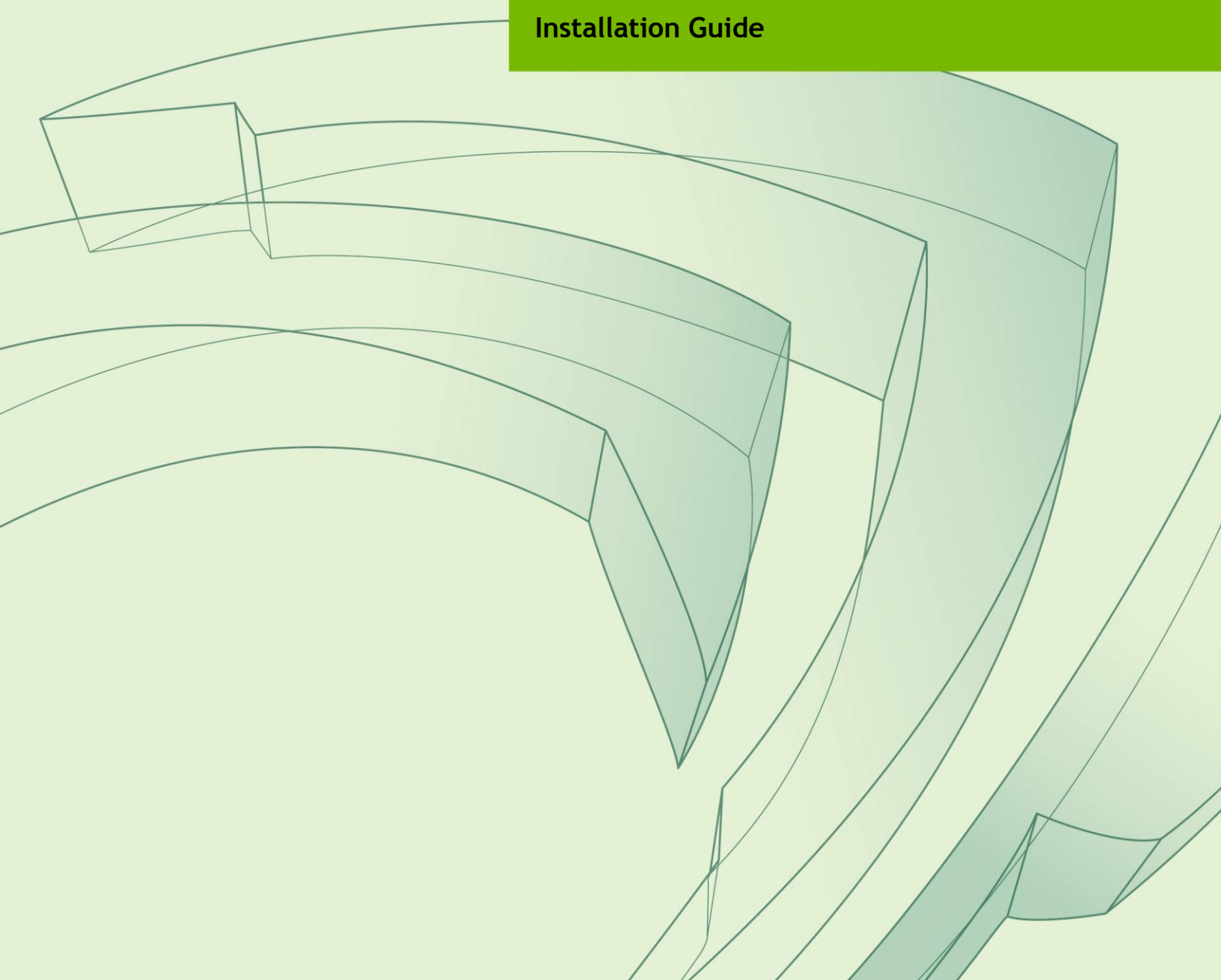




NVIDIA DEBUG MANAGER FOR ANDROID NDK

DA-05628-001_v01 | December 2010

Installation Guide



DOCUMENT CHANGE HISTORY

Document Number

Version	Date	Authors	Description of Change
01			Initial release

TABLE OF CONTENTS

Software Requirements	1
Software Components.....	1
Pre-Installation Requirements	1
Installation	1
Installing Eclipse SDK	1
Installing C/C++ Development Tools.....	1
Installing ADT 0.9.9	3
Installing NVIDIA Debug Manager for Android NDK Eclipse Plugin.....	5
Installing Cygwin.....	6
Troubleshooting	11
Limitations	11
Reporting Issues	11

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 r07
- ▶ Android NDK r4b
- ▶ Eclipse Classic 3.6.1 (Helios)
- ▶ Eclipse C/C++ Development Tools 7.0.1
- ▶ Android Development Tools (ADT) 0.9.9

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 r07 and the Android NDK r4b. 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_r07-windows.zip and <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.

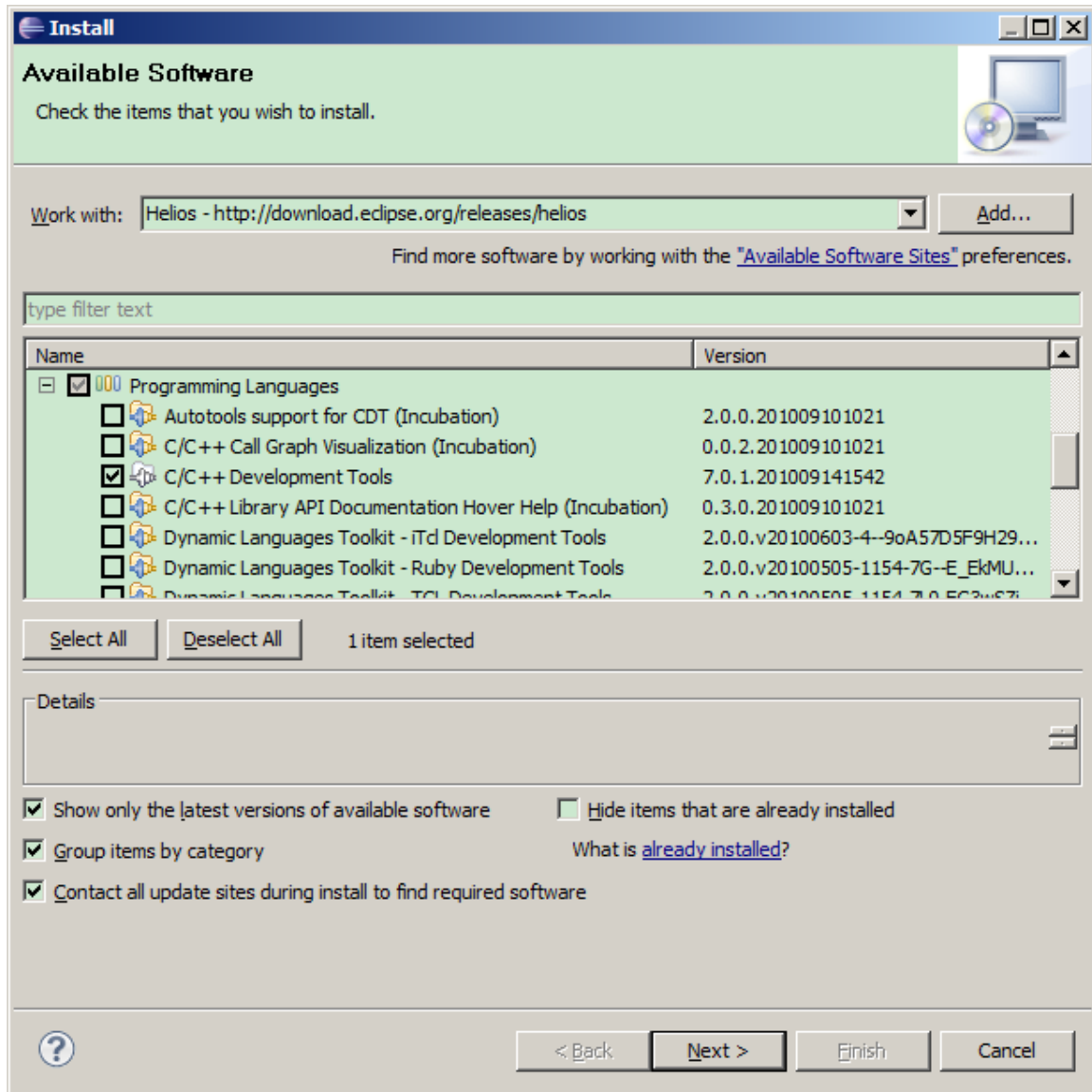
INSTALLATION

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”.



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".

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 0.9.9

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

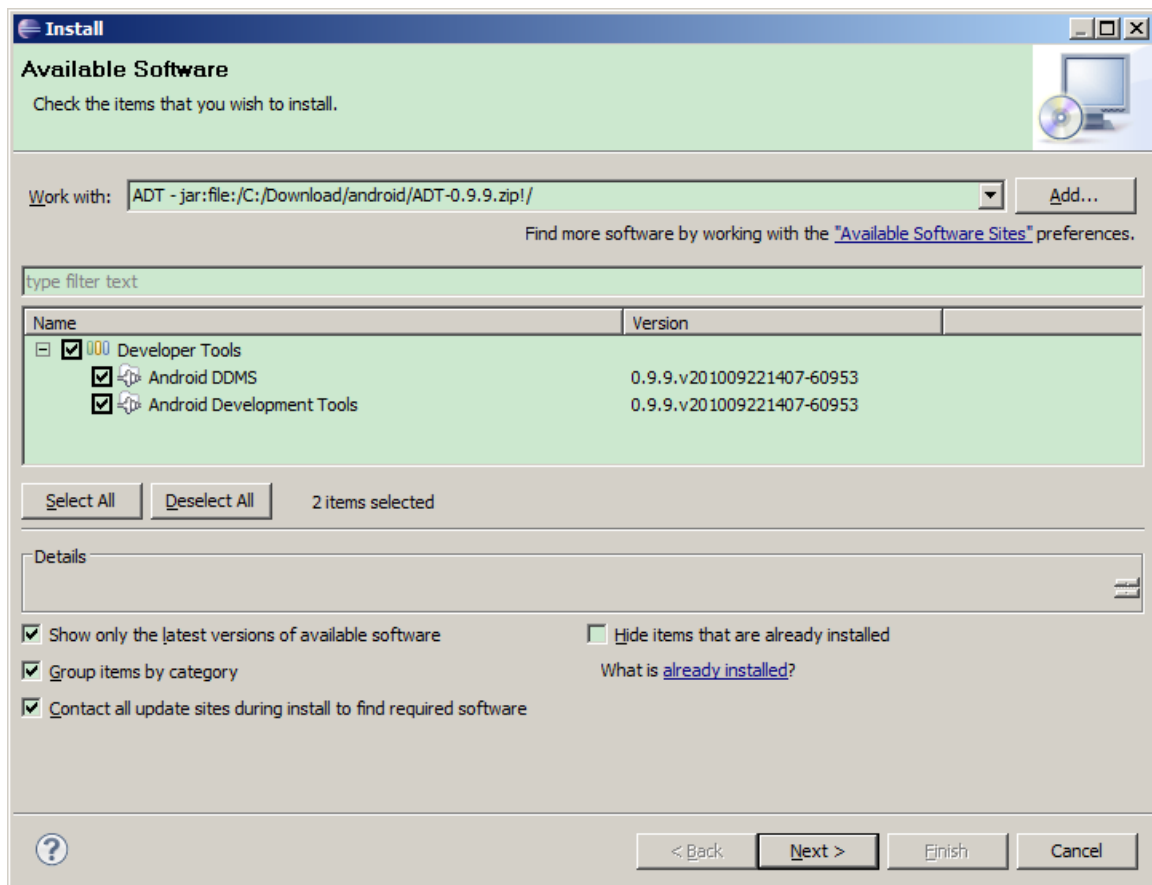
11. Open Help / Install New Software and press the “Add” button.
12. Type in “ADT” into the “Name” field and “https://dl-ssl.google.com/android/eclipse/” into “Location” and press “Ok”.

Note: if the version 0.9.9 is not available on this site, you can download an archive zip at:

<http://dl.google.com/android/ADT-0.9.9.zip>

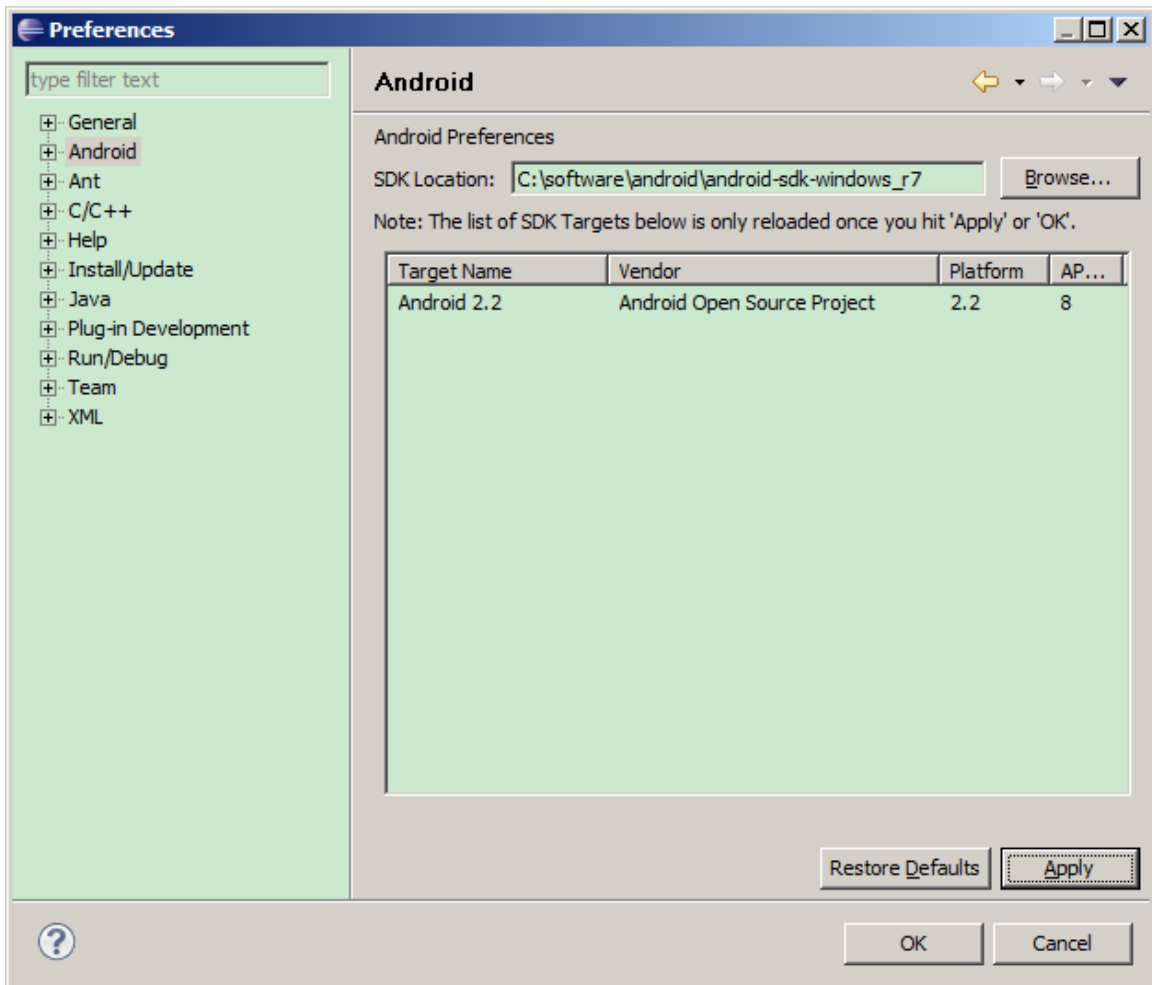
Press “Archive...” and enter the proper path of ADT-0.9.9.zip when “Add Repository”

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

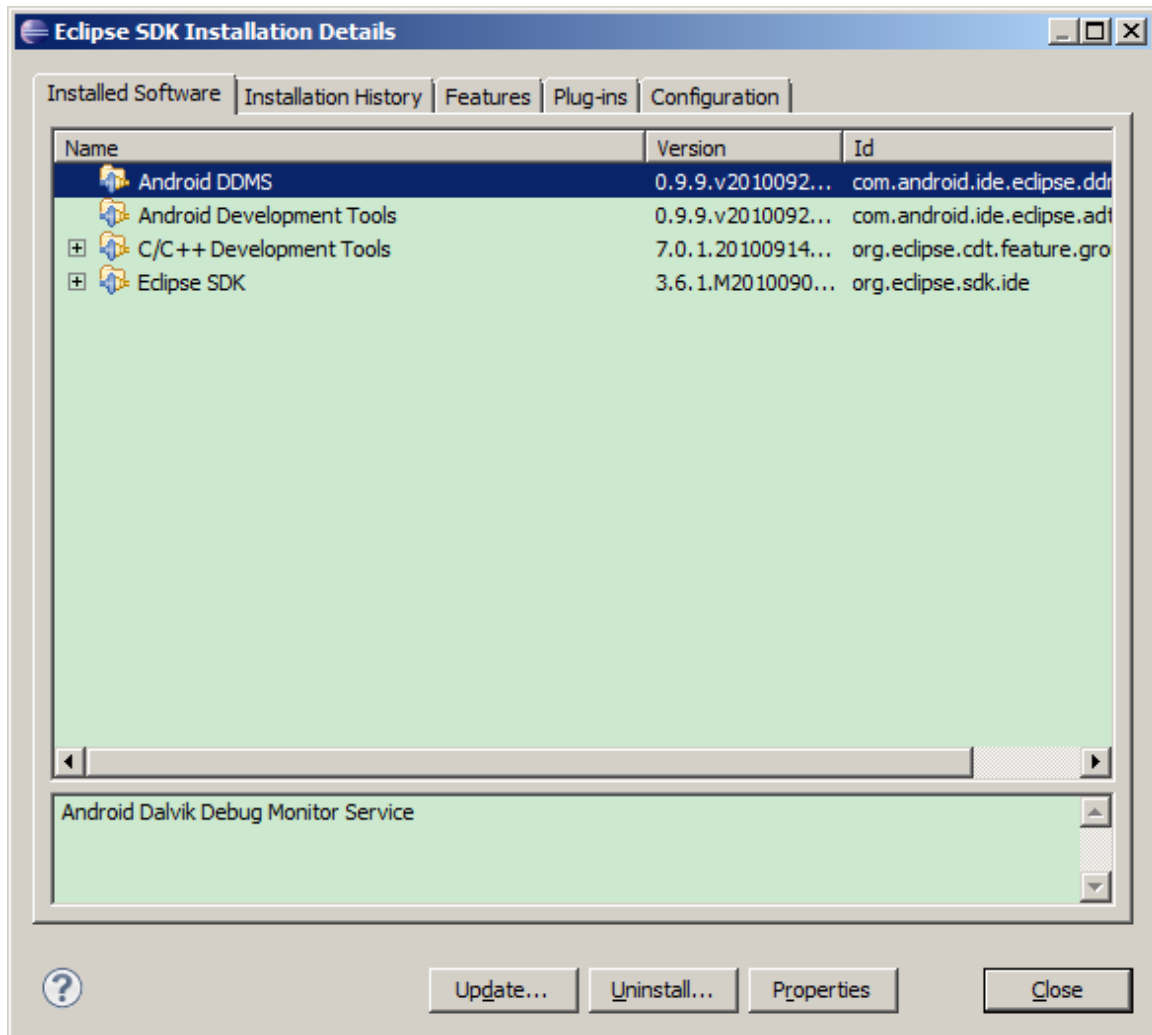


14. Press Next, then Next again, then read and accept End User License Agreement and press the “Finish” button.

15. 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”.
16. After Eclipse restarts open “Window” / “Preferences”, select “Android” on the left. Enter the correct Android SDK location on the right and press “Ok”.

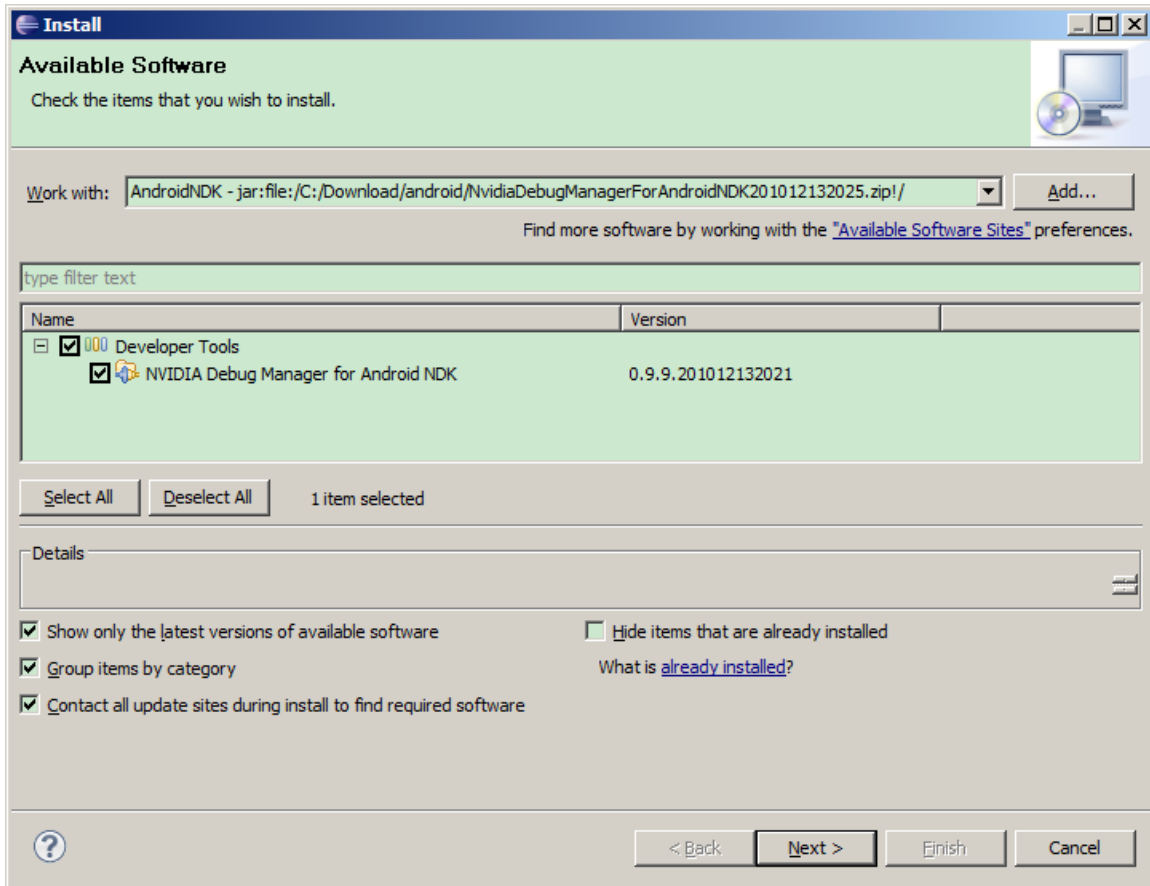


17. Verify that ADT 0.9.9 is installed by opening Help / About Eclipse SDK / Installation Details. You should see “Android DDMS 0.9.9” and “Android Development Tools 0.9.9” in the list of installed software. Close the “Eclipse SDK Installation Details” dialog and then the “About Eclipse SDK” dialog.



INSTALLING NVIDIA DEBUG MANAGER FOR ANDROID NDK ECLIPSE PLUGIN

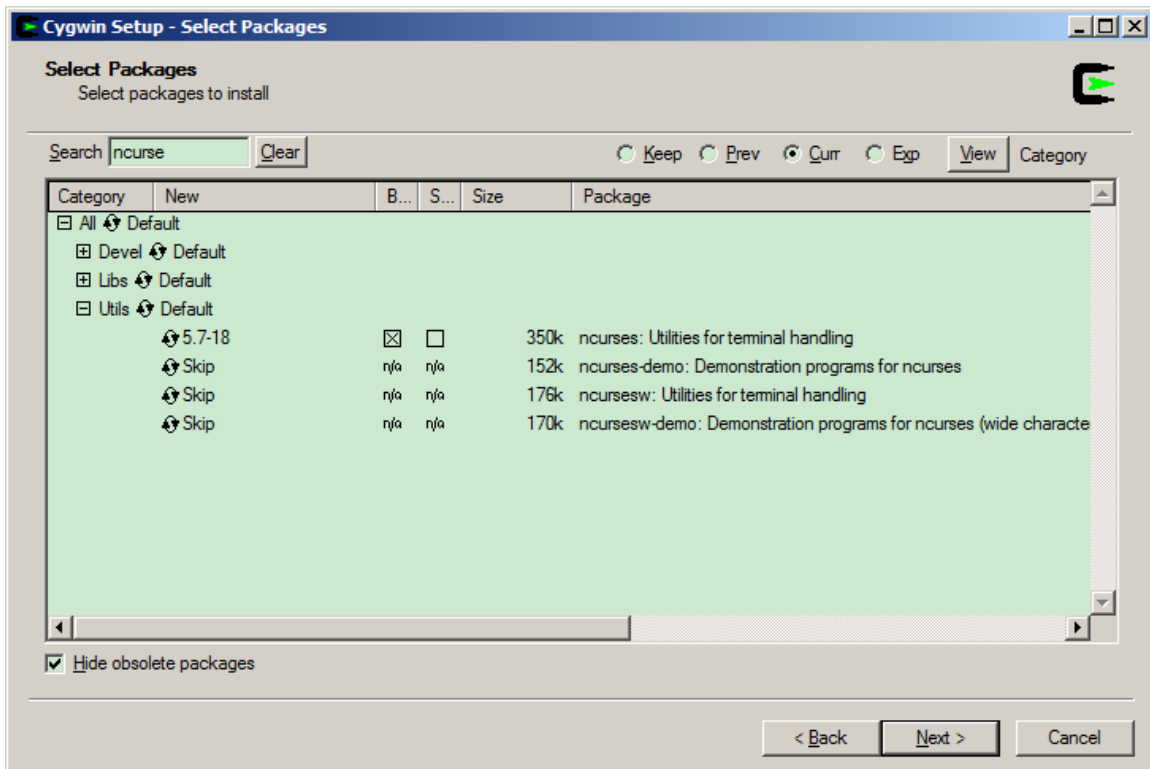
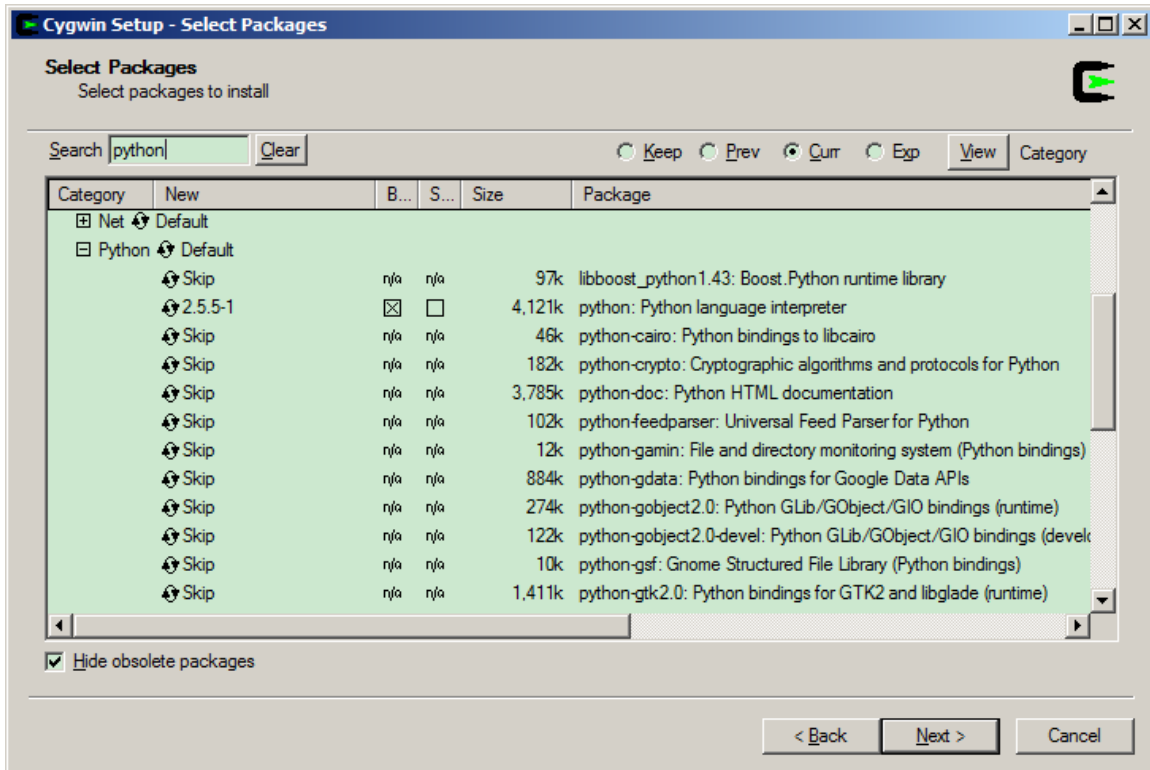
18. Open Help / Install New Software and press the “Add” button.
19. Press “Archive” and enter the location of the supplied update site archive (zip file). Then press “Ok”.
20. Open the “Developer Tools” folder in the available software list and select “NVIDIA Debug Manager for Android NDK”.



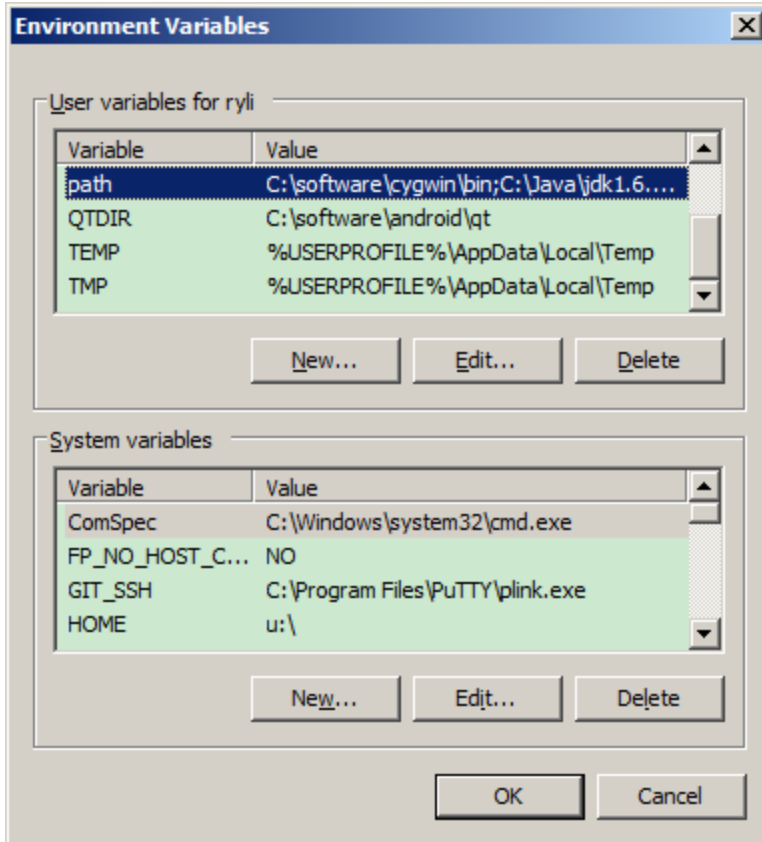
21. Press Next, then Next again, then read and accept the End User License Agreement and press “Finish”.
22. Press “Ok” button on the security warning claiming that you are installing unsigned content. Wait for installation to complete, then agree to restart Eclipse by pressing “Restart Now”.

INSTALLING CYGWIN

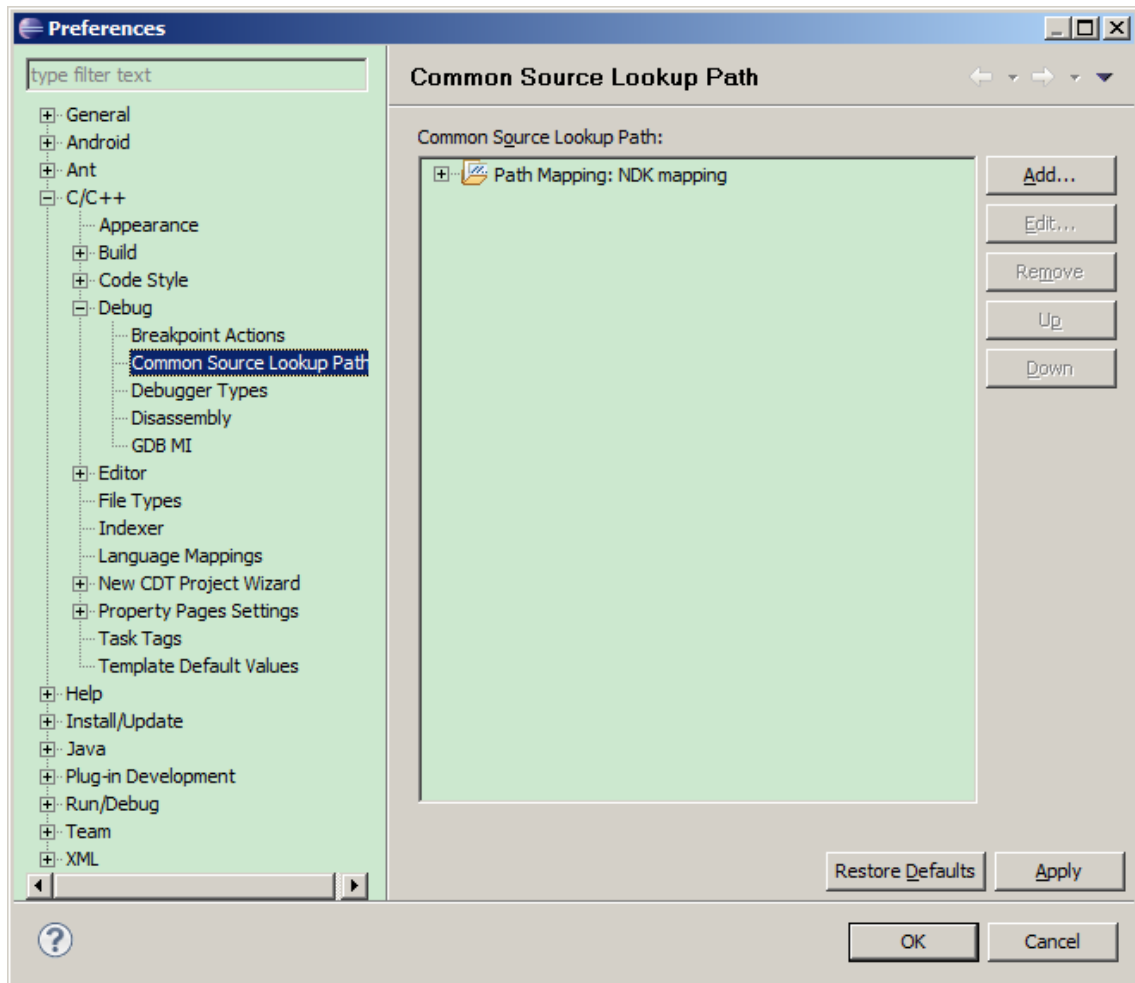
23. Install Cygwin and make sure Python 2.5 and Ncurses are selected to be installed.



24. Add `$CYGWIN_INSTALL_DIR\bin` to your `PATH` environment variable.

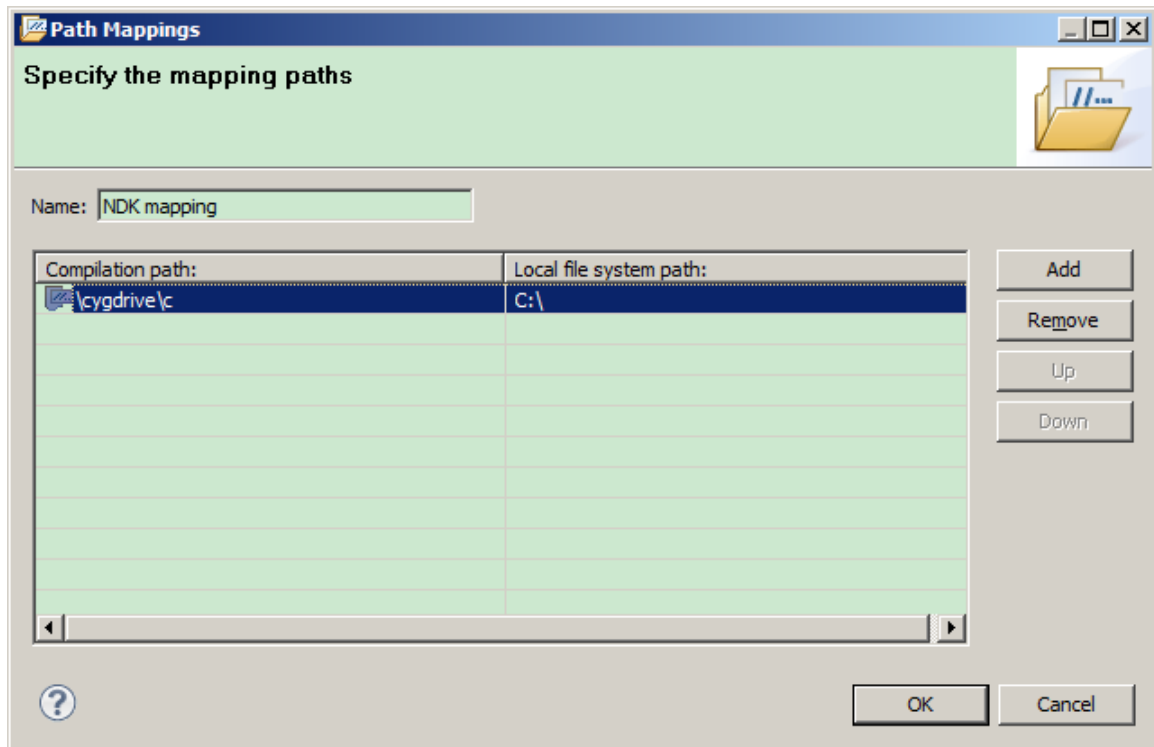


25. In Eclipse, select the menu item "Window->Preferences", select "C/C++ -> Debug -> Common Source Lookup Path", click "Add" and select "Path Mapping".



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.



26. The installation now is complete. To verify installation import event_lorenz from the Tegra sample projects, build the project, right click the project->Debug as->Android NDK application.

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>

Notice

ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE.

Information furnished is believed to be accurate and reliable. However, NVIDIA Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use. No license is granted by implication of otherwise under any patent rights of NVIDIA Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all other information previously supplied. NVIDIA Corporation products are not authorized as critical components in life support devices or systems without express written approval of NVIDIA Corporation.

Trademarks

NVIDIA, the NVIDIA logo, and <add all the other product names listed in this document> are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

© 2010 NVIDIA Corporation. All rights reserved.