Development Host System Setup

From UMG Wiki

Contents

Summary

Owner: bjwood <brian.j.wood@intel.com> (mailto:brian.j.wood@intel.com)

Note: *Newer instructions are provided at one-time setup which cover both Ubuntu and Fedora. If you are moving to Android for the first time, look for the tutorial at UMG Software Development.*

This page explains setting up a Ubuntu 9.10 (Karmic Koala) development host with the proper *.deb packages needed for using Eclipse with the ADT plugin, Installing the Android SDK, and Rebuilding the Android GAID source code.

Note: Preliminary instructions are being added for Fedora 1[0|1|2] as well since many current UMG developers are using this platform from previous projects.

Development Host Setup

Setting up your Ubuntu host for building the Andoid src code (using the software archives)

1. Install the necessary *.deb packages needed by the "build.sh" script.

Note*: It is recommended that you use the "apt-get <package_name>" method from the command line, using the "Applications->Ubuntu Software Center"' (for example) can lead to an incomplete installation of certain packages (we saw an issue with Eclipse from this that was causing errors for installing the ADT plugins*:

Ubuntu 9.10 32bit You'll want to install the following packages for the Ubuntu 9.10 32bit OS:

```
$ sudo apt-get install git-core gnupg flex bison gperf libsdl-dev \
libesd0-dev libwxgtk2.6-dev build-essential \
zip curl libncurses5-dev zlib1g-dev eclipse tofrodos
```

Ubuntu 9.10 64bit You'll want to install the following packages for the Ubuntu 9.10 64bit OS:

```
sudo apt-get install git-core gnupg flex bison gperf build-essential \
zip curl zlib1g-dev gcc-multilib g++-multilib libc6-dev-i386 \
lib32ncurses5-dev ia32-libs x11proto-core-dev libx11-dev \
lib32readline5-dev lib32z-dev eclipse tofrodos
```

Ubuntu 10.04 64bit You'll want to install the following packages for the Ubuntu 10.04 64bit OS:

```
sudo apt-get install git-core gnupg flex bison gperf build-essential \
zip curl zlib1g-dev gcc-multilib g++-multilib libc6-dev-i386 \
lib32ncurses5-dev ia32-libs x11proto-core-dev libx11-dev \
lib32readline5-dev lib32z-dev eclipse tofrodos
```

```
sudo ln -s /usr/bin/fromdos /usr/local/bin/dos2unix
```

***Note**: the package "`sun-java5-jdk`" will not be present in Ubuntu 9.10..."sun-java6-jdk" is what is available, change this package name in the apt-get call above if you want to add in the Ubuntu 9.10 (Karmic Koala) supported Java jdk (we'll be by-passing the usage of this anyway later by modifying the PATH variable.)*

***Note**: An alternative way to get the `sun-java5-jdk` package installed on your Ubuntu 9.10 system is to add the appropriate APT source list entries for Ubuntu 9.04 "Jaunty Jackalope" to your `/etc/apt/sources.list` file:*

```
deb http://us.archive.ubuntu.com/ubuntu/ jaunty multiverse
deb http://us.archive.ubuntu.com/ubuntu/ jaunty-updates multiverse
```

Then run:

```
sudo apt-get update
sudo apt-get install sun-java5-jdk
```

If you are using Ubuntu and install the Java 1.5 JDK using the method above skip to step 6a.

Fedora 12 32 bit Not completely accurate on all the package names needed; rule of thumb, follow Ubuntu but chop off the 'lib' and use 'devel' instead of 'dev' to names. These are the ones that are known to be needed:

```
 $ sudo yum install git-core gnupg flex bison gperf SDL-devel wxGTK-devel zip curl ncurses-devel unix2do
```

'build-essentials', 'libesd0-dev/esd0-devel', 'sun-java5-jdk', and 'zlib1g-dev/zlib1g-devel' do not exist as packages in Fedora. You will need to download Sun's 1.5 Java from the link below and set your JAVA_HOME and PATH to to point to the appropriate java exe's.

2. If you want to do memory debugging, memory leak detection, profiling, etc... you'll want to add the "valgrind" package

```
$ sudo yum install valgrind
```

3. If you didn't install the "sun-java5-jdk" above download the Java SE Development Kit 5.0u22 for Linux x86 (32bit or 64bit) by following this link:

```
https://dct.sun.com/dct/forms/reg_us_0809_958_0.jsp
```

***Note**: Fill out this form on the webpage page and a link will be sent to your email for the download location.*

Select the version (32 or 64 bit) you need for your Ubuntu development host:

- "jdk-1_5_0_22-linux-i586.bin" for 32bit OS
- "jdk-1_5_0_22-linux-amd64.bin" for 64bit OS

4. Change the permissions of this binary to execute:

```
$ chmod +x jdk-1_5_0_22-linux-xxxx.bin
```

5. Run the installer and follow the on-screen prompts:

```
$ ./jdk-1_5_0_22-linux-xxxx.bin
```

A good installation location would be into your `$HOME/bin/` directory.

***Note**: JAVA gotchas:* Android is currently using the 1.5.0 JDK to build src code due to how the "@Override" annotation (used to "override" a method declaration in a superclass) is implemented in the latest version of the 1.6.0 JDK. Also at issue is how Linux distributions (like Ubuntu) use the gnu-java by default and it can be a challange to get the correct JAVA to be used. The "/usr/bin/java" in Ubuntu 9.10 is actually a link to "/etc/alternatives/java", which will be a link to a java other than what you want it to be for our compilation purposes.

6a. If you're installing the Java 1.5.0 JDK using the "sudo apt-get sun-java5-jdk..." method listed in step 1 for Ubuntu then use the following to correct the shell's PATH environment variable so it always puts the "jdk1.5.xxx/bin" before the /usr/bin/* version. (You can edit your .profile (or .bashrc) script to add the following line to the head of your search path, just remove the $ used in example to represent the command line):

```
$ export PATH=/usr/lib/jvm/java-1.5.0-sun/bin:$PATH
example:
$ export PATH="/usr/lib/jvm/java-1.5.0-sun/bin:$PATH"
```

6b. If you're NOT installing the Java 1.5.0 JDK using the "sudo apt-get sun-java5-jdk..." method, instead using the steps in 3->4->5 above, use the following to correct the shell's PATH environment variable to always put the "jdk1.5.xxx/bin" before the /usr/bin/* version. (You can edit your .profile (or .bashrc) script to add the following line to the head of your search path, just remove the $ used in example to represent the command line):

```
$ export PATH=<path-to-JDK-bin-directory>:$PATH
example:
$ export PATH="$HOME/bin/jdk1.5.0_22/bin:$PATH"
```

7. As a final step prior to build the Android source code you need to make sure that the shell used is "Bash" not "Dash" as can be setup on development systems to make them POSIX compliant. Use the following command to reset the shell to Bash:

```
$ sudo dpkg-reconfigure dash  (choose "No" in the ncurses window that pops up)
```

Eclipse Setup with the ADT (Android Development Toolkit) plugin

See the following link for instructions on setting up Eclipse with the ADT plugin: Setting up Eclipse and ADT Plugin (http://umgwiki.intel.com/wiki/?title=Setting_up_Eclipse_and_ADT_Plugin)

Installing the Android SDK

See the following link for instructions on setting up the Android SDK: Setting up Android SDK (http://umgwiki.intel.com/wiki/?title=Setting_up_Android_SDK)

TODO

TODO

TODO