

Installing Cellphone Development Environment

Version : 2.8
 Date : June 14, 2010
 Author : Matt Brenner

1. Here is a list of the required software (all available for free):

Java SE
 NetBeans
 Nokia Ovi Suite
 Symbian Series 40 Emulator
 Symbian Series 60 Emulator
 Samsung Emulators
 LG Emulators

Here's where to download the software:

Java SE	(java.sun.com downloads)
Java SE docs	(java.sun.com downloads)
Nokia Ovi Suite	(www.nokiausa.com and search for "Ovi Suite download") (www.nokiausa.com/get-support-and-software/software/nokia-ovi-suite)
NetBeans (6.8)	(www.netbeans.org , click "Download NetBeans IDE" then download the Java bundle that contains Java ME)
Symbian Emulators	(forum.nokia.com/Resources_and_Information/Tools/Platforms/)
Samsung Emulators	(innovator.samsungmobile.com/down/cnts/toolSDK.detail.view.do?platformId=3&cntsId=5640)
LG Emulators	(developer.lgmobile.com)

I've had many minor, but quite irritating problems getting the NetBeans/Emulator/PC Suite development environment to work. I suspect the problems have to do with the order in which various pieces of the system are installed. Therefore, the exact sequence of steps described below should be carefully followed. Any deviation may produce undesirable results.

Notation

Please note that throughout this document when text is shown inside of square brackets it indicates a selection you should make by left-clicking. So this,

[Next] → [Finish] → [Close]

should be understood to mean: click the button (or tab) marked *Next*, then the one marked *Finished*, then the one marked *Close*. Nothing surprising in this. When two sets of square brackets are used,

[[xxx]]

you must double-left-click. When the square brackets are reversed,

]xxx[

it should be understood to mean that you must right-click.

2. Install *Java SE*:

JDK Note: *it is easy to end up with more than one JDK/JRE installed on a system. Some computers come with a JRE already installed. Some applications (e.g. OpenOffice) will install a JRE. I was careful to remove Java from my system before I began, and install the version I want to use for NetBeans development.*

3. Install *NetBeans* (6.8). Double-click on installer:

When the installer begins click:

[Customize]

The only portions needed for cellphone development are (ALL 5 ARE NECESSARY):

Base IDE
 Java SE
 Java Web and EE
 Java ME
 Features on Demand

(Declining to install additional portions will save ~200 MB)

Note: you cannot remove (uninstall) NetBeans without Java SE installed on the machine.

(You may wish to drag a copy of *NetBeans IDE* to your desktop)

3.1 Update NetBeans:

[Help] → [Check for Updates] → [Installed] → [Next] → [Restart IDE Now] → [Finish]

After restart:

[Tools] → [Plugins] → [Update] → [Next] → [Restart IDE Now] → [Finish]

NOTE: Java ME will get activated when a Java ME project gets created.

NOTE: after creating a Java ME project, but *before* installing any emulators the NetBeans “Show JavaDoc” feature works correctly. Clicking,

[Tools] → [Java Platforms] → [Java(TM) Platform Micro Editon] → [JavaDoc]

shows:

Program Files\NetBeans 6.8\mobility8\Java_ME_platform_SDK_3.0\docs\api*
 Program Files\NetBeans 6.8\mobility8\Java_ME_platform_SDK_3.0\docs\api\html-helpset
 Program Files\NetBeans 6.8\mobility8\Java_ME_platform_SDK_3.0\docs\api\html-InstallWinMobile

Add the Samsung emulators:

[Tools] → [Java Platforms] → [J2ME] → [Add Platform] → [Java ME MIDP Platform Emulator] → [Next]

Highlight:

[Samsung_SDK_112] → [Open] → [Next]

Add the LG emulators:

[Tools] → [Java Platforms] → [J2ME] → [Add Platform] → [Java ME MIDP Platform Emulator] → [Next]

Check the box:

[LGMobile\JavaME_SDK_15]

Click:

[Next] → [Finish] → [Close]

4. Install emulators. No need to unzip any of the files, just double click:

Nokia Series 60 3rd Edition Feature Pack 1:
(double-click *nS60_jme_sdk_3rd_e_FP1.zip* and double-click *setup.exe*)

Nokia Series 40 3rd Edition Feature Pack 2:
(double-click *S40_SDK_3rd_Edition_FP2_1.0.zip* and double-click *setup.exe*)

Nokia Series 40 5th Edition emulator:
(double-click *Series_40_5th_Edition_SDK_Feature_Pack_1_1_0.zip* ...)

Nokia Series 40 6th Edition emulator:
(double-click *Series_40_6th_Edition_SDK_v1_0_en.zip* ...)

(Note: installing Nokia emulators also installs *Nokia Connectivity Framework*)

Samsung Emulators:
(double-click *Samsung_SDK_1.1.2.zip*)
It appears that Samsung's emulators require *Java 1.6* or higher, and *Quicktime Player 7*:

tools\samsung\QuickTimeInstaller.exe

4. Install *Nokia Ovi Suite*.

5. Setup up Nokia Ovi Suite (Note: if you don't have a Nokia phone at hand you won't be able to complete this setup):

Make sure computer is Bluetooth enabled (plug in Bluetooth dongle)
Bring a bluetooth-enabled phone(s) to computer.
Make SURE phone is set to Bluetooth ON and Bluetooth DISCOVERABLE

[<Nokia Ovi Suite in system tray>] → [Open Noki Ovi Suite] → [Tools] →
[Add new device] → [Bluetooth] → [Next]

Ovi Suite will search for devices and present a list of both connected and newly discovered ones. Click on phone:

[E72-1] → [Next] → [Next arrow]

Ovi Suite will offer a *passcode*. Type the passcode into the phone.

[123] → [OK]

Ovi Suite will be busy for 60+ seconds while it is “installing the Bluetooth modem.” When done click:

[Next]

Ovi Suite will request a name for the device:

E72-1

Click:

[Finish]

Repeat for additional phones.

6. Apprise NetBeans of the S60 emulator. Start NetBeans. Then:

[Tools] → [Java Platforms] → [J2ME] → [Add Platform] →
[Java ME MIDP Platform Emulator] → [Next]

A FileChooser will appear. Specify:

[C:\S60\devices\S60_3rd_MIDP_SDK_FP1](#)

Click:

[Open]

7. Apprise NetBeans of S40 emulator. Start NetBeans.

First, activate Java ME:

[File] → [New Project] → [Java ME] → [Next]

This will take a minute. When NetBeans is ready:

[Cancel]

Now, locate the emulators:

**[Tools] → [Java Platforms] → [J2ME] → [Add Platform] →
[Java ME MIDP Platform Emulator] → [Next]**

NetBeans will find the Series 40 emulators:

C:\Nokia\Devices\S40_6th_Edition_SDK
C:\Nokia\Devices\S40_5th_Edition_SDK_Feature_Pack_1
C:\Nokia\Devices\S40_SDK_3rd_Edition_Feature_Pack_2

Make sure all are “selected”

[Find More Java ME Platform Folders]

Browse to:

[C:\S60\devices\S60_3rd_MIDP_SDK_FP1] → [Open]

The S60 emulator will appear in the list of platform folders:

[Finish] → [Close]

8. **(OPTIONAL)** To work with sound files install Audacity. Locate Audacity at:

audacity.sourceforge.net

Also, install FFmpeg library necessary for loading various sound formats:

Start audacity,

[Edit] → [Preferences] → [Libraries] → [FFmpeg Library: Download]

to download the FFmpeg library. After downloading the library click,

[FFmpeg] → [Locate]

to install the library.

Creating a New MIDlet with NetBeans 6.8

1. Open NetBeans. Close the “Start Page” tab in the main pane.

2. Choose:

[File] → [New Project]

A form will appear. Highlight:

JavaME and **Mobile Application**

Click:

[Next]

A new form will appear.

3. Fill in the form:

Project Name: **Hello**
 Project Location: **C:\active\celldev**
 Project Folder: **C:\active\celldev\Hello**

Set as Main Project: **checked**
 Create Hello MIDlet: **unchecked**

Click:

[Next]

A new form will appear.

4. Fill in the form:

Emulator Platform: **S40 SDK 3rd Edition**
 Device: **S40 SDK_3rd_Edition**
 Device Configuration: **CLDC-1.1**
 Device Profile: **MIDP-2.0**

Choice of CLDC-1.0 and MIDP-1.0 will allow application to run on more devices

Click:

[Next]

A new form will appear.

5. This form is where one can specify that a project should build multiple version of the code for different kinds of phones. Don't make any changes to this form.

Click:

[Finish]

6. Create the Java file containing the “main” method (extends MIDlet). Choose:

[File] → [New File]

A form will appear. Highlight:

MIDP and **MIDlet**

Click:

[Next]

A new form will appear. Fill it in:

MIDlet Name: **Hi**
 MIDP Class Name: **Hi**
 Project: **Hello**
 Created File: **C:\active\celldev\Hello\src\Hi.java**

Click:

[Finish]

Paste code into Hi.java

7. Create additional necessary Java files. Choose:

[File] → **[New File]**

A form will appear. Highlight:

Java and **Java Class**

Click:

[Next]

A new form will appear. Fill it in:

Class Name: **Helloscreen**

Click:

[Finish]

Paste code into Helloscreen.java

8. Build a Series 60 version and run in emulator. In **Projects** pane (upper left):

]Hello[→ **[Properties]** → **[Platform]**

Fill in form:

Select Platform Type: **CLC/MIDP**
 Emulator Platform: **S60 3rd Edition FP1 SDK for MIDP**
 Device: **S60Emulator**

Click:

[OK]

Clean and Build Main Project

Run Main Project

9. Deploy to S60 phone. Create deployment instance. In **Projects** pane (upper left):

]Hello[→ [Properties] → [Deploying]

From pull down menu:

Select Deployment Method: **Nokia Terminal connected via PC Suite**

Click:

[Manage Deployment] → [default] → [Add Instance]

New Instance Name: **N95**

Click:

[OK]

[Deploy to Selected Devices]

In “All connected devices” pane:

[N95] → [Close] → [OK]

Select deployment instance:

]Hello[→ [Properties] → [Deploying]

From “Deployment Instance” menu select:

[N95] → [OK]

Clean and Build Main Project

10. Build a Series 40 version. I **Projects** pane (upper left):

]Hello[→ [Properties] → [Platform]

Fill in form:

Select Platform Type:	CLC/MIDP	
Emulator Platform:	S40 6th Edition SDK	
Device:	S40_6th_Edition_SDK	
Optional Packages	uncheck all	(this is not necessary)

Click:

[OK]

Clean and Build Main Project

Run Main Project

Moving an Existing MIDlet from One Machine to Another

Assume a project has been created using NetBeans, the entire project (and all subdirectories) have been backed up, and that backup should be restored to the original (or another) machine. Example:

Entire cell phone software directory (c:\active\celldev) has been saved to a pendrive. The celldev directory contains multiple project directories, including Bongo.

To move Bongo to a new, naked machine:

1. Perform all steps in “Installing Cellphone Development Environment.”
2. Perform steps 1 – 6 in “Creating a New MIDlet with NetBeans 6.5.1” being careful to replace project name “Hello” with “Bongo,” and MIDlet name “Hi” with “Bongo.”
4. Close Bongo.java in NetBeans. Copy all source files and resources (image and sound) files into the src directory that NetBeans created for Bongo:

c:\active\celldev\Bongo\src

5. Compile the system.
6. Set emulator:

[Bongo] → [Properties]

Set,

Emulator Platform: **S60 3rd Edition FP1**
Device: **S60 Device**

Switch panes:

[Deploying]

Tell NetBeans to use PC Suite:

Select Deployment Method: **Nokia Terminal connected via PC Suite**
Deployment Instance: **default**

[Manage Deployment] → [Deploy to all connected devices] → [Close]

Sharing Classes Between Projects

Many classes are reusable across classes. In order to maintain a single copy of the source code, and share the class files NetBeans allows this approach:

- create a new 'Library Project' for the common code
- apprise NetBeans of the new library
- let each application project use the library

Create a Library Project (called unme2)

1. Create a new Library Project:

[File] → [New Project] → [Java ME] → [Mobile Class Library] → [Next]

Project Name: **unme2**
 Project Location: **C:\active\celldev**
 Project Folder: **C:\active\celldev\unme2**

Click:

[Next]

A new form will appear:

Device Configuration: **CLDC-1.1**
 Device Profile: **MIDP-2.0**

Click:

[Next]

A new form will appear:

(that may be ignored)

Click:

[Finish]

2. Make the new Library Project the “main project”:

[Projects] →]unme2[→ [Set as Main Project]

3. Add a source file to the Library Project:

[File] → [New File] → [Java] → [Java Class] → [Next]

A new form will appear:

Class Name: XXX

Click:

[Finish]

Paste or edit file as necessary

4. Clean and build Library Project. This will create the necessary dist\lib directory that will contain the .jar file:

unme2.jar

Apprise NetBeans of the New Library

1. Add the new library to NetBeans list of Global libraries.

[Tools] → [Libraries] → [New Library]

A form will appear:

Library Name: **unme2**
Library Type: **Class Libraries**

Click:

[OK]

and the list of Global Libraries will be expanded to include unme2.

2. Specify classpath for library.

[Classpath] → [Add Jar/Folder]

A file chooser dialog will appear. Navigate to:

C:\active\celldev\unme2\dist\lib\unme2.jar

Click:

[Add Jar/Folder]

3. Specify the source file location:

[Sources] → [Add Jar/Folder]

A file chooser dialog will appear. Navigate to:

C:\active\celldev\unme2\src

Click:

[Add Jar/Folder]

Click:

[OK]

Use Library in Other Projects

In this example, let the Bongo project use the unme2 library. To do this:

1. Make Bongo the Main Project:

[Projects] → [Bongo] → [Set as Main Project]

2. Tell Bongo to use the new library:

[Projects] → [Bongo] → [Properties] → [Libraries & Resources]

Set:

Project Configuration: **Default Configuration**

Click:

[Add Library] → [unme2] → [Add Library] → [OK]

That's it! Repeat for each project that needs the library.

Multiple Versions of a Library

It can be useful to create multiple versions of a library, as for debugging. The unme2 Java classes include some debugging classes:

```
Log.java
Packetdebug.java
Memuse.java
```

None of these classes should be used by, or included in, any production software. So, I created the unme2 library, as already described. Then I:

- added a second *Project Configuration* named *Debug*
- arrange for debugging classes only to be used in the Debug configuration
- created another library: unme2debug
- tell Bongo to use unme2debug in its Debug configuration

Here are the steps:

1. Add another project configuration:

[unme2] → [Properties] → [Manage Configurations] → [Add]

Set:

New Configuration Name: **Debug**

Click:

[Close]

2. Ignore debugging classes in Debug configuration:

[unme2] → [Properties]

Select:

Project Configuration: **DefaultConfiguration**

Ignore:

[Sources Filtering]

Uncheck debugging files:

Log.java Packetdebug.java Memuse.java

Reset Configuration:

Project Configuration: **Debug**

Uncheck:

Use Values from "DefaultConfiguration"

Make sure all debugging classes (Log.java, Packetdebug.java ...) ARE checked.

Click:

[OK]

3. Clean and compile unme2 in *DefaultConfiguration* and *Debug* configuration.

4. Create another library called unme2debug:

[Tools] → [Libraries] → [New Library]

Specify:

Library Name: **unme2debug**
Library Type: **Class Libraries**

Click:

[OK]

Specify library details:

[Classpath] → [Add JAR/Folder]

Choose:

C:\active\celldev\unme2\dist\Debug\unme2.jar

[Sources] → [Add JAR/Folder]

Choose:

C:\active\celldev\unme2\src

Click:

[OK]

5. Adjust Bongo *Debug* configuration to use unme2debug library:

[Projects] → [Bongo] → [Properties] → [Libraries & Resources]

select:

Project Configuration: **Debug**

Uncheck:

Use Values from “DefaultConfiguration”

Click:

[unme2] → [Remove]

Click:

[Add Library] → [unme2debug] → [Add Library] → [OK]

NetBeans Problems

I've had all variety of problems getting NetBeans to work correctly:

1. Can't find Java SE documentation
2. Emulators complain about missing capabilities on startup
3. Nokia Connectivity Framework presents empty screen
4. Signed MIDlets are trusted on S40 but not S60 phones

I need to get to the bottom of this mess.

1. I removed NetBeans, all Nokia emulators, Nokia PC Suite and Nokia Connectivity Framework.
I deleted directories:

[c:\Nokia](#)

[c:\S60](#)

[c:\Documents](#) and Settings\Matt\.nbi

[c:\Documents](#) and Settings\Matt\.netbeans

[c:\Documents](#) and Settings\Matt\.netbeans-registration

[c:\Documents](#) and Settings\Matt\.Nokia

[c:\Documents](#) and Settings\Matt\Application Data\Nokia

2. I installed NetBeans 6.8 and recreated Bongo project.

Result: NetBeans works, and all J2SE/J2ME works correctly

3. I installed Series 60 emulator

- 4.