# Archived:Getting started with Java on Maemo

Archived: This article is **archived** because it is not considered relevant for third-party developers creating commercial solutions today. If you think this article is still relevant, let us know by adding the template {{ReviewForRemovalFromArchive|user=~~~~|*write your reason here*}}.

The article also has many broken links.

## Introduction

Nokia does not *officially* support Java on its Maemo platform, yet. Of course it is technically possible to develop every type of application in C/GTK. But enabling the Internet Tablet to run Java programs offers possibilities beyond the programming language: Java is the most used language for development of third party mobile applications, and the Java platform is the unrivaled Number One for development of enterprise server side applications.

Enabling the Internet Tablets to run Java means enabling the wide audience of Java developers to deploy their ideas to the Internet Tablets and to use the same technology for the development of the front end, as for the development of back end services.

This article covers the development for Maemo using the Jalimo platform. Beside Jalimo, there are other approaches to bring a Java platform to Maemo. In contrast to the others, unique characteristic of jalimo is not only to concentrate on a runtime for the device, but to support the complete application development and life cycle. In short, the aim of Jalimo is to provide everything a Java developer needs to feel comfortable with developing for mobile and embedded Linux.

## Access to the device

If you start developing for your Internet Tablet, at first you should get shell access. At your convenience follow the short Maemo tutorial How Do I Become Root. After that, you are able to login on the device using ssh (e.g. using PuTTY on Windows). With ssh, you also have a very comfortable solution for file exchange with the device, using the scp command (check out WinSCP on Windows).

To have full control over all installed packages in your device, you should enable the Red Pill mode of the Application Manager. This is of course not necessary, if you prefer using the *apt-get* command over ssh.

## Jalimo

The central point for access to information about Jalimo is www.Jalimo.org. Beyond Maemo 3 and Maemo 4, Jalimo supports OpenMoko and other Mobile and Embedded distributions, as well. For every supported distribution there is a package repository with pre-built packages.

The Jalimo packages contain the Virtual Machine cacao, together with GNU Classpath. This combination aligns on the Java 1.5 SE capabilities, but currently does not provide Swing/AWT on Maemo. At the moment, the best supported Java GUI solution on Maemo is Eclipse SWT, so only this is covered in this article. The choice of SWT was made because it is very powerful and its fast GTK peers, which render with the platforms look and feel. Furthermore it is one of the most portable solutions for Java (running on Linux, Windows ,as well as Windows Mobile and Symbian S80 in its eSWT variant).

## Installing Packages

To install the packages and verify, that they are working correctly, simply point your Internet Tablet to http://jalimo.evolvis.org/repository/maemo/jalimo-swt-example-install.php of the Jalimo example application. Alternatively you can do this manually, following this steps:

Add the jalimo package repository, using the Application Manager:

| | |
|---|---|
| **Repository:** | *http://jalimo.evolvis.org/repository/maemo/* |
| **Distribution:** | *Maemo 3.2* or *chinook* for Maemo 4.x |
| **Component:** | *user* |

After that, you can install the packages of your choice (e.g. using the Application Manager or the shell). Installing the package


(C) Copyright Nokia 2014. All rights reserved.

*libswt-gtk-3.4-java* will also install *cacao* as well as *classpath*, since it depends on them. The above example is contained in the package *jalimo-swt-example*. To install all together in a shell, simply execute the following as root:

```
echo "deb http://jalimo.evolvis.org/repository/maemo/ chinook user" >>
/etc/apt/sources.list
apt-get update && apt-get install jalimo-swt-example
```

Or for your IT OS 2007:

```
echo "deb http://jalimo.evolvis.org/repository/maemo/ Maemo3.2 user" >>
/etc/apt/sources.list
apt-get update && apt-get install jalimo-swt-example
```

## First Development

For your Java development, you can choose your preferred IDE (e.g. Eclipse would be a good choice). Of course it is also suitable to start with a simple text editor and any console java compiler (e.g. the Sun *javac*). Save the example code in HelloWorld.java

```
public class HelloWorld {
 public static void main(String[] args) {
                System.out.println("Hello Jalimo");
 }
}
```

and compile it with *javac HelloWorld.java*. Copy the *HelloWorld.class* to your Internet Tablet and call

```
cacao HelloWorld
```

There you are! Running your own code on the Internet Tablet in just a few minutes, using a modern programming language and no need to install big toolchains or do cross-compiling.

## Serious Development

For serious development, you need of course more than a console application. Jalimo provides different solutions for GUI development on Maemo. At first, there is support for SWT, as shown in the following example.

### SWT Sample application

```
import org.eclipse.swt.SWT;
import org.eclipse.swt.layout.RowLayout;
import org.eclipse.swt.widgets.*;

public class SimpleSwtSample {

 public static void main(String[] args) {
  Display display = Display.getDefault();
  final Shell shell = new Shell(display);
  shell.setLayout(new RowLayout(SWT.VERTICAL));

  Label label = new Label(shell, SWT.CENTER);
  label.setText("Hello maemo");
```


(C) Copyright Nokia 2014. All rights reserved.

```
  Button button = new Button(shell, SWT.NONE);
  button.setText("close");
  button.addListener(SWT.Selection, new Listener() {
   public void handleEvent(Event arg0) {
    shell.dispose();
   }
  });

  shell.open();
  while (!shell.isDisposed())
   display.readAndDispatch();
 }
}
```

Compile this sample code against the Eclipse SWT libraries using any java compiler. You can get the libraries from the SWT Homepage for different platforms) . Run the application on your host. Copy all *.class* files to your jalimo enabled device (e.g. using scp, or WinSCP on Windows) and call

```
cacao -cp /usr/share/java/swt-gtk.jar SimpleSwtSample
```

You can find this and example together together with a slightly bigger one in the svn. This full example contains a complete packaging description and menu integration.

## Debian packaging

Of course an application needs a proper packaging. For Maemo this means packaging in debian (.deb) archives. The most java developers are not familiar with linux distribution integration. Therefore we have created a packaging plugin for apache maven (the defacto standard build system for java). To use this plugin, simply create a maven project and add the evolvis plugin repository, as well as the packaging configuration to your pom.xml. You can find an example here or in the above SWT example. After that, you simply have to run

```
mvn pkg:deb
```

in your project to get a suitable .deb archive. One of the benefits of the plugin is the capability to translate the maven dependencies to debian package dependencies.

To get a Maemo menu entry for you application, simply create a proper .desktop file and add it to the resources description of the packaging plugin configuration, with the target location */usr/share/applications/hildon/*. If you want your application to be installed by normal users using the Maemo Application Manager, you should choose *user/<SUBSECTION>* as configured debian package section. See the Application Manager How To for further details.

## Further reading

Of course, Jalimo has some more espects e.g. the scratchbox integration or SCIO, the XML communication library form small devices. Take a look at the Jalimo website for this information.


(C) Copyright Nokia 2014. All rights reserved.

(C) Copyright Nokia 2014. All rights reserved.