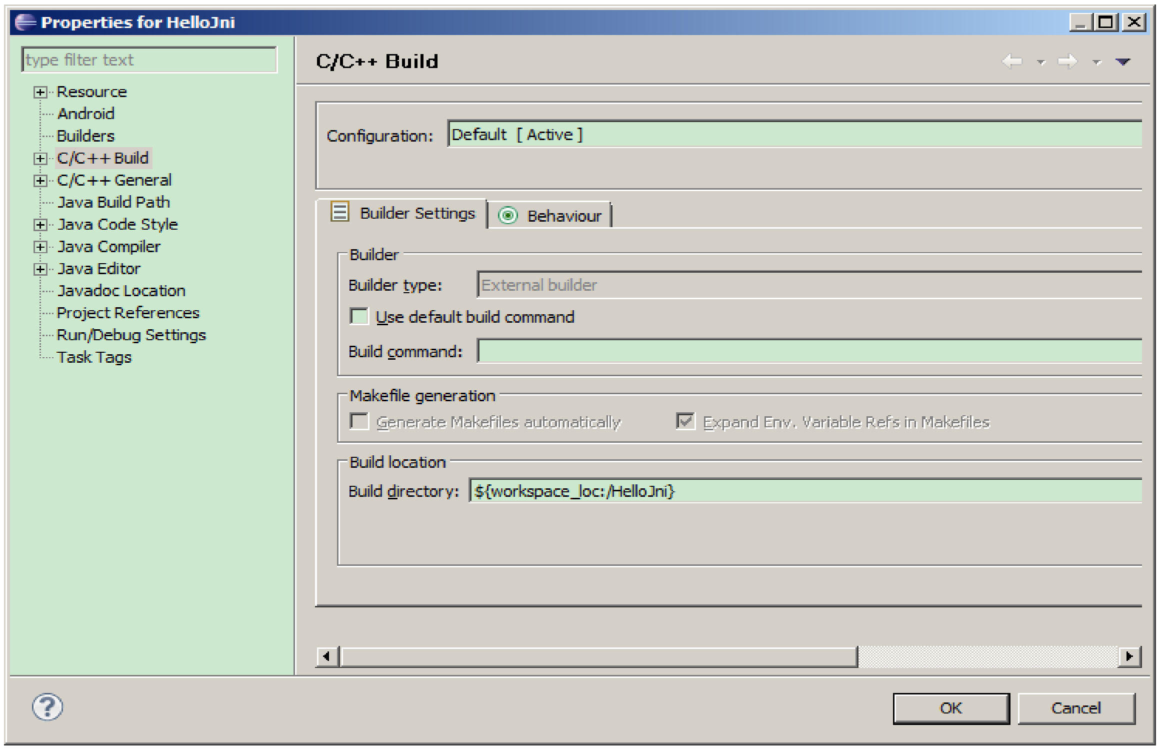
To debug Tegra sample projects, simply import the project into eclipse, build the project, and then right click the project -> Debug as -> Android NDK application, the debug configuration will be properly created and you can start debugging.

DEBUGGING ANDROID NDK SAMPLES

To debug Android NDK sample projects, you need to create Android project from existing source, convert it to C/C++ project (refer "Tegra Android Setup Guide" which can be downloaded from http://developer.nvidia.com/tegra/downloads for details) and then start debugging.

DEBUGGING YOUR OWN PROJECT BUILT OUTSIDE OF ECLIPSE

APK projects built using build systems other than Eclipse (such as those built using command-line ndk-build and Java "ant") must be imported into Eclipse in order to debug them via NVDM. Generally, the process is analogous to that for setting up a debugging project for the Android NDK samples: importing the Android Java/APK project into Eclipse from existing source and then converting to a C/C++ project. Note that Eclipse C/C++ projects include a C/C++ build command. Since your custom-built application does not use Eclipse's build system, you may need to set the C/C++ build command to be custom (and set it to be blank). Failing to do so will cause Eclipse to run "make" on your project directory, which is likely to fail. Finally, NVDM's default search paths for native code symbols may not match your application's custom build tree. This can cause missed breakpoints. In this case, you may need to set the shared library search path manually after you create the debug configuration, before launching your first debugging session.



Configuration: Default [Active]

Builder type: External builder

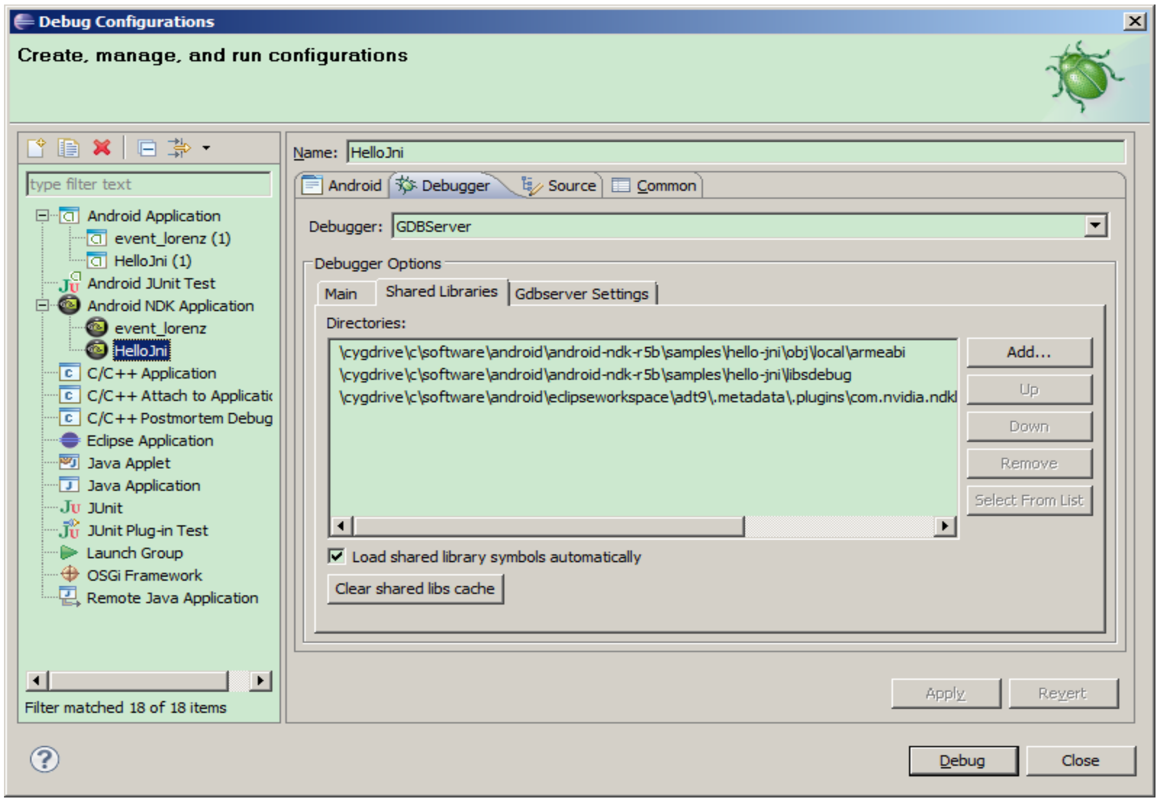
[ ] Use default build command

Build command:

[ ] Generate Makefiles automatically

[✓] Expand Env. Variable Refs in Makefiles

Build directory: ${workspace_loc:/HelloJni}



Create, manage, and run configurations

Filter matched 18 of 18 items

Name: HelloJni

Debugger: GDBServer

Debugger Options

Directories:

```
\cygdrive\c\software\android\android-ndk-r5b\samples\hello-jni\obj\local\armeabi
\cygdrive\c\software\android\android-ndk-r5b\samples\hello-jni\libsdebug
\cygdrive\c\software\android\eclipseworkspace\adt9\.metadata\.plugins\com.nvidia.ndkl
```

☑ Load shared library symbols automatically

TROUBLESHOOTING

LIMITATIONS

Missing Breakpoints

The current version of the plugin may be missing breakpoint in native code executed before GDB is attached. We recommend that the developer wanting to debug native code that could be executed during this transient period add a while(i) with int i would be initialized to a non-zero value, and changing the i to zero once GDB is attached, in order to resume the debugging session and hit the breakpoint.

REPORTING ISSUES

In case of problems working with the NVIDIA Debug Manager for Android NDK plugin the following information might be helpful to provide with a bug report:

- Exact versions of Android SDK and Android NDK.
- Exact versions of installed Eclipse software (From Eclipse it is easy to obtain this information from "Help" / "About Eclipse SDK" / "Installation Details" / "Configuration".
- Eclipse error log which resides in $workspace_location/.metadata/.log, where $workspace_location is the location of your Eclipse workspace.
- Android console output.
- General description what the problem is and how to reproduce the problem.

and post the issue to the following Tegra Android Development support forum with a subject line referring to the Debug Manager plugin:

http://developer.nvidia.com/tegra/forums/tegra-forums/android-development