

# Beginner's Guide to Building J2ME Apps

First install Java SDK<sup>1</sup>(and don ask me why) If ya have lazy comp such as mine ( PII, 350 MHz, 160MB RAM, Win XP), I recommend to install Java 2 SDK **Version 1.4.2\_08** . You might install Tiger (Java 1.5) but Eclipse will hang a little bit (if you don know what's Eclipse don worry).

Where to find Java SDK? Go to Sun's web site, I think it is <http://www.sun.com> , find their **Search** box and type: **j2sdk-1\_4\_2\_08-windows-i586-p.exe** or Java 2 SDK 1.4.2\_08

When you download it on yr comp, just start typical installation, don change default settings. That's how you'll have both Java SDK and Public Java RE<sup>2</sup>.

Bravo! Now it's time to install Eclipse. It's really coooool IDE<sup>3</sup>, you can fill it up with coooool plugins such as J2ME-Polish, or ElipseME so you can develop coool J2ME apps.

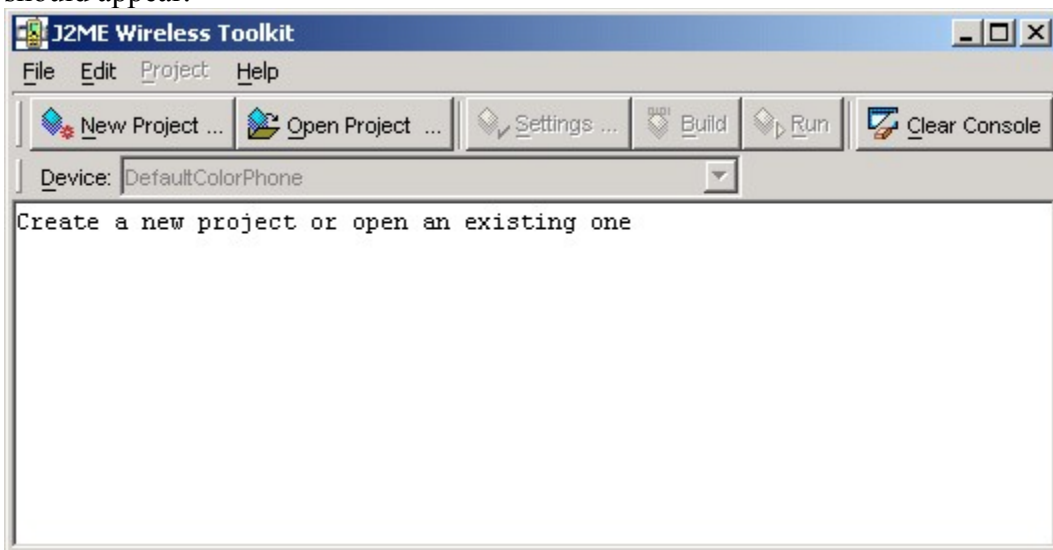
Download and installation of Eclipse is easier than installing Java SDK. Just go to Google and type "Eclipse 3.0" and that's it? On Eclipse site you better find version 3.0. File name is probably **eclipse-SDK-3.0-win32.zip** or something like that.

Download it on yr comp and just extract it somewhere. I always extract it at ROOT C:\

I use WinRAR for that task.

Now that you installed Java and Eclipse, go again to Sun's website and search for J2ME Wireless Toolkit. It's free. The file name should be **j2me\_wireless\_toolkit-2\_2-windows.exe**

Download it and install it. When ya finish it up, start its tool **Ktoolbar**, just to see the magic of emulation. First, this window should appear:

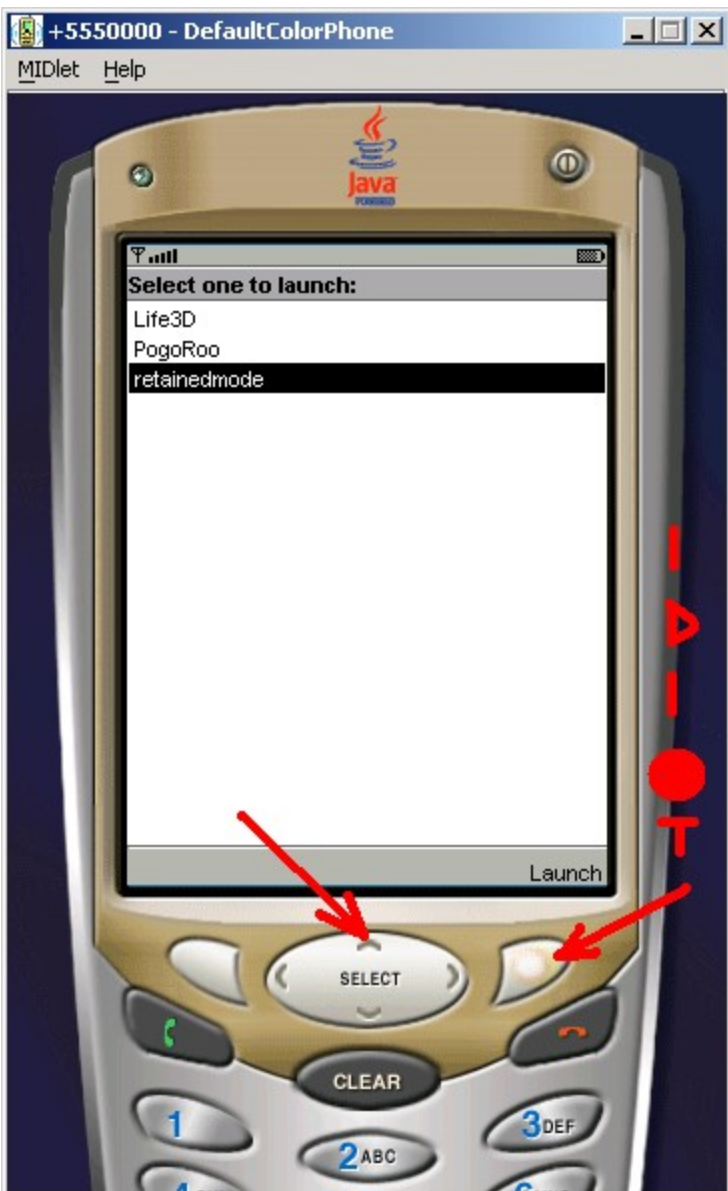


Choose **Open project**, select **Demo3D**, **Open Project**, then **Run** and emulator will appear

<sup>1</sup> Java Standard Development Kit – coooool stuff; you can develop anything what you can imagine with Java SDK for free. What you need beside SDK is Notepad or Wordpad or MS Word, and the world of programming is yours. But...

<sup>2</sup> Java Runtime Enviroment – send me email if you want to hear my opinion about it (zareac@gmail.com)

<sup>3</sup> IDE is Integrated Development Enviroment. It helps you develop yr poetry in Java, it helps you edit yr sonets, scrub 'em, compile, run, debug, observe, implement, inherit and all other sweet things from one place. You can do it all in MS Word or Notepad, and then click File/Save, and then go on Start/Run/cmd.exe and then change directory with cd\ and then find emulator.exe or compiler javac.exe, and so on, and on – it will kill you. Definitely use any IDE, and it's best if IDE is free, such is Eclipse



If you want to see application you should definitely click on buttons pointed by red arrows. It's cool isn't it? Guys from Sun did their job. But... What if you want to create yr own apps. That's where Eclipse jumps in. Go to folder where you installed Eclipse and start the tool. This shouldn't be a problem. You'll find there an icon **eclipse.exe**



Eclipse.exe

It's nice, isn't it. Start the eclipse, and follow the steps on the following pages. Actually, you have to install EclipseME Plugin. It's cool stuf, it lets you run emulator from Eclipse IDE. That means you can write yr Java apps, compile them, and then run them in emulator, just like we did with Demo3D. Installation of EclipseME is little bit tricky if yr an idiot such as me. Why is that? There are two ways to install it: over the net or from yr hard drive. If ya **didn't** download **eclipseme.feature\_0.7.5\_site.zip**

file from EclipseME web site, and you have fast link, you might start the installation over the Net. But if you're poor bastard with dial-up, then you better go to your ex-girlfriend and download the file to install it locally, that is from yr hard drive. And don worry, you have all the prerequisites for doing the installation, but if you're not sure check next chapters. Now what follows is description of EclipseME installation. It is description of TWO DIFFERENT WAYS to install the bitch, so be carefull what you want, and what you are. Two options are

**Installing via the EclipseME Update site!!!!** (over the Net)

**Installing via a downloaded archive site!!!!** (from the hard drive)

Now read this EclipseME tutorial, and after that I'm coming back.



## Installation

This document provides instructions on how to install or upgrade EclipseME.

### **0. Before you begin**

1. **Install the Wireless Toolkit(s)**
2. **Installing EclipseME**
  - a. **Decide on your approach**
  - b. **Installing via the EclipseME Update site**
  - c. **Installing via a downloaded archive site**
3. **Configure the Wireless Toolkit Settings**
4. **Change Eclipse's Debug Settings**
5. **Configure ProGuard (Optional)**
6. **Configure OTA options (Optional)**
7. **Updating your EclipseME installation**
  - a. **If you installed via the EclipseME Update Site**
  - b. **If you installed a downloaded archive site**
8. **Removing EclipseME**
  - a. **The "Normal" way**
  - b. **The "Brute Force" method**
  - c. **Removing EclipseME traces from a project**

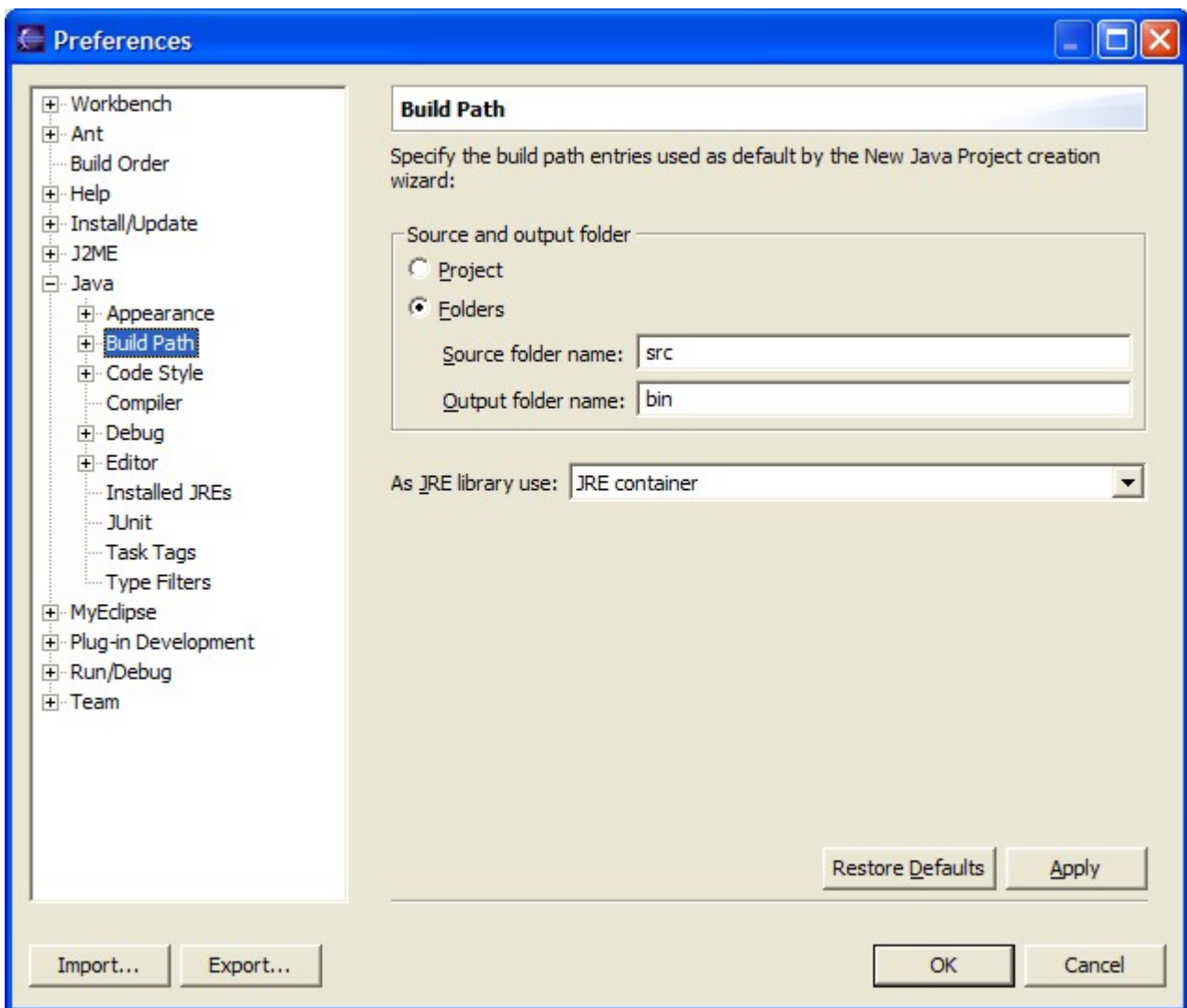
---

### **0. Before you begin**

1. Please make sure you have all the [prerequisites](#). (you have, don worry)
2. If you have a version of EclipseME prior to 0.5.0 installed, remove it before installing a new one. In addition, read the [migration instructions](#) to see if there are any specific steps you need to take to update a project created with the old version to the new version.

If you have EclipseME version 0.5.0 or later installed, you do not need to remove it. The installation procedure below will automatically upgrade your EclipseME installation.
3. Although it is not mandatory, we strongly, strongly suggest that you configure Eclipse to use separate source and output folders in new projects. Regrettably, this is not Eclipse's default setting. **(do this!!!)**

To make this change, Select the *Preferences* menu item from Eclipse's *Window* menu, expand the *Java* item in the left pane and click on the *Build Path* item. In the right-hand pane, select the *Folders* radio button. After you do this, the dialog should look like this:



If you do not configure Eclipse for separate source and output folders, some features of EclipseME, such as resource folders, will not work.

4. Finally, if you have been using Eclipse without EclipseME to do J2ME development using a Wireless Toolkit, your Eclipse project may need some classpath changes. See the section on [converting an existing project to an EclipseME project](#).

## 1. Install the Wireless Toolkit(s)

Install any of the wireless toolkits you want to use. When you do this, make note of the base directory into which each is installed, as you will need this information in Step 3. (**you did this, didn't you?**)

Unix users - if you install a Wireless Toolkit for use by more than one user on your system, please ensure that all the users have the appropriate read and execute permissions on the contents of the toolkit. For example, WTK's contain a preverifier that EclipseME will need to access during the build process. If the user doing the build does not have execute permissions on the preverifier, the build will fail. Similarly, if the user does not have execute permissions on the emulator, he or she will not be able to test the build.

## 2. Installing EclipseME

Beginning with version 0.5.5, EclipseME is provided as an Eclipse "archive site." With version 0.5.0 and earlier, one simply unzipped the distribution file into the Eclipse installation directory or plugin directory. **This is no longer the installation procedure.** Please follow the instructions below instead.

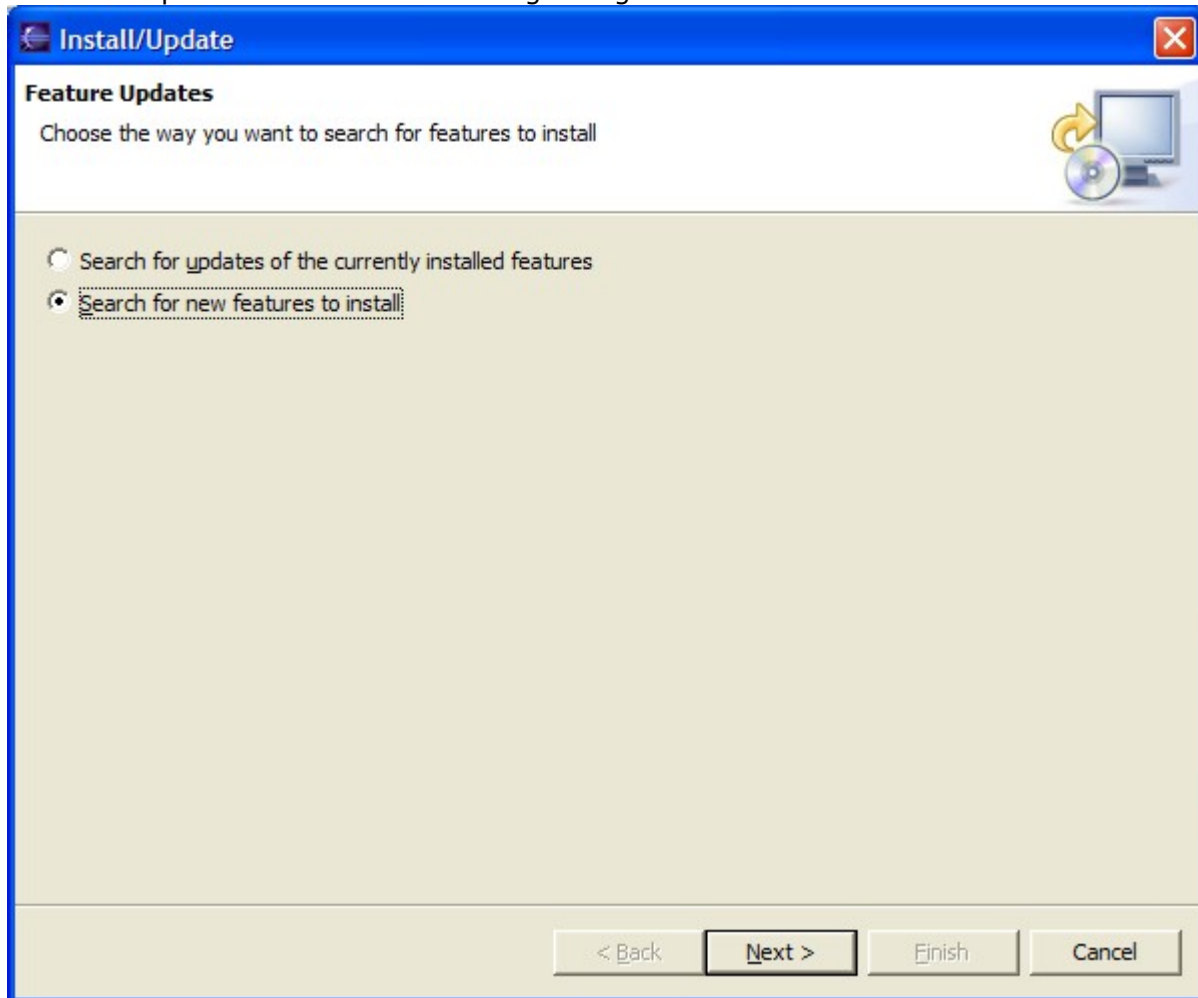
If you are updating EclipseME, you should also use the procedure below. New installations and updates follow the same series of steps.

There are two basic ways of installing or updating EclipseME:

1. You can simply to use Eclipse's built-in updates feature to directly install or update via EclipseME's update site.
2. You can download an archive site file manually, and then install from it.

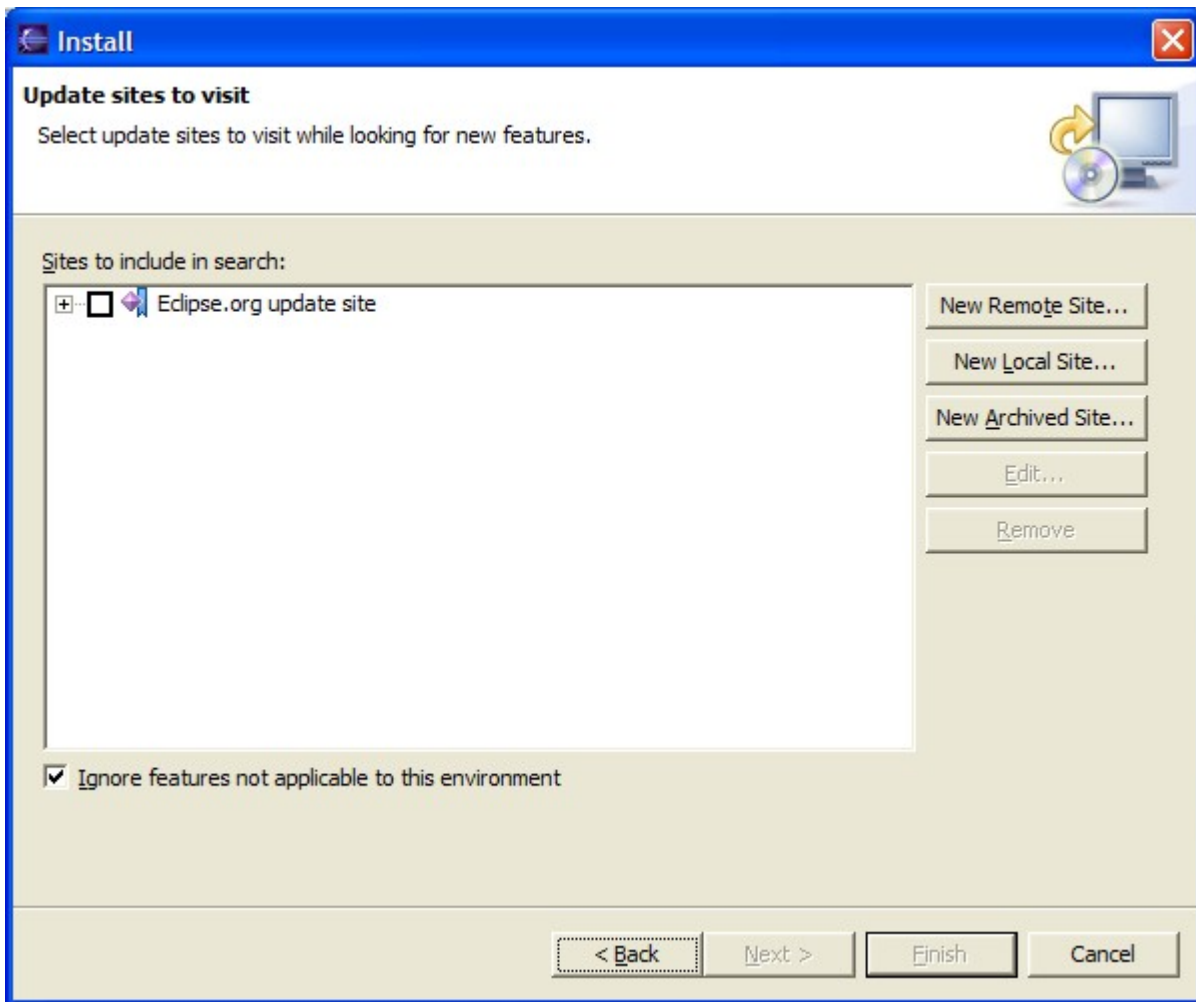
## **Installing via the EclipseME Update site!!!!**

1. From the Eclipse *Help* menu, select *Software Updates* and then *Find and install...*
2. You will be presented with the following dialog:



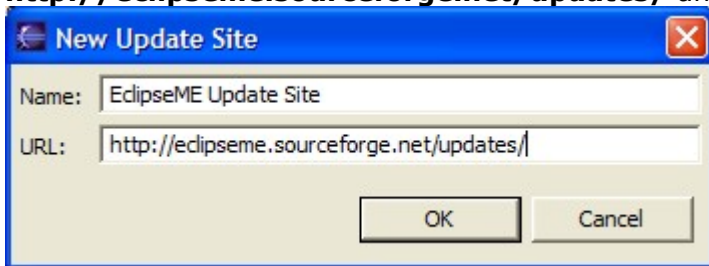
Select the **Search for new features to install** radio button.  
Press **Next**.

3. You will next be presented with the following dialog:

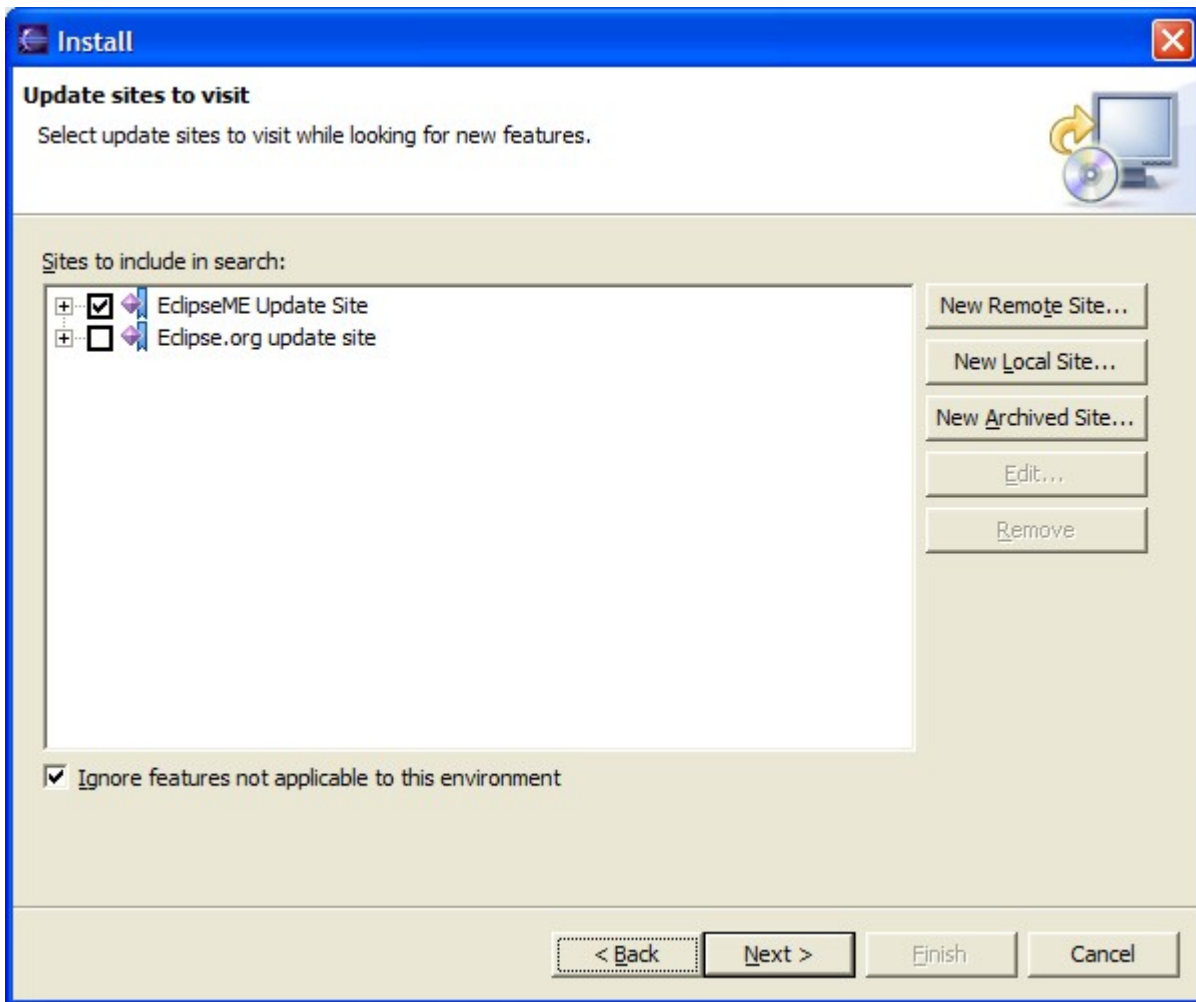


Press the **New Remote Site...** button.

4. In the **New Update Site** dialog that appears, enter a name for the update site and the URL **<http://eclipseme.sourceforge.net/updates/>** and press **Open**.

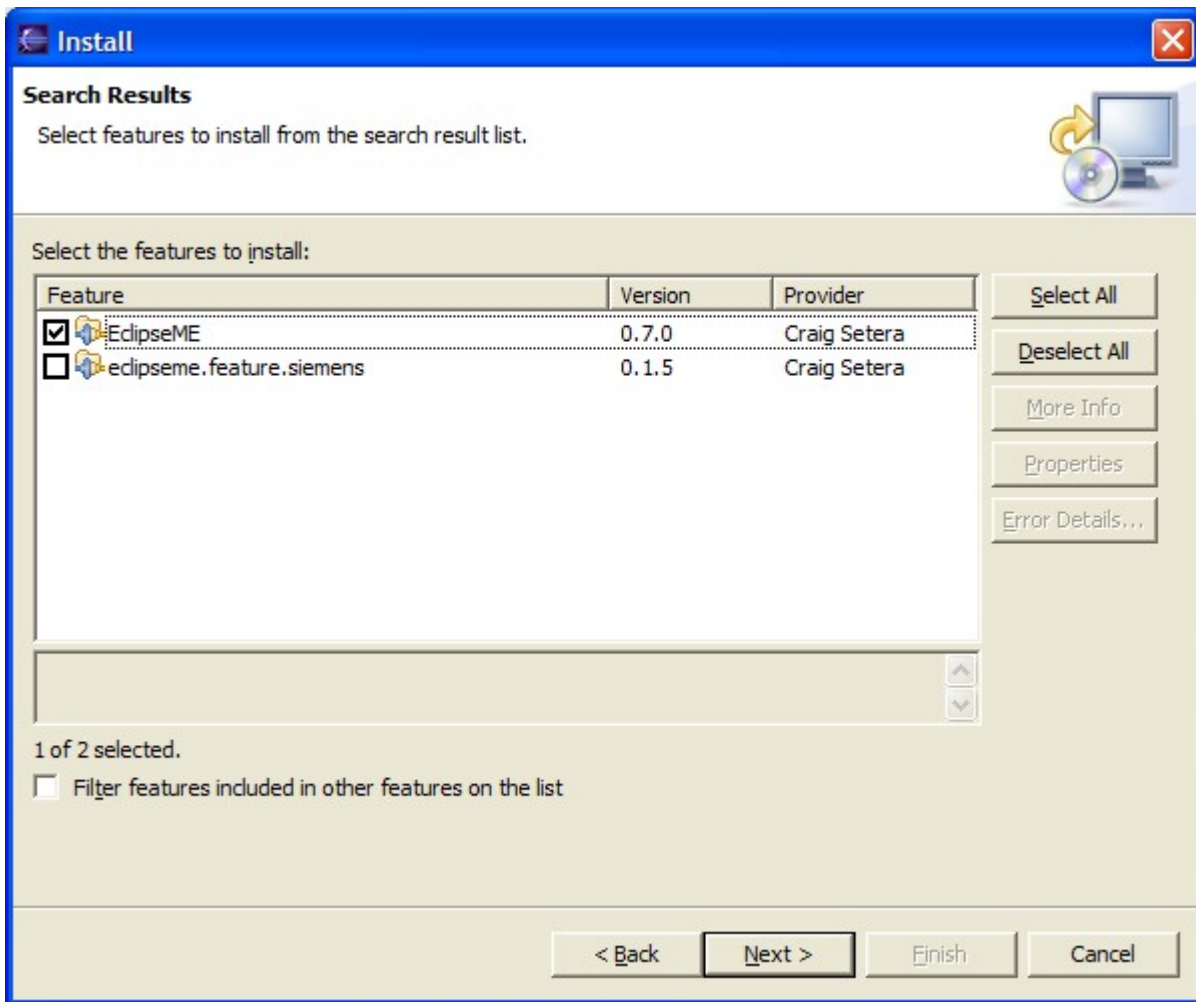


5. The update site will now be listed in the **Install** dialog.



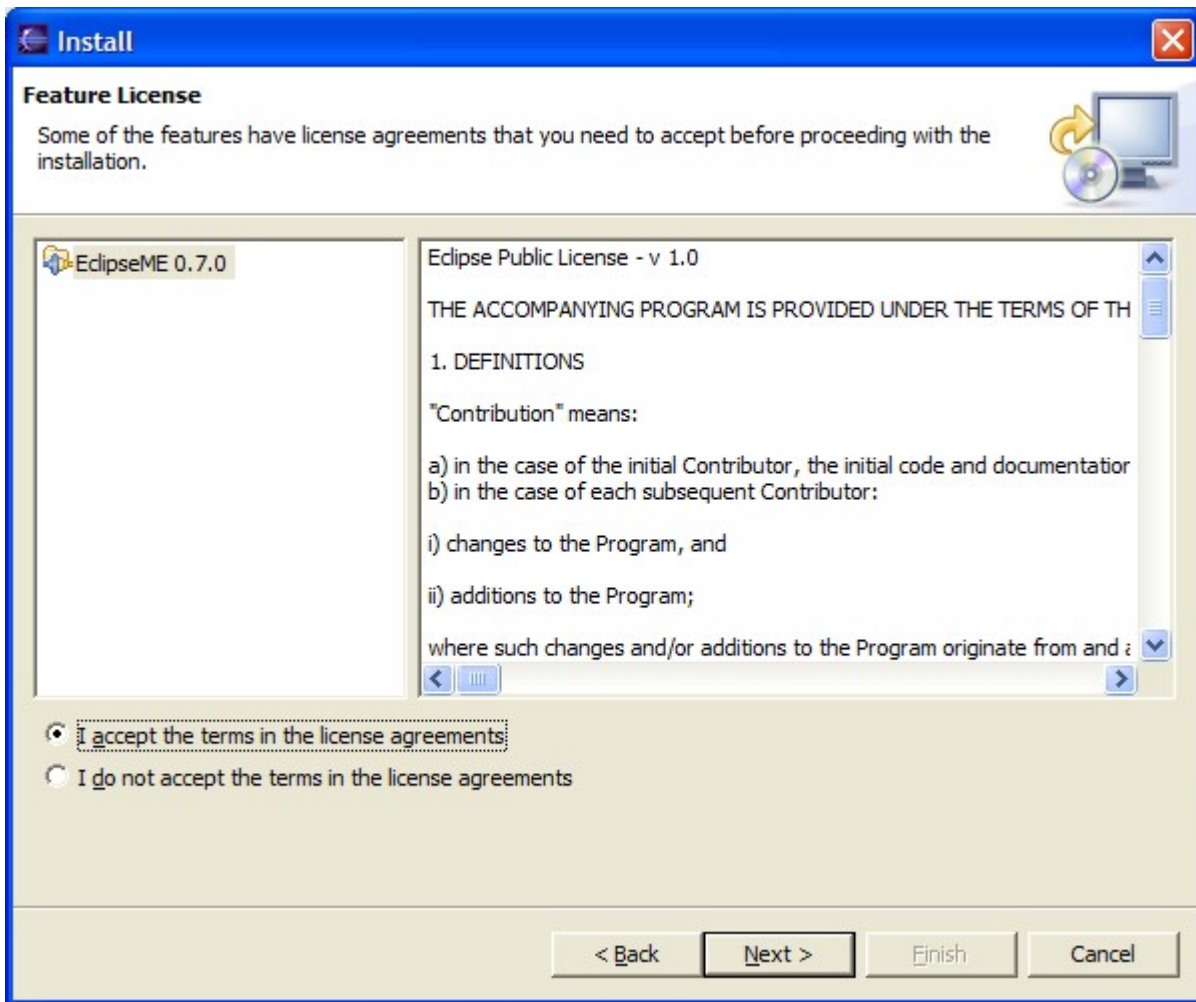
Ensure that there is a check next to the EclipseME update site, then press **Next**.

6. You will next be presented with the following dialog:



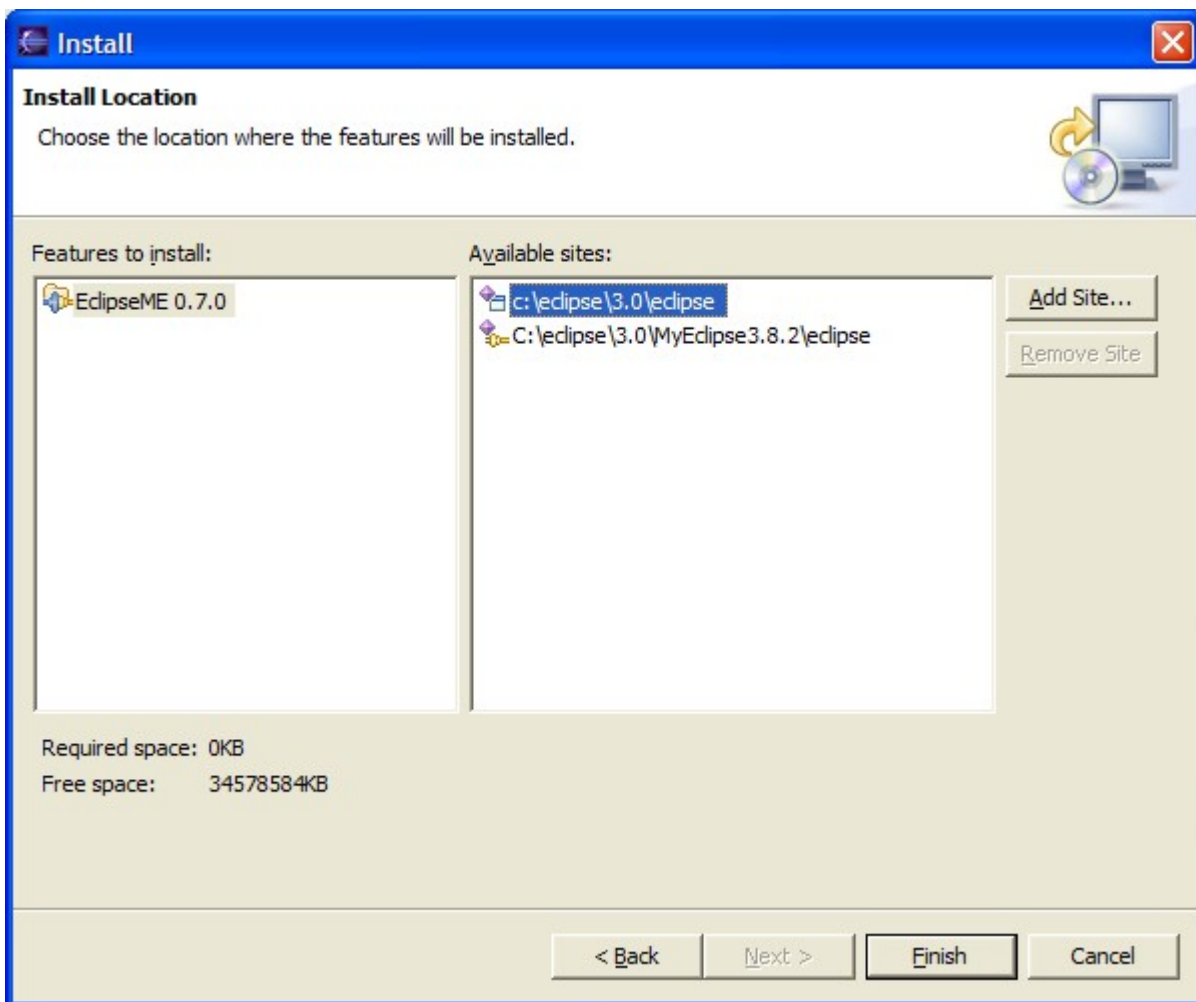
Ensure that there is a check next to **EclipseME**. Other available features can be installed by checking them as well. Press **Next**.

7. The installation process will next display the license agreement for EclipseME.



Select **I accept the terms of the license agreements** and press **Next**.

8. The next dialog that appears displays the possible locations into which you can install EclipseME.



In many cases, the only site that will be listed will be the main Eclipse installation directory. If you have other plugins or features installed, however, you may see additional sites.

Although you may install EclipseME elsewhere, we recommend that you install it in your main Eclipse installation directory.

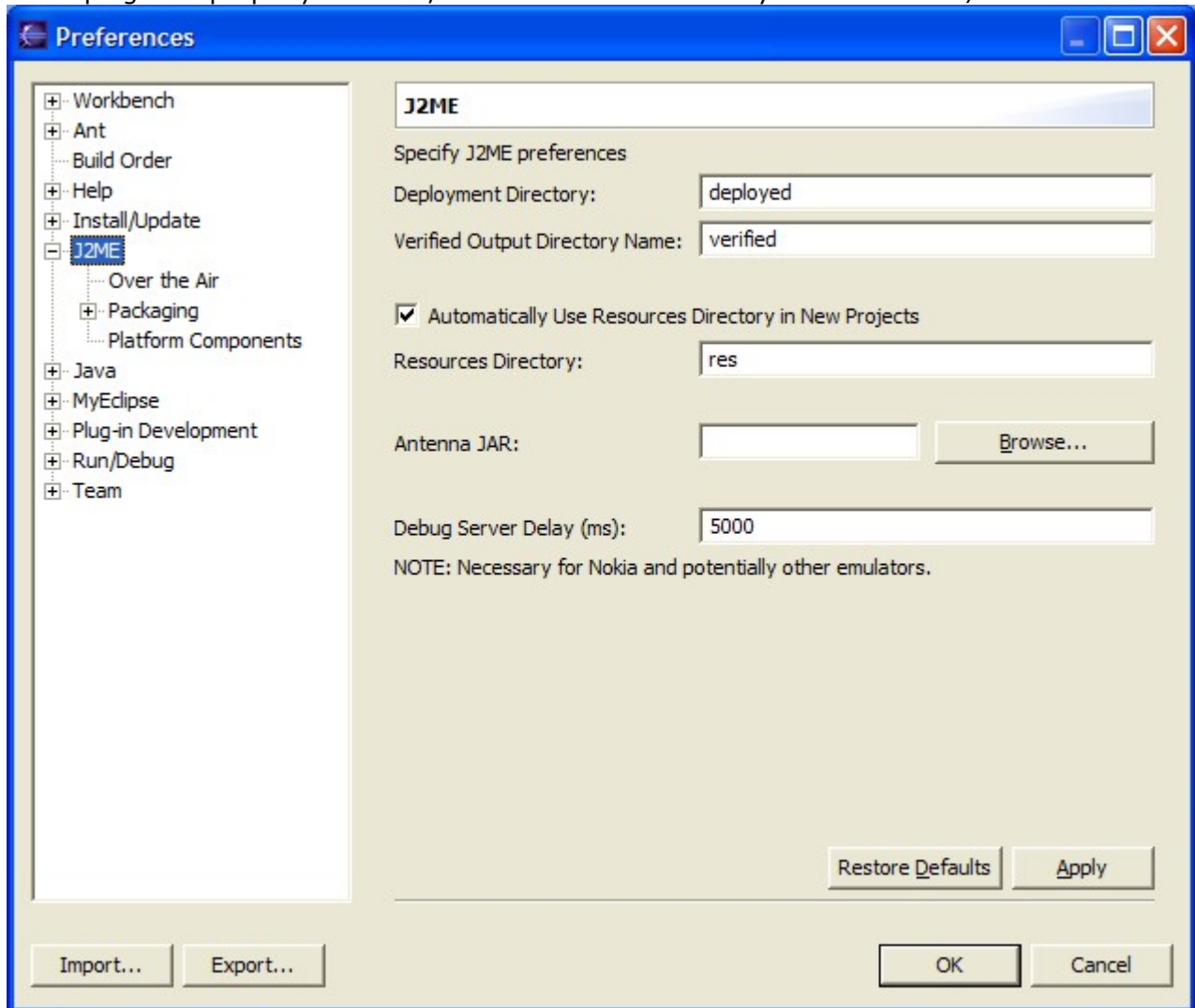
Once you have selected a site, press **Finish**.

9. At present the EclipseME package is not digitally signed. (Maintaining the keys required to digitally sign JAR files costs \$400+/year. If anyone is interested in funding EclipseME to this extent, we'll be happy to sign the JAR files.) As a result, the following warning dialog will be displayed:



shown during new installations suggests that it may be possible to continue without restarting, you should definitely restart Eclipse at this point.

12. Once Eclipse restarts, the installation procedure is complete.
13. In the vast majority of the cases, the installation procedure above will properly handle updating a previous version of EclipseME.
14. Under some rare circumstances, even after Eclipse restarts, EclipseME will not end up properly "registered" inside Eclipse. This can generally be corrected by closing Eclipse and then restarting Eclipse adding the `-clean` parameter to the other parameters you normally use. Using the `-clean` option forces Eclipse to rescan and update all its plugin information.
15. If the plug-in is properly installed, there will be a *J2ME* entry in the *Window / Preferences* dialog.

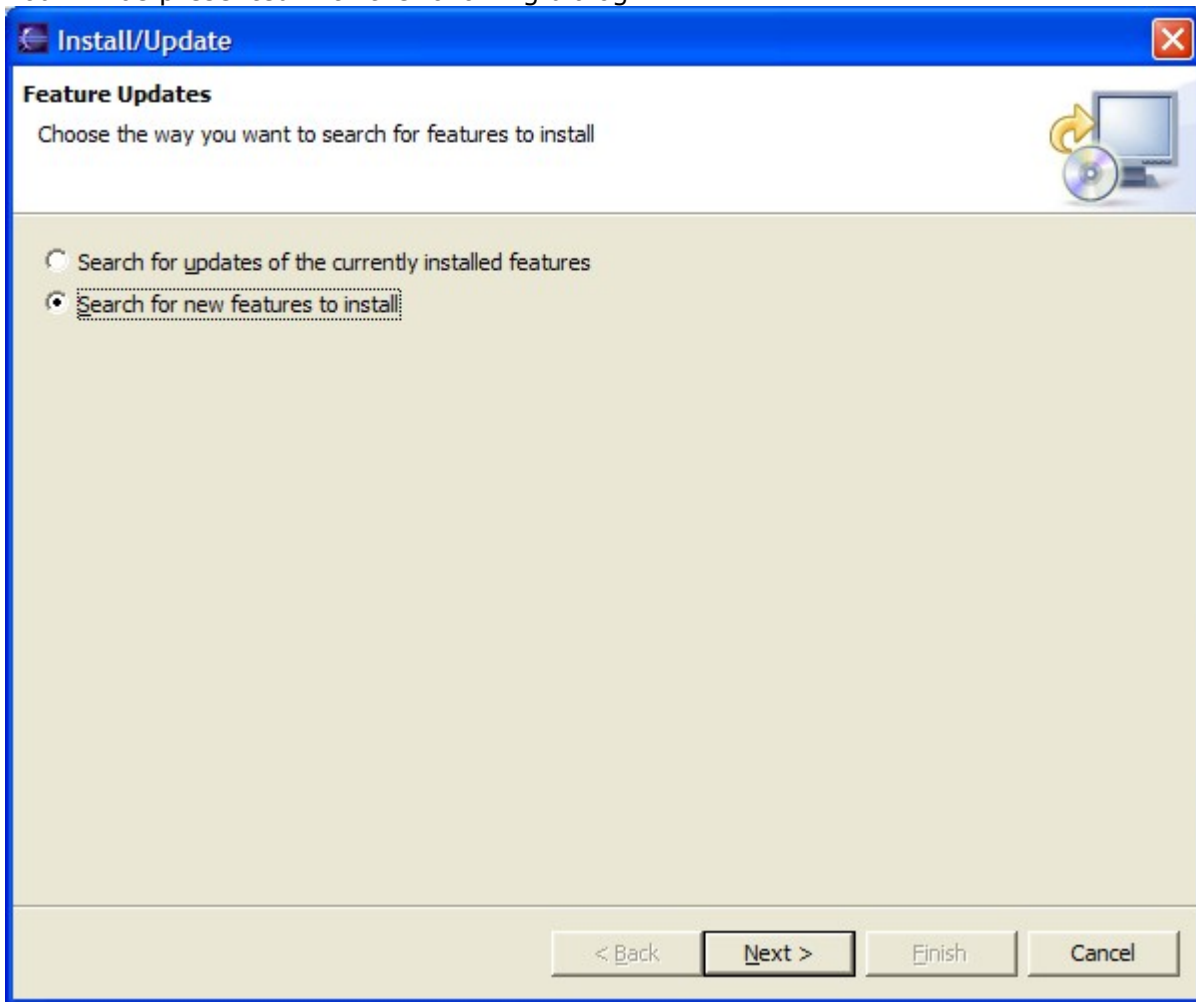


## **Installing via a downloaded archive site!!!**

The process of installing EclipseME via a downloaded archive file is very similar to that of installing it via the EclipseME update site.

1. Download the latest archive site from the [EclipseME downloads page](#).
2. From the Eclipse *Help* menu, select *Software Updates* and then *Find and install...*

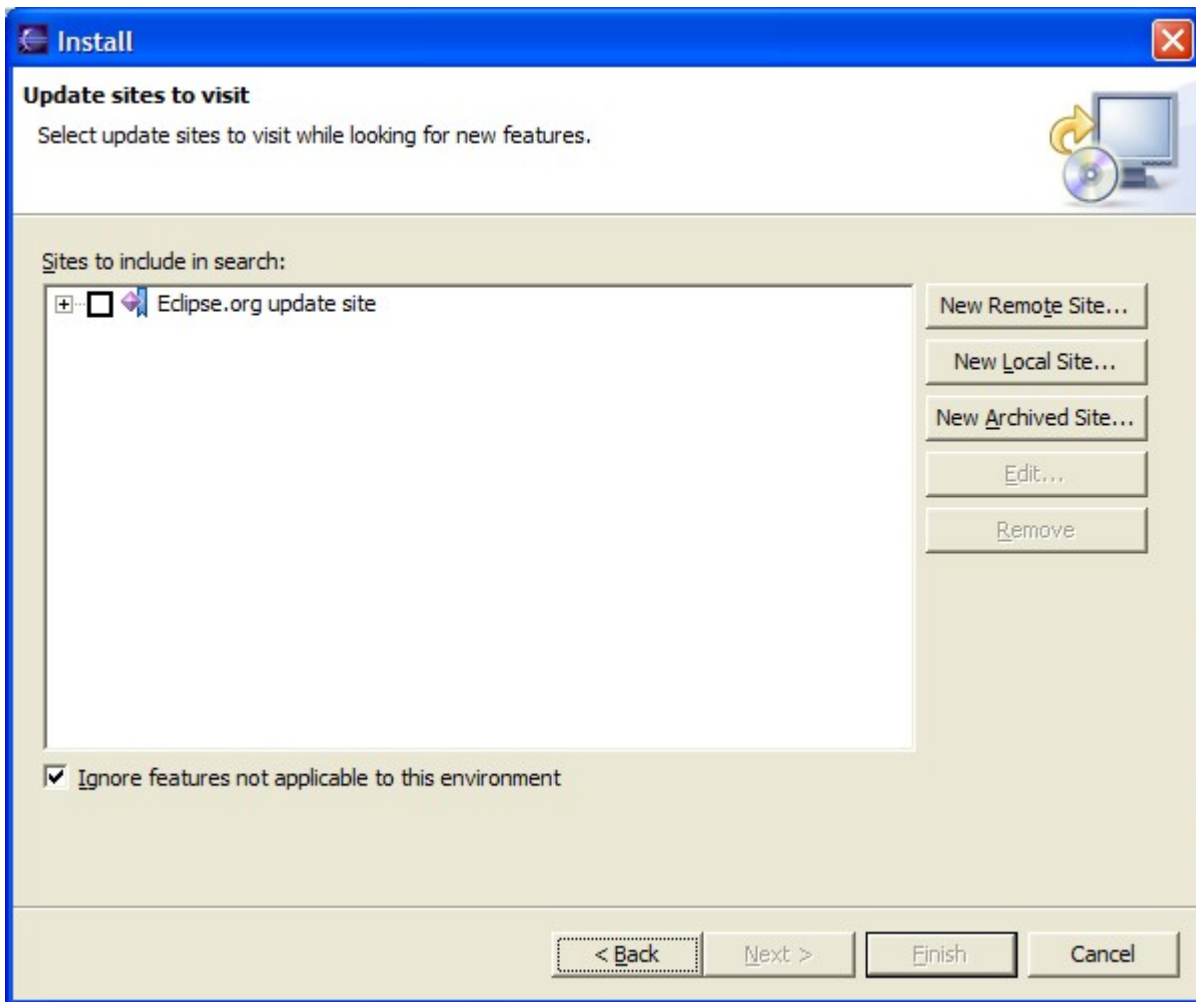
3. You will be presented with the following dialog:



Select the **Search for new features to install** radio button. You should select this option even if you are updating EclipseME.

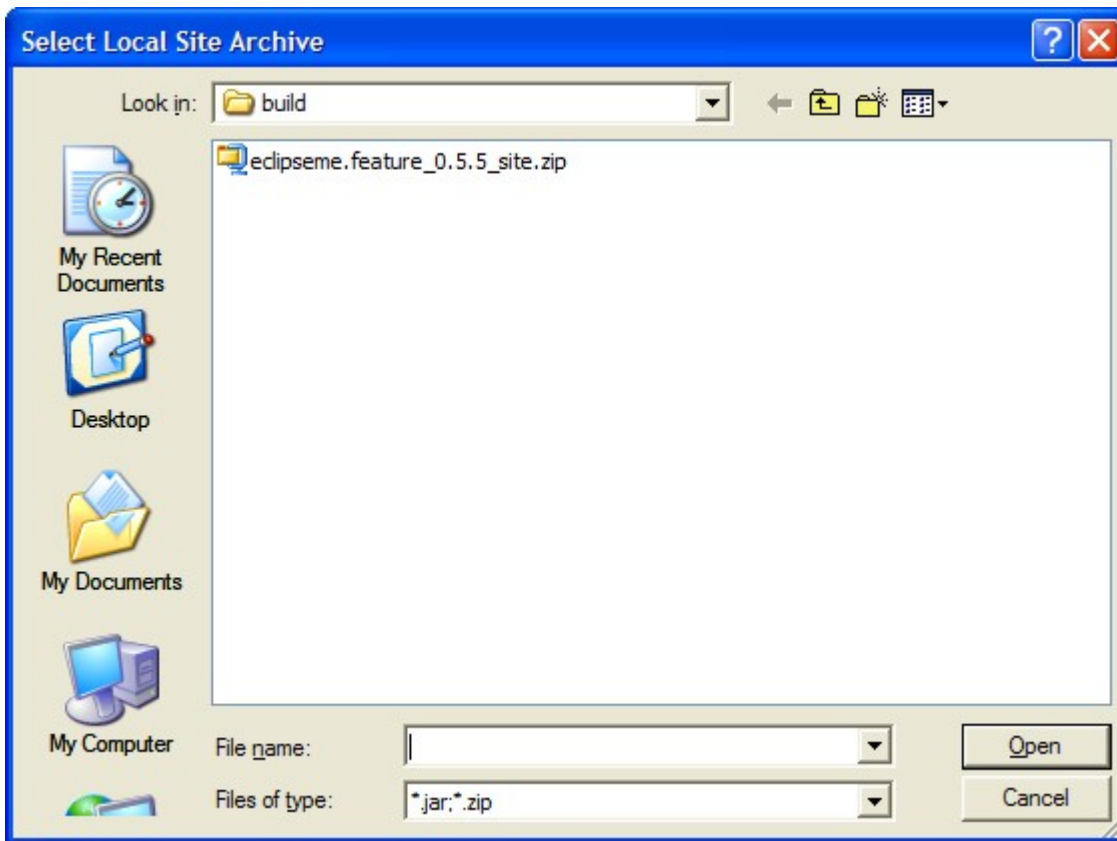
Press **Next**.

4. You will next be presented with the following dialog:

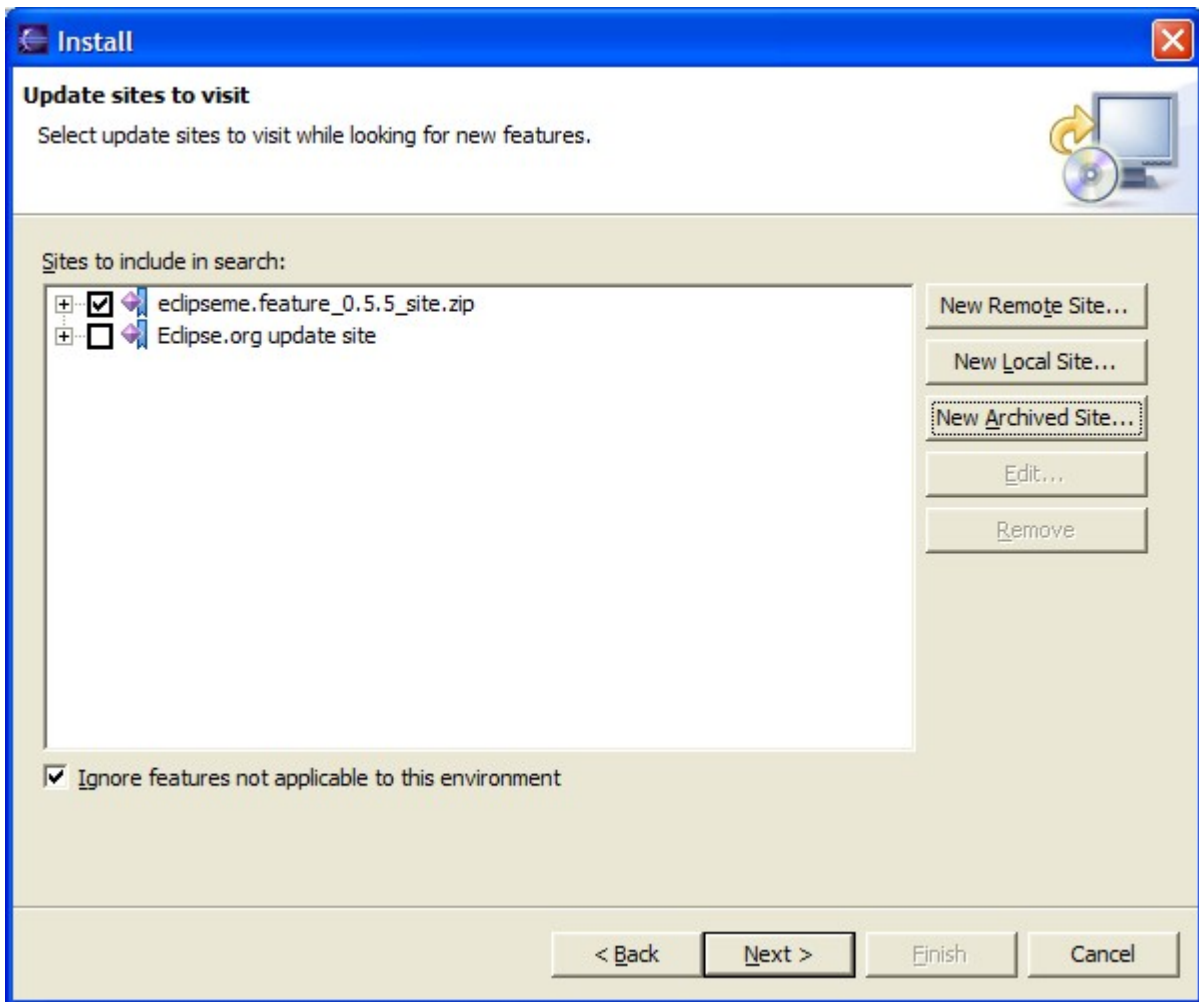


Press the **New Archived Site...** button.

5. In the **Select Local Site Archive** dialog that appears, select the distribution ZIP file and press **Open**.

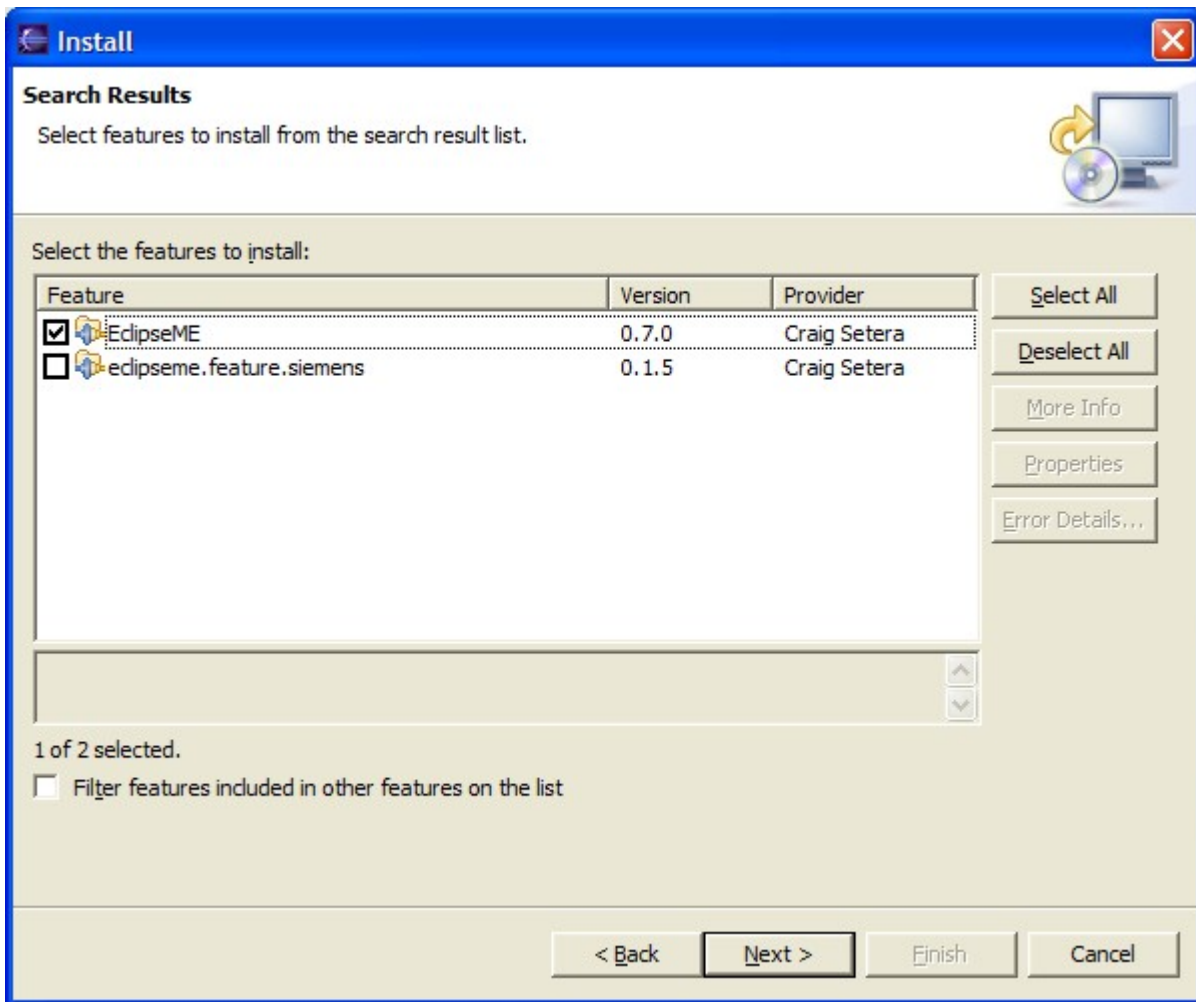


6. The site archive file will now be listed in the **Install** dialog.



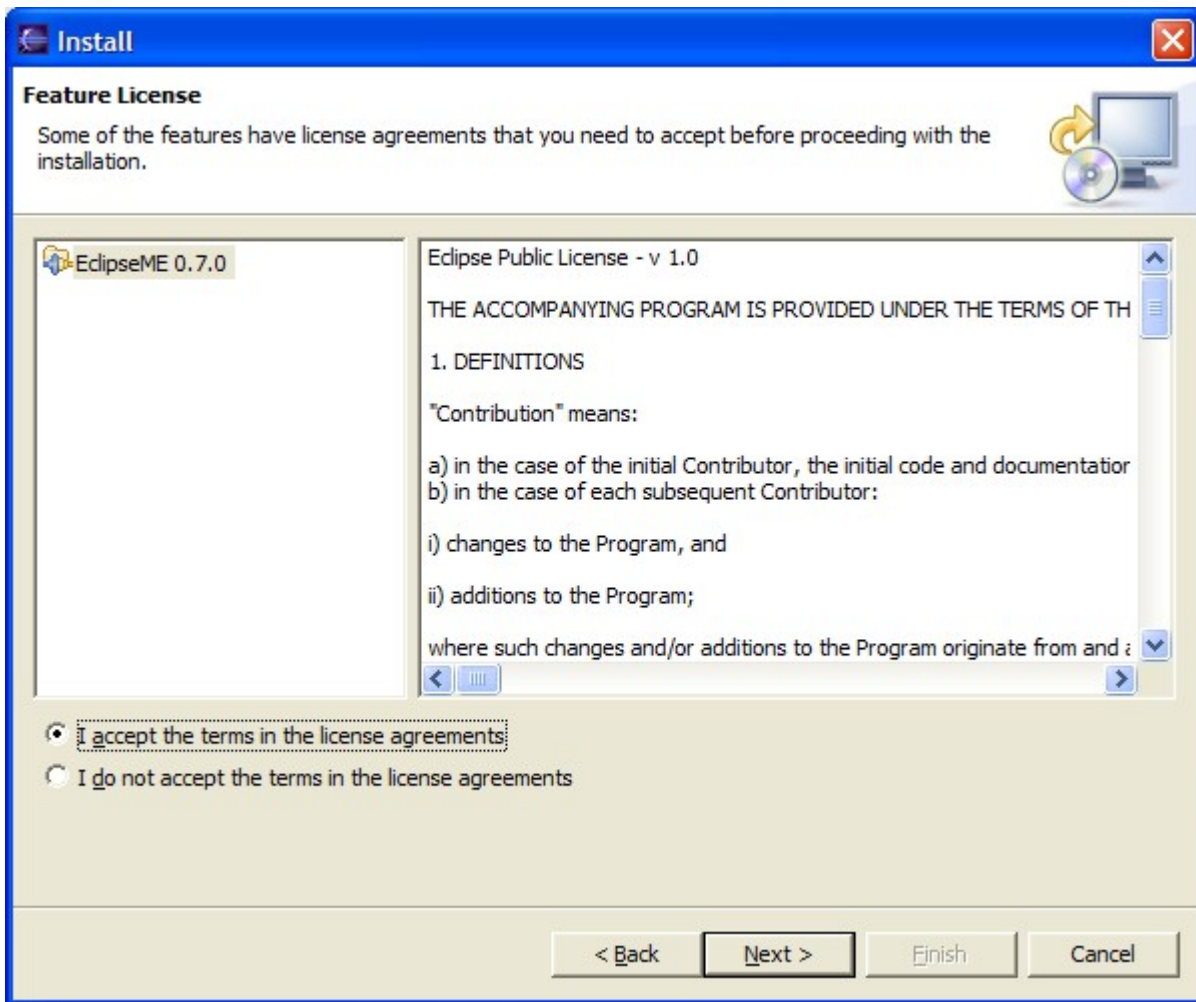
Ensure that there is a check next to the distribution file, then press **Next**.

7. You will next be presented with the following dialog:



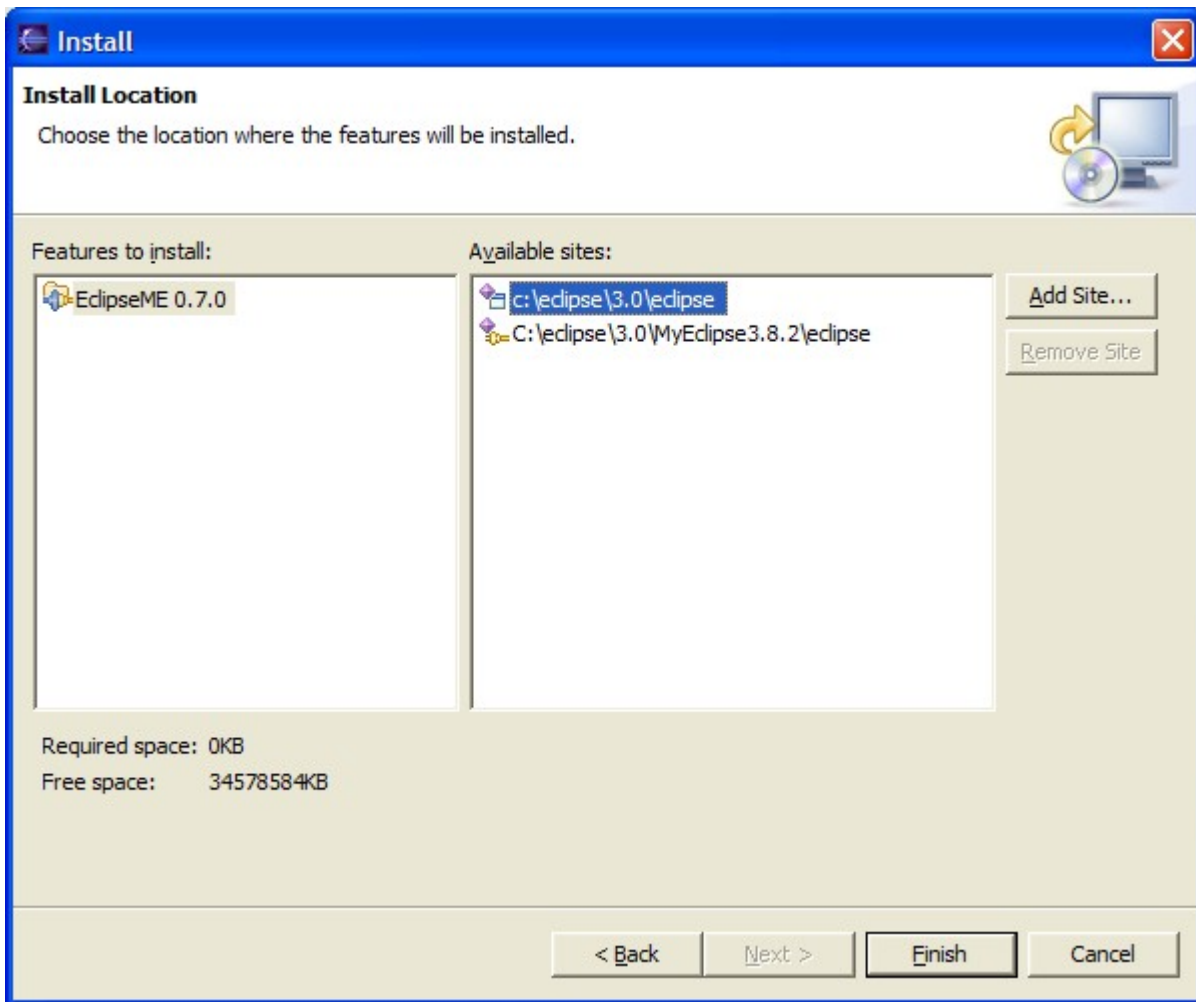
Ensure that there is a check next to **EclipseME**. Other available features can be installed by checking them as well. Press **Next**.

8. The installation process will next display the license agreement for EclipseME.



Select **I accept the terms of the license agreements** and press **Next**.

9. The next dialog that appears displays the possible locations into which you can install EclipseME.

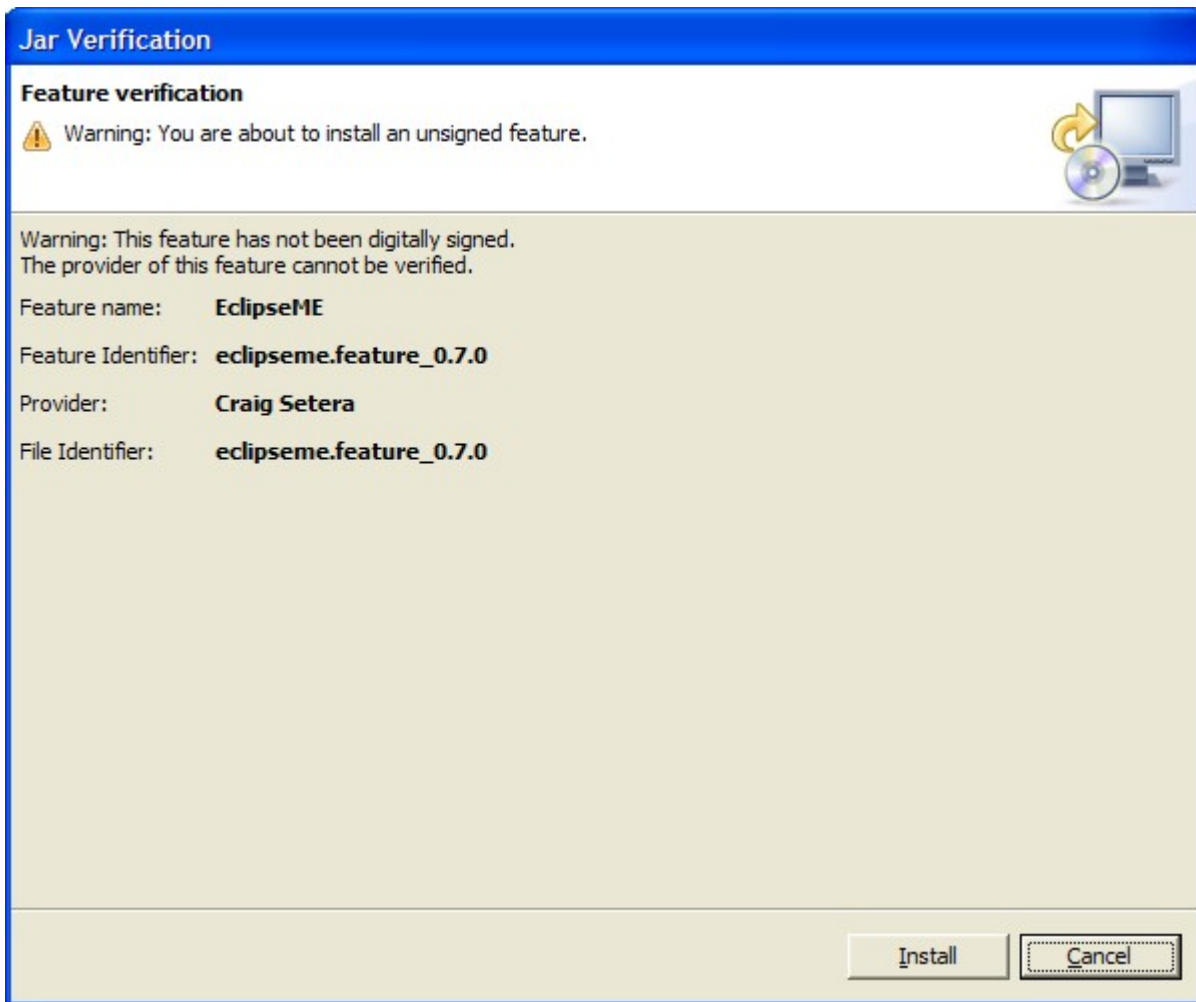


In many cases, the only site that will be listed will be the main Eclipse installation directory. If you have other plugins or features installed, however, you may see additional sites.

Although you may install EclipseME elsewhere, we recommend that you install it in your main Eclipse installation directory.

Once you have selected a site, press **Finish**.

10. At present the EclipseME package is not digitally signed. (Maintaining the keys required to digitally sign JAR files costs \$400+/year. If anyone is interested in funding EclipseME to this extent, we'll be happy to sign the JAR files.) As a result, the following warning dialog will be displayed:

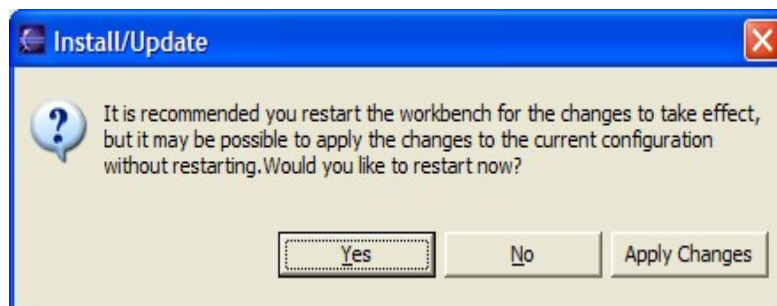


Presuming that you are certain that you obtained the EclipseME distribution file from a reputable source (such as directly downloading it yourself from SourceForge.net), it should be safe to trust the installation package.

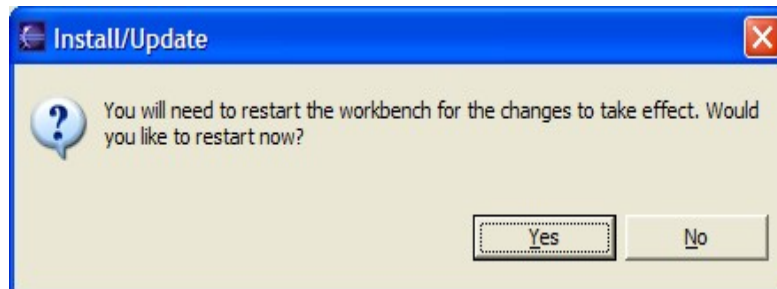
If you choose to continue, press the **Install** button.

11. At this point, Eclipse will begin to install EclipseME from the site archive file. When the installation is complete, you will see one of the following two dialogs:

**New  
installatio  
ns**



**Updates**

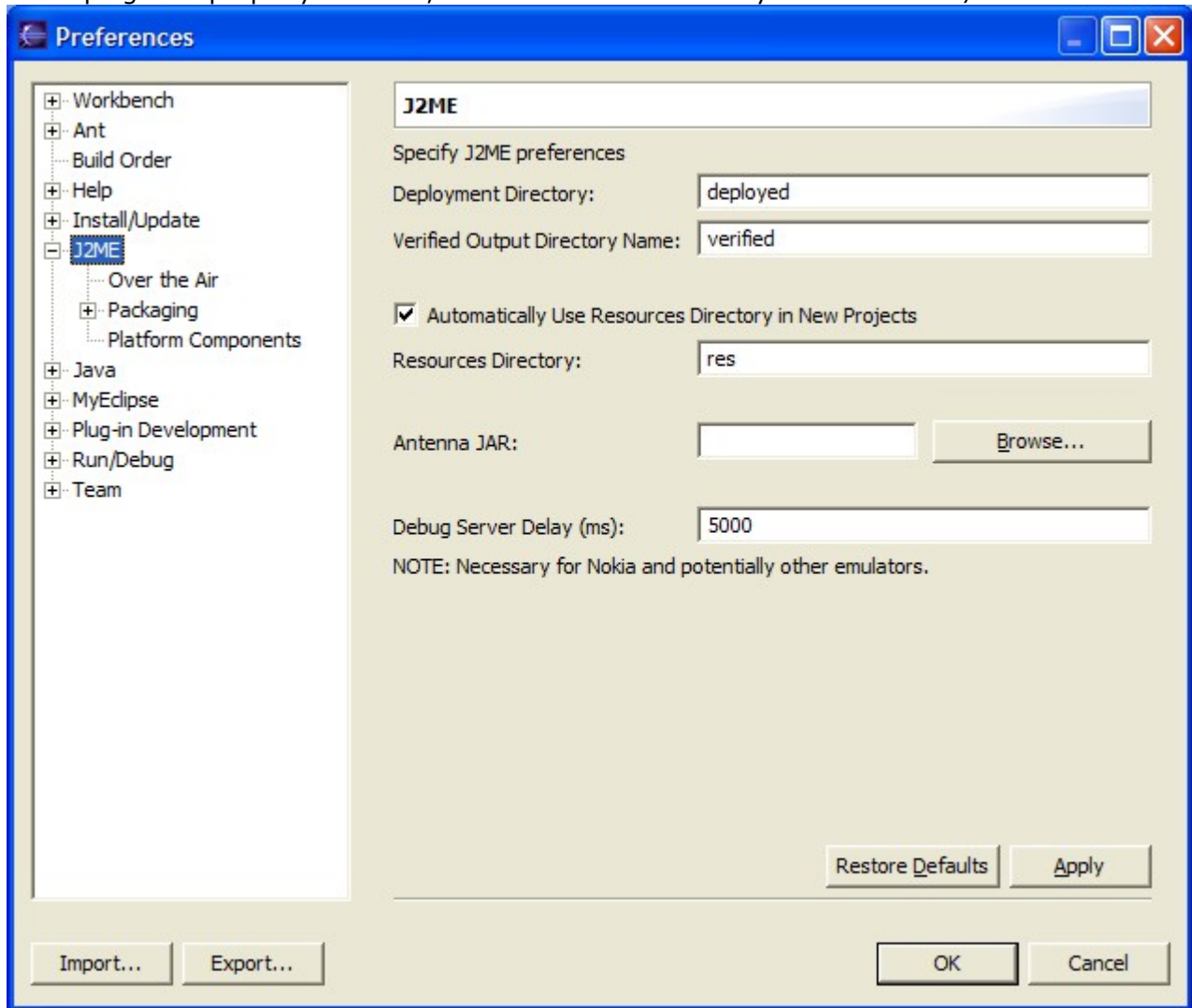


- 12.

Even though the dialog

shown during new installations suggests that it may be possible to continue without restarting, you should definitely restart Eclipse at this point.

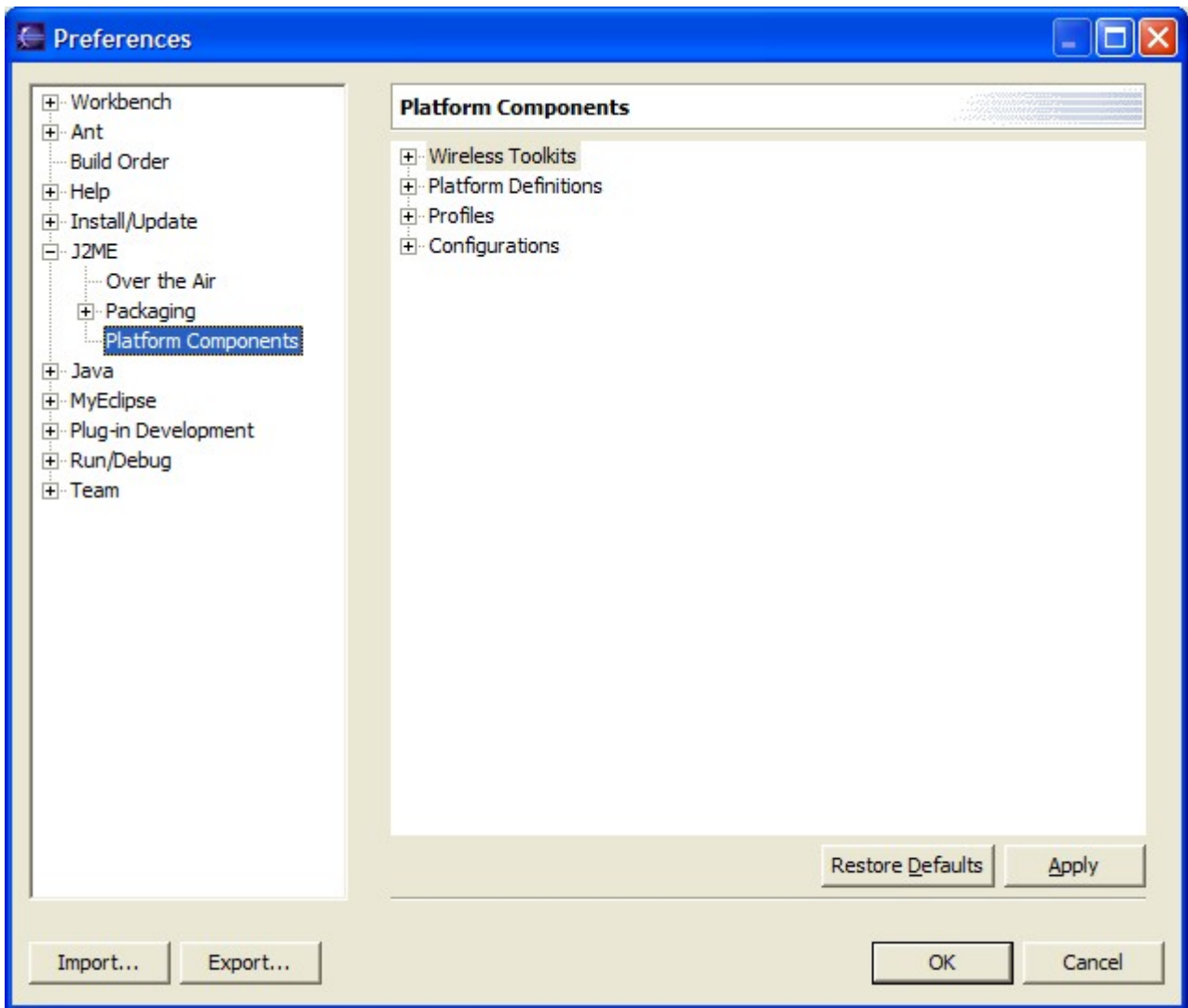
13. Once Eclipse restarts, the installation procedure is complete.
14. In the vast majority of the cases, the installation procedure above will properly handle updating a previous version of EclipseME.
15. Under some rare circumstances, even after Eclipse restarts, EclipseME will not end up properly "registered" inside Eclipse. This can generally be corrected by closing Eclipse and then restarting Eclipse adding the `-clean` parameter to the other parameters you normally use. Using the `-clean` option forces Eclipse to rescan and update all its plugin information.
16. If the plug-in is properly installed, there will be a *J2ME* entry in the *Window / Preferences* dialog.



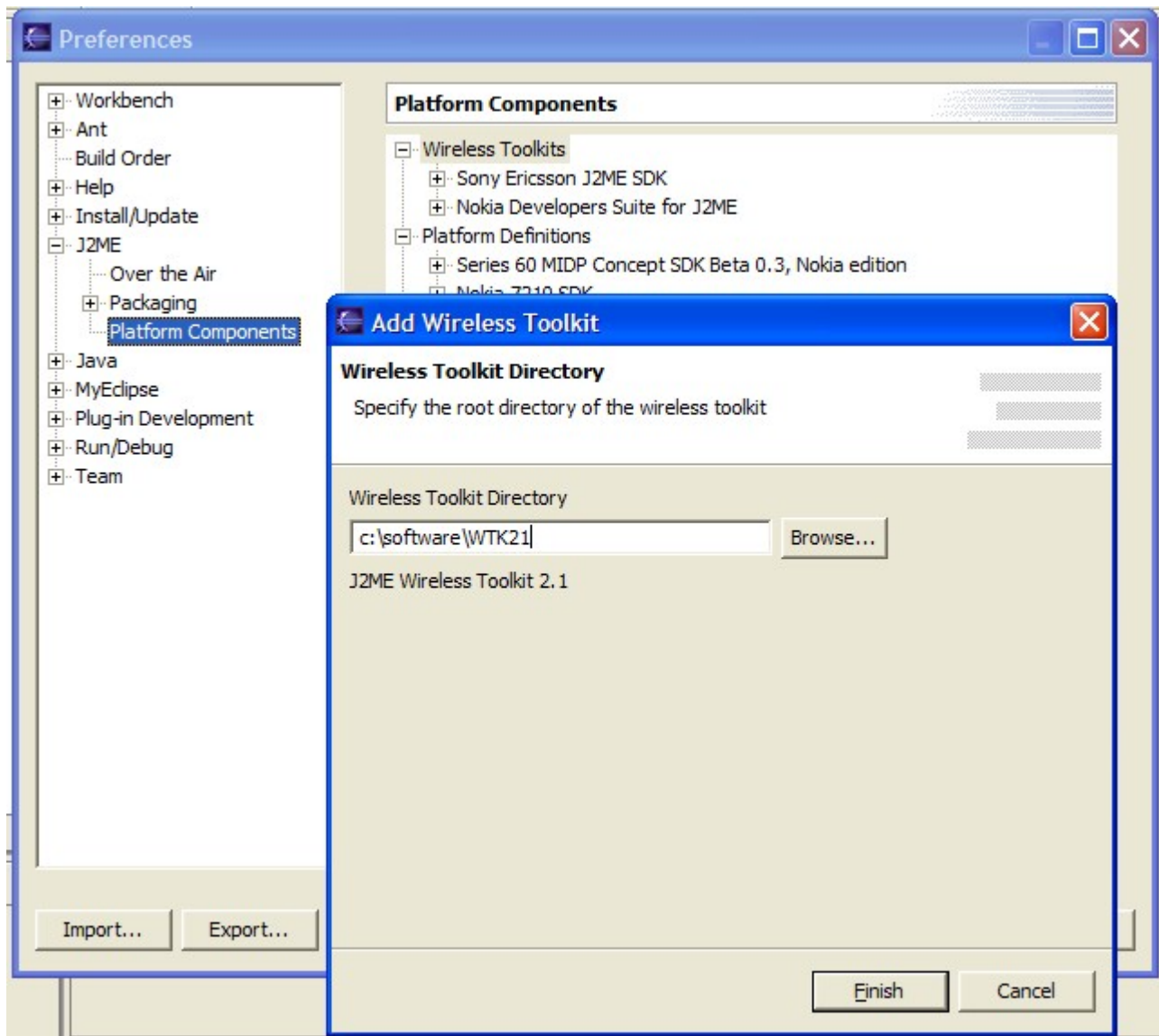
### 3. Configure the Plug-In's Wireless Toolkit Settings

In order to use the plug-in, you must configure at least one Wireless Toolkit. In order to do this, perform the following steps:

1. Select the *Preferences* menu item from Eclipse's *Window* menu.
2. Expand the *J2ME* item in the pane to the left and click on *Platform Components*.



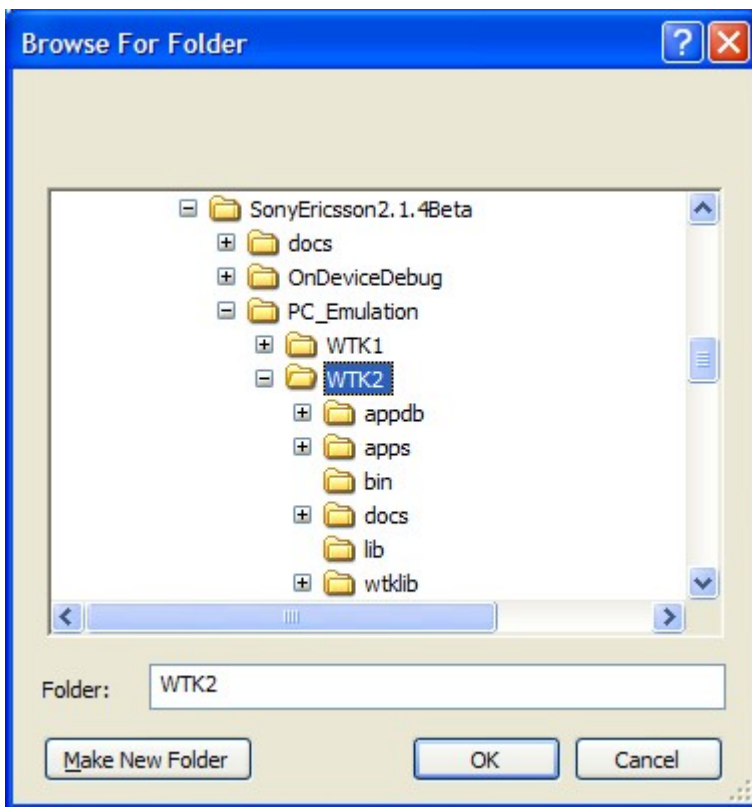
3. Right-click on *Wireless Toolkits* and select *Add Wireless Toolkit*.
4. In the resulting dialog, select the root directory of the wireless toolkit installation.



If you're adding a Wireless Toolkit for which EclipseME has explicit support (or for certain Universal Emulator Interface (UEI) compatible WTK's) once you have selected the root directory of the WTK installation, EclipseME should be able to identify the toolkit, and you can press the *Finish* button to complete the operation.

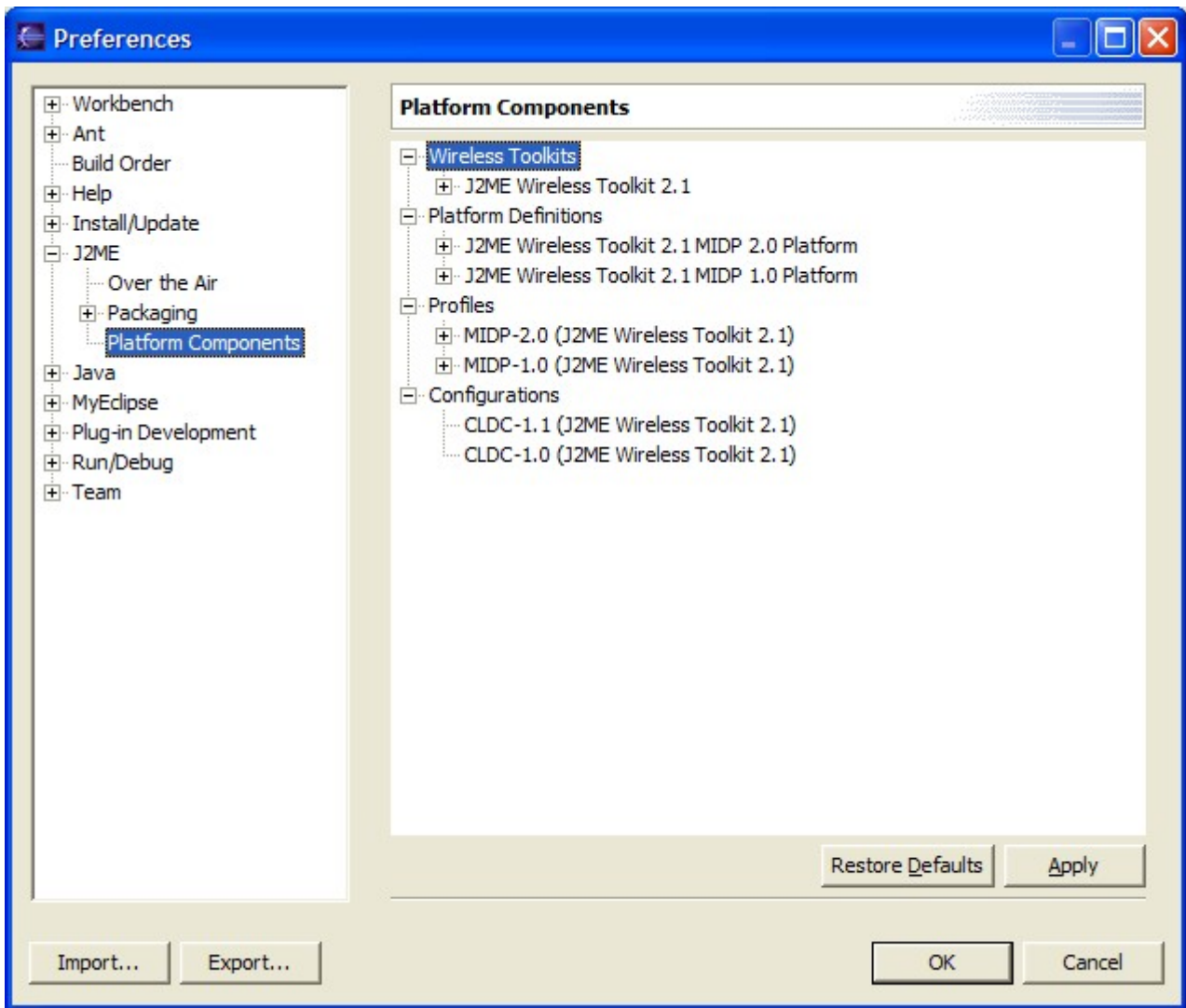
If you select the root install directory for your WTK and EclipseME responds with an "Invalid wireless toolkit root," then you are attempting to add a WTK for which EclipseME does not provide direct (native) support, and for which it could not locate Universal Emulator Interface (UEI) information. The UEI interface allows EclipseME to query the emulator to determine the required settings, classpaths, etc. EclipseME assumes that the emulator is located in a directory named **bin**. This is usually, but not always, located under the root of the WTK installation area. If it is, EclipseME will find it automatically when you select the WTK root directory.

If the **bin** directory is not directly under the root of the WTK installation, then you need to help EclipseME locate it so that EclipseME can interrogate the emulator. Find the **bin** directory, and select that directory's *parent* in the Browse dialog. Thus, in the example below, the **bin** directory was nested several layers deep. Selecting **WTK2** (**bin**'s parent) allows this toolkit to be recognized.



If EclipseME persists in showing the "Invalid wireless toolkit root" error even when you have selected the parent of the **bin** directory, then either the emulator has a non-standard name, or the emulator does not support the UEI standard. In either case, EclipseME can not support it for the time being. In this case, please feel free to submit an [RFE](#) to ask for support for this WTK to be added.

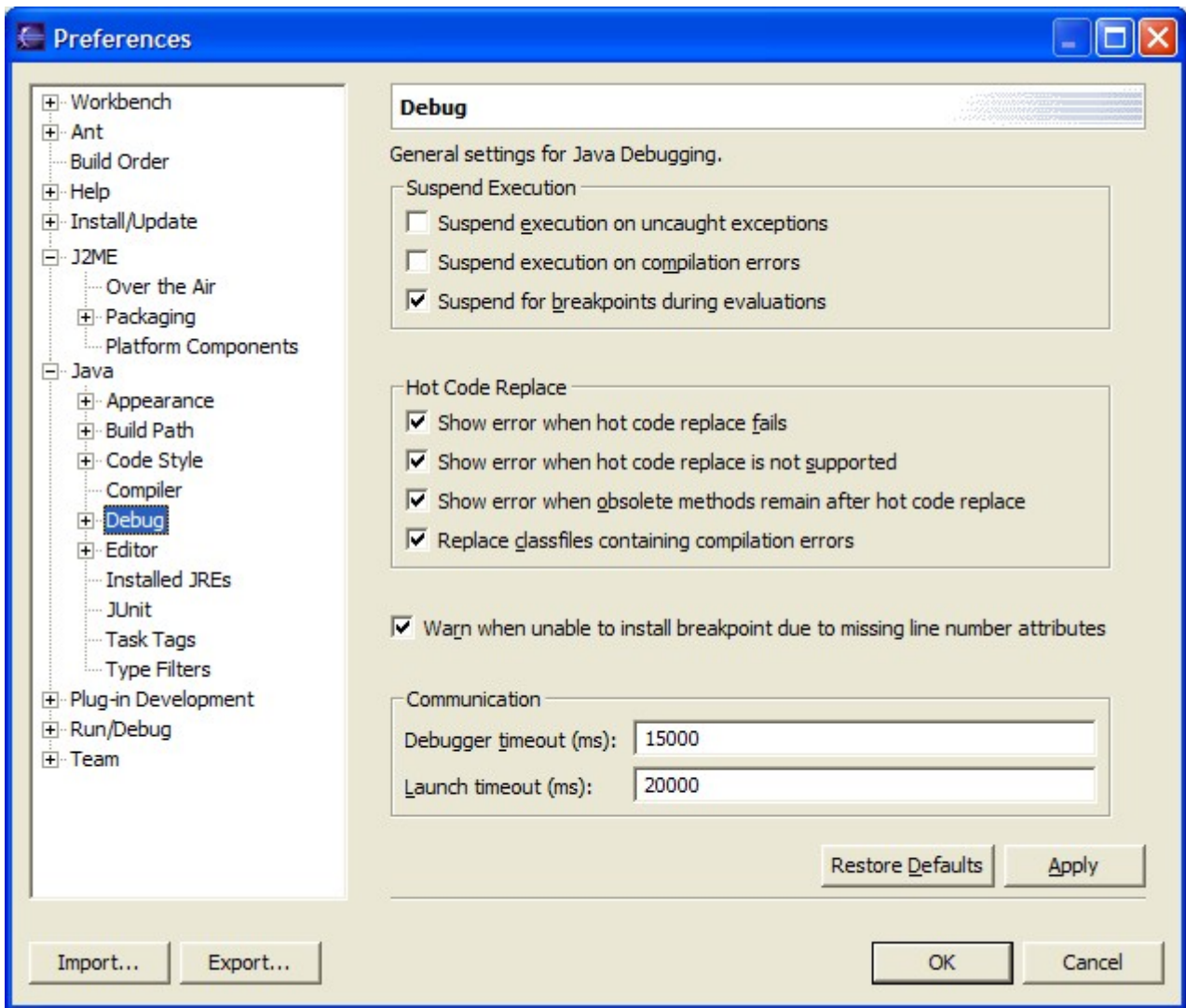
5. When you have successfully completed adding the toolkit, you should be able to expand the *Wireless Toolkits* and other items in the right-hand pane and see the toolkits, definitions, profiles and configurations that are installed.



#### 4. Change Eclipse's Debug Settings

Because of some quirks in the wireless toolkits, Sun's in particular, if you are going to debug your MIDlet using Eclipse, you must change several of the default debug settings. To do this:

1. Select the *Preferences* menu item from Eclipse's *Window* menu.
2. Expand the *Java* item in the left pane and click on the *Debug* entry.
3. Ensure that both *Suspend execution on uncaught exceptions* and *Suspend execution on compilation errors* near the top of the dialog are **NOT** checked.
4. Increase the *Debugger timeout* near the bottom of the dialog to at least 15000 ms.
5. The resulting settings should look something like this:

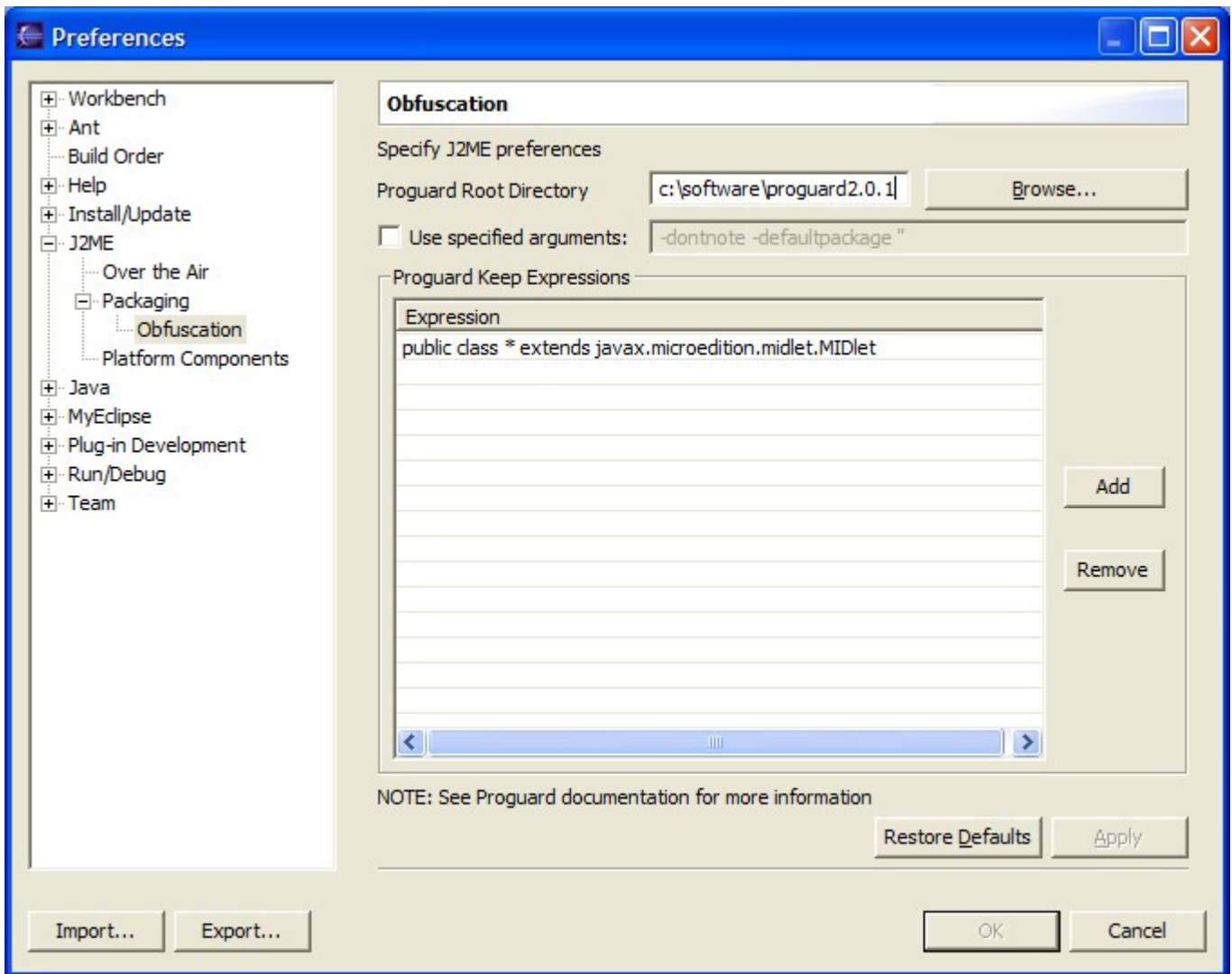


If you do not make these changes you will get errors when you try to run your MIDlet.

## 5. Configure ProGuard (Optional)

If you will be using ProGuard to produce obfuscated packages, you will need to configure it into the plug-in. To do this:

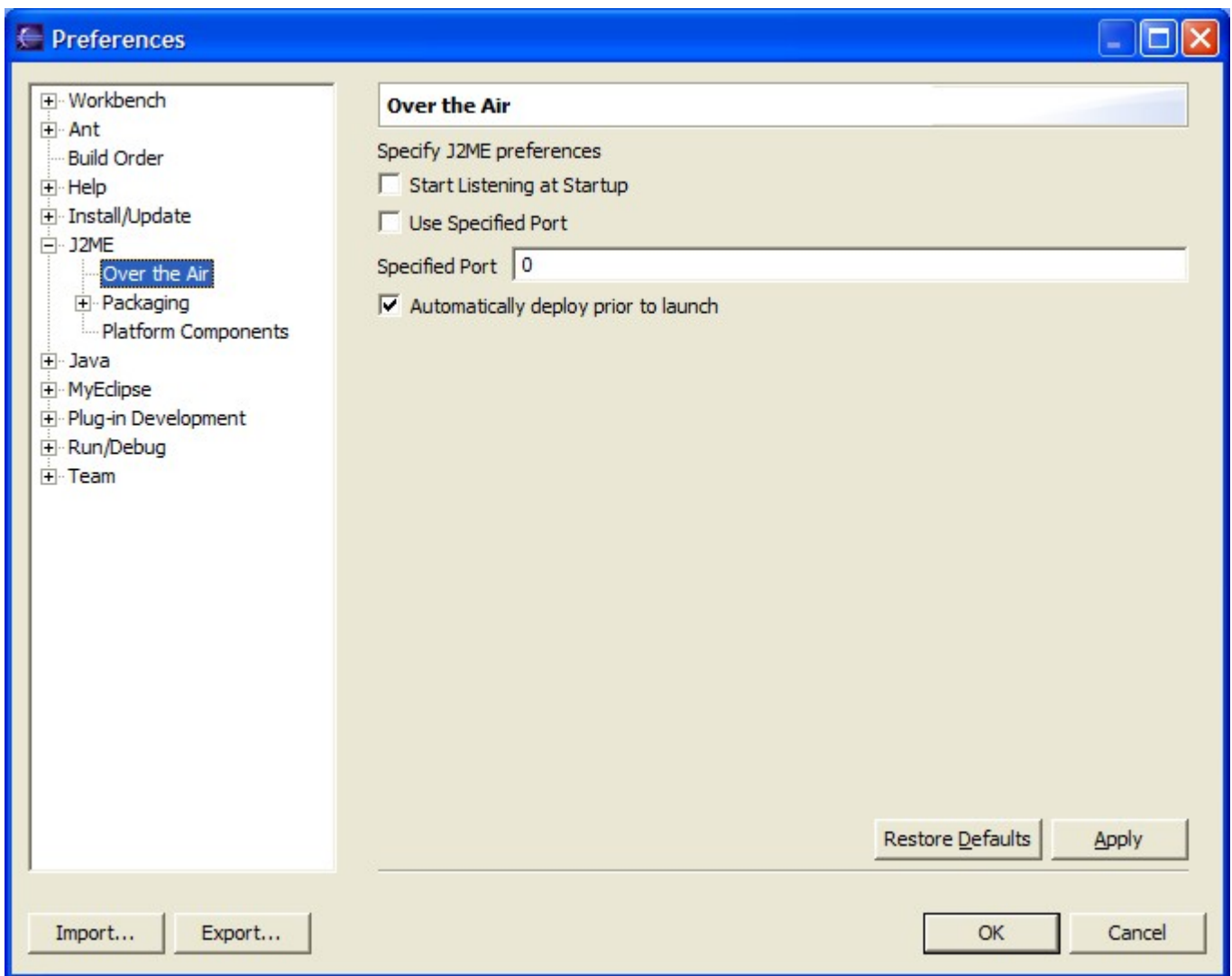
1. Select the *Preferences* menu item from Eclipse's *Window* menu.
2. Expand the *J2ME* item in the left pane and click on the *Obfuscation* entry.
3. Configure the ProGuard Root Directory near the top of the dialog.
4. Configure any other ProGuard-specific settings that you need. For more information on ProGuard, see the [ProGuard SourceForge site](#).
5. The resulting settings should look something like this:



## 6. Configure EclipseME's Over The Air (OTA) options (Optional)

If you will be using EclipseME to debug your MIDlet in Over The Air (OTA) mode, you may want to adjust the OTA Preferences. To do this:

1. Select the *Preferences* menu item from Eclipse's *Window* menu.
2. Expand the *J2ME* item in the left pane and click on the *Over The Air* entry.
3. The default settings look like this:



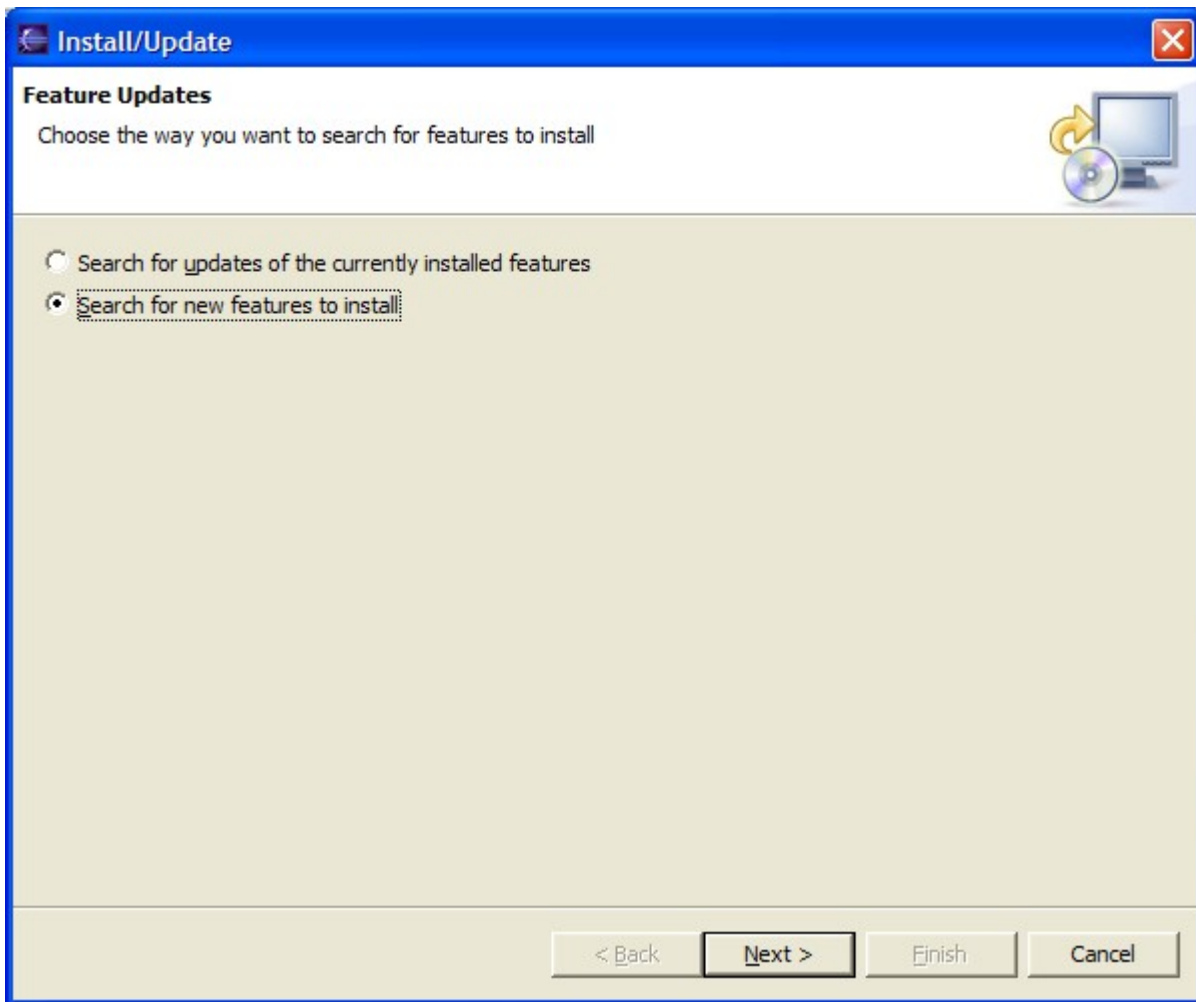
At this point, you are ready to use EclipseME to create MIDlet suites.

## **Updating your EclipseME installation**

Eclipse implements a versioning system that allows you to update features and plugins without having to first remove the older versions. As a result, the process of upgrading to a new version of EclipseME is extremely easy.

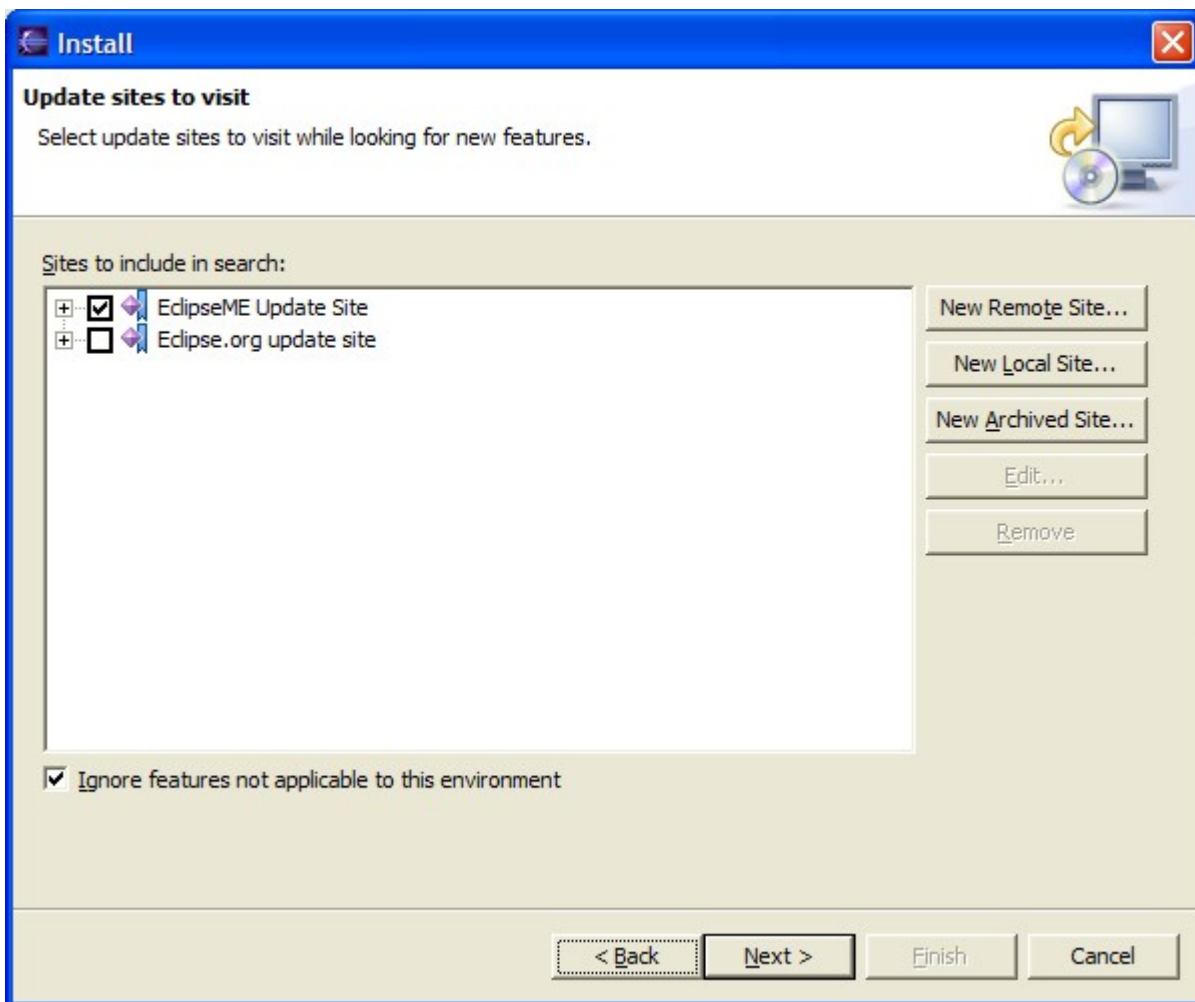
### **If you installed via the EclipseME Update Site**

1. From the Eclipse *Help* menu, select *Software Updates* and then *Find and install...*
2. You will be presented with the following dialog:



Select the **Search for new features to install** radio button.  
Press **Next**.

3. The update site you previously configured will be listed in the **Install** dialog.



Ensure that there is a check next to the EclipseME update site, then press **Next**.

From this point, the process is identical to the original installation procedure outlined [above](#).

Note that updating EclipseME via the *Search for updates of currently installed features* does not work at this time. You need to follow the procedure above to update directly from the SourceForge site. We anticipate correcting this in the future.

### **If you installed from a downloaded archive site**

The process of upgrading is identical to the process of initial installation. Simply download the new archive site file and go through the process [above](#) using that file. If you wish, before upgrading using the new file, you may use the **Remove** button to remove the old archive file from the installer.

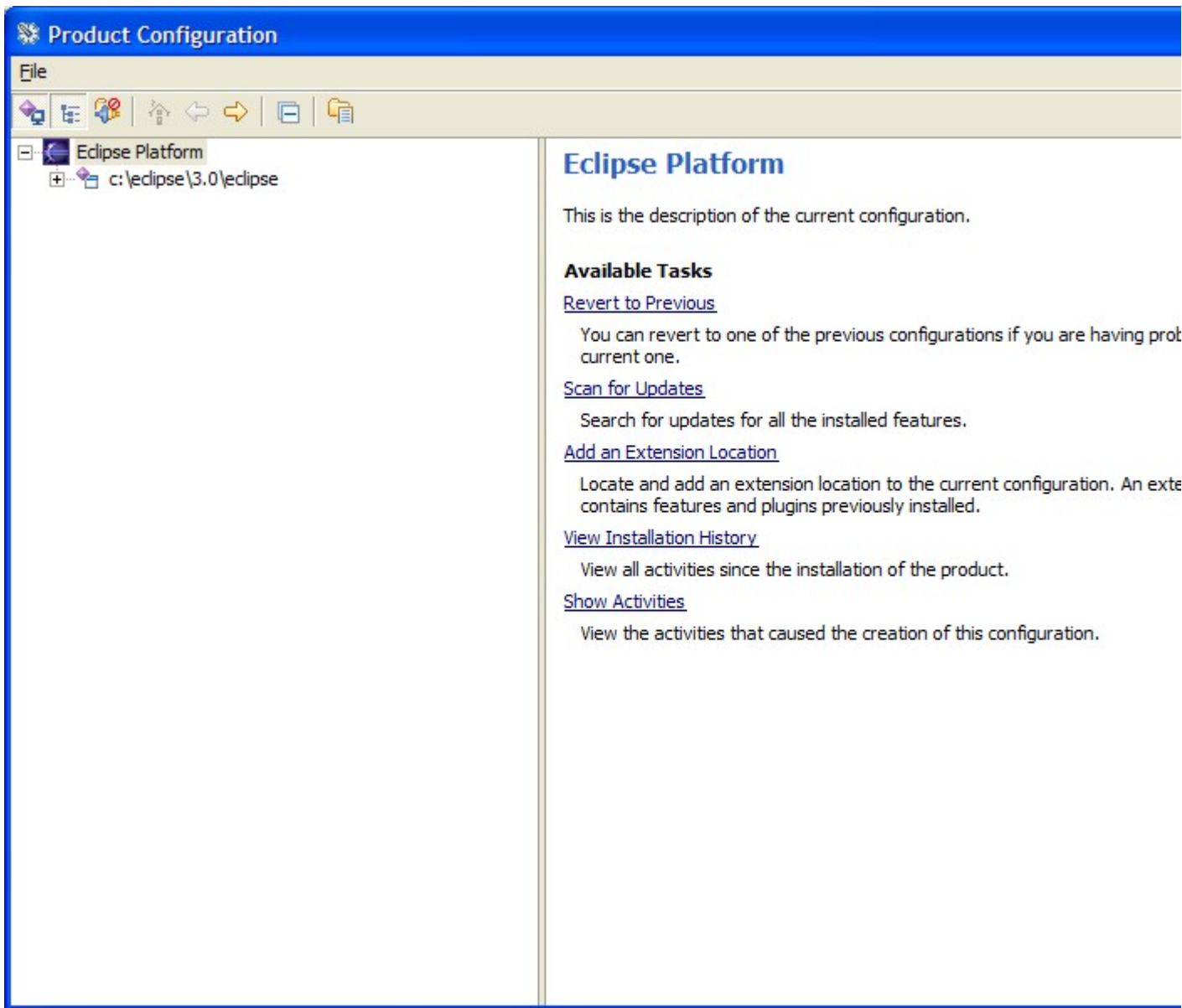
### **Removing EclipseME**

Obviously we hope that you find EclipseME so useful that you will never want to be without it. If you really feel you need to remove it, however, or if you want to do a completely "clean" install, you can follow these steps to remove EclipseME from your Eclipse installation:

#### **The "Normal" way**

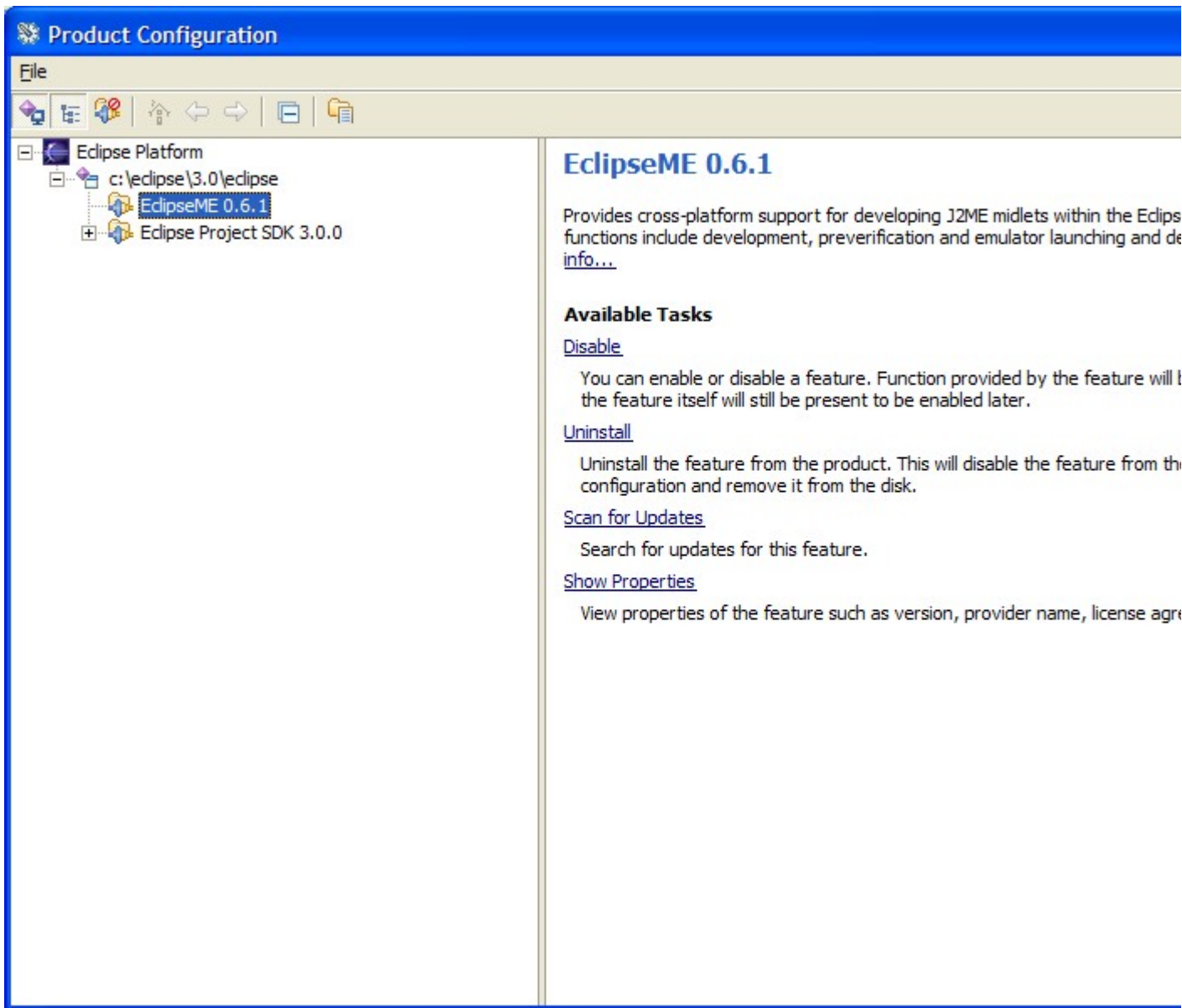
The most straightforward way to remove EclipseME is to use Eclipse's configuration management.

1. From the Eclipse *Help* menu, select *Software Updates* and then *Manage Configuration...*
2. You will be presented with the following dialog:



Expand the entry in the left pane for your Eclipse installation directory and click on the entry for EclipseME.

3. At this point, the dialog should look something like this:



Click on the **Uninstall** link in the right pane.

4. Eclipse will ask you to confirm that you want to uninstall EclipseME. Answer "Yes", and EclipseME will be removed. Note that Eclipse will have to restart the workbench after the uninstall so that it can update its feature and plugin metadata.
5. If you have any of the optional EclipseME features (such as the Siemens feature) installed, Eclipse will require that you uninstall them before uninstalling EclipseME itself. In this case, it is OK not to restart the workbench between uninstalling the feature and uninstalling EclipseME itself.

Note that removing EclipseME using this technique will remove the current version of the EclipseME feature and plugins from your Eclipse installation directory. If you updated EclipseME from an older version, the plugin and feature directories for the older versions will still be present inside your Eclipse directory. If you really want to remove all vestiges of EclipseME, you can get rid of those directories using the "Brute Force" method below.

### **The "Brute Force" method**

If the "Normal" method doesn't work, or if you want to make absolutely sure that everything is gone, you can always remove EclipseME the "brute force" way:

1. Close any running copies of Eclipse.
2. Open the `features` directory within your Eclipse installation directory.
3. Delete any directories whose name begin with "eclipseme". (e.g. `eclipseme.feature_0.6.1`)
4. Open the `plugins` directory within your Eclipse installation directory.
5. Delete any directories whose name begin with "eclipseme". (e.g. `eclipseme.core_0.6.1`, `eclipseme.docs_0.6.1`, etc.)
6. Invoke eclipse using the `-clean` command line option. This will force Eclipse to re-scan its lists of installed features and plugins in order to ensure that the appropriate updates to Eclipse's metadata are performed.

## **Removing EclipseME traces from a project**

If you want to remove all traces of EclipseME from a project, you need to do the following from outside Eclipse:

1. EclipseME creates a file named `.eclipseme` in the root of each J2ME project. Remove this file.
2. EclipseME adds a build command and a "nature" to the `.project` file associated with each Eclipse project. A typical Eclipse `.project` file looks like this:
3. `<xml version="1.0" encoding="UTF-8"?>`
4. `<projectDescription>`
5. `<name>PaperClick Java Go Window</name>`
6. `<comment></comment>`
7. `<projects>`
8. `</projects>`
9. `<buildSpec>`
10. `<buildCommand>`
11. `<name>org.eclipse.jdt.core.javabuilder</name>`
12. `<arguments>`
13. `</arguments>`
14. `</buildCommand>`
15. `<buildCommand>`
16. `<name>eclipseme.core.preverifier</name>`
17. `<arguments>`
18. `</arguments>`
19. `</buildCommand>`
20. `</buildSpec>`
21. `<natures>`
22. `<nature>org.eclipse.jdt.core.javanature</nature>`
23. `<nature>eclipseme.core.nature</nature>`
24. `</natures>`
- `</projectDescription>`

The underlined items are the ones you should remove.

# THIS IS WHERE I COME BACK

Just to remember: these are the members of Gang of Four

**j2sdk-1\_4\_2\_08-windows-i586-p.exe**

**eclipse-SDK-3.0-win32.exe**

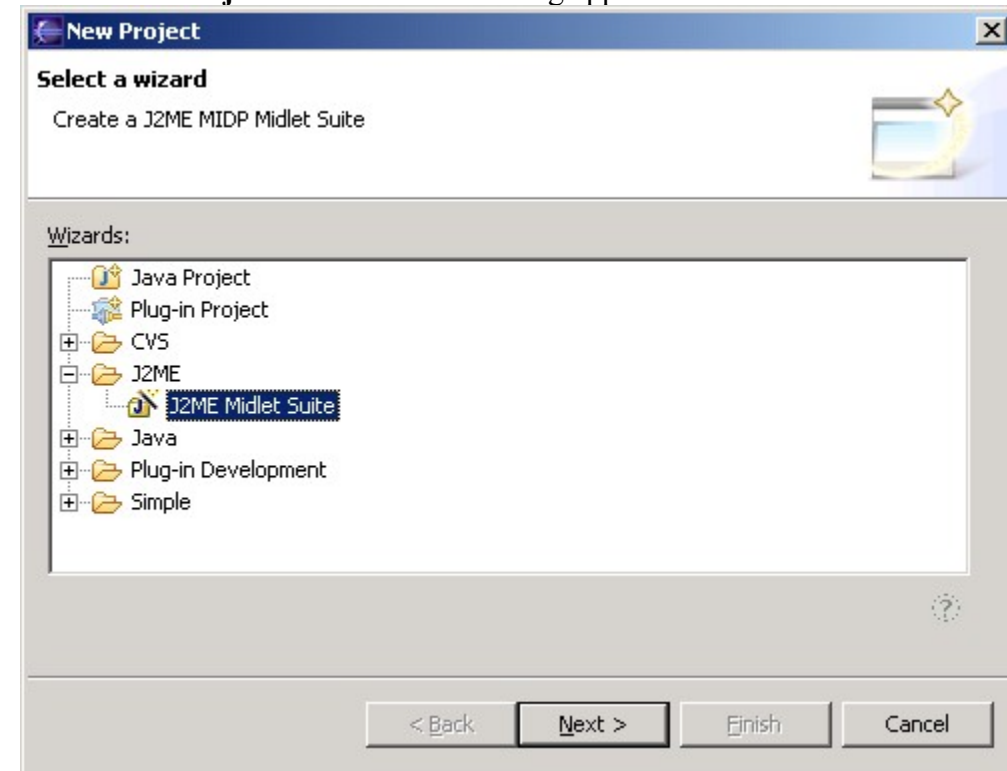
**eclipseme.feature\_0.7.5\_site.zip**

**j2me\_wireless\_toolkit-2\_2-windows.zip**

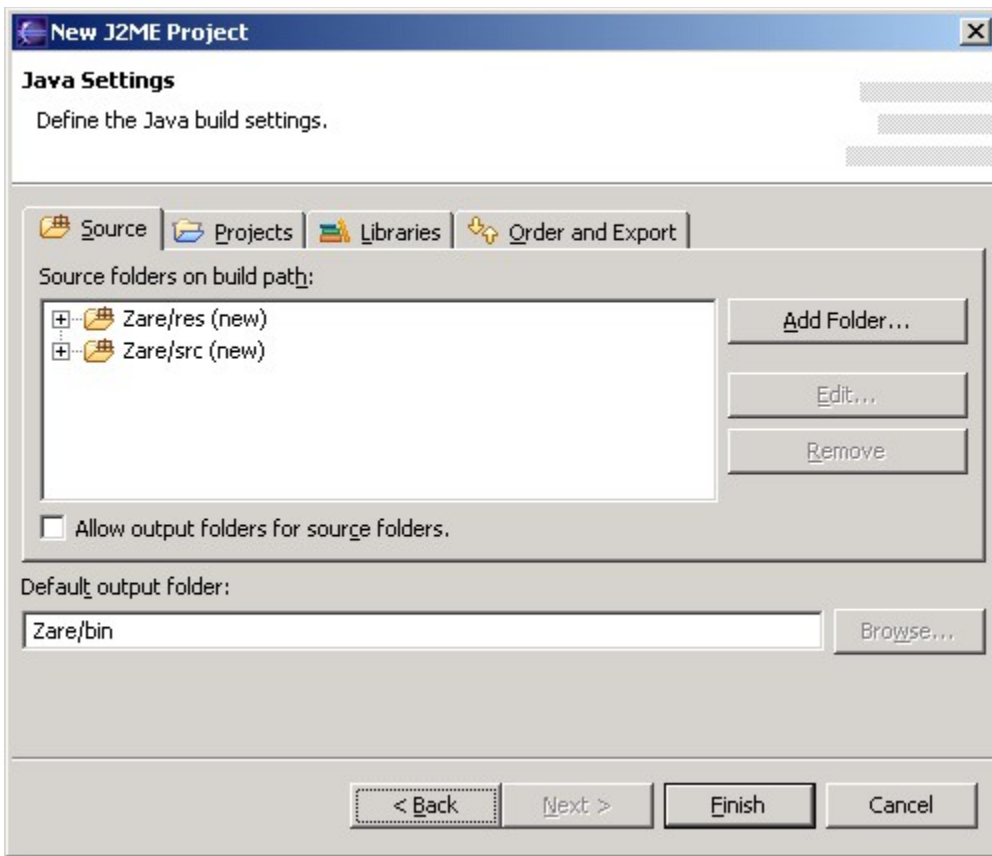
Bravo! I think you made it. Now what? Well we might develop some stuff and run it...

Restart the Eclipse and choose

**File / New /Project...** When this dialog appears choose **J2ME Midlet Suite**



I invented a project called **Zare** (you better call yr project with the same name) and then Eclipse gave me some fancy *nfo* about where and how my **/src** files will be placed. That's exactly what next dialog shows:

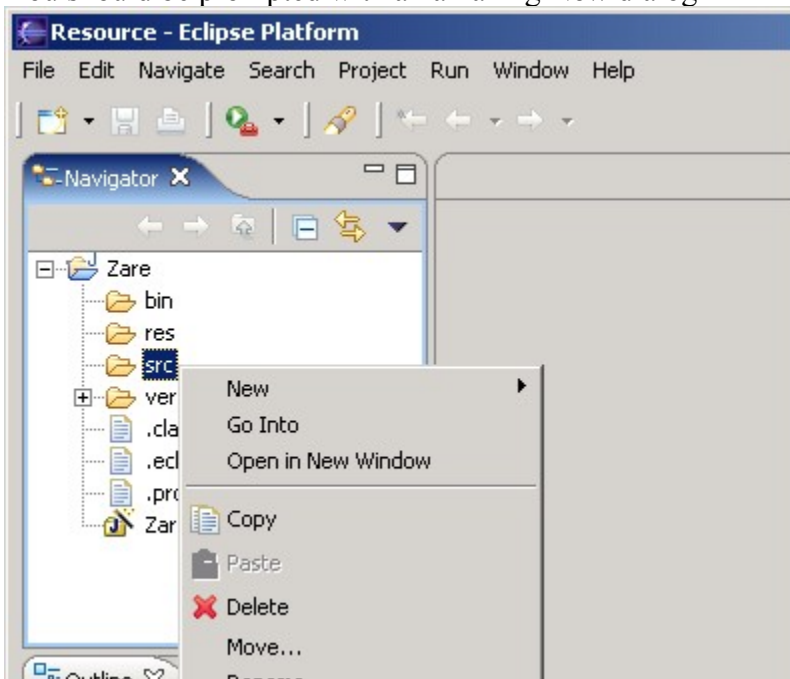


You should get somethin like this. If you didn't, check chapter **0. Before you begin**

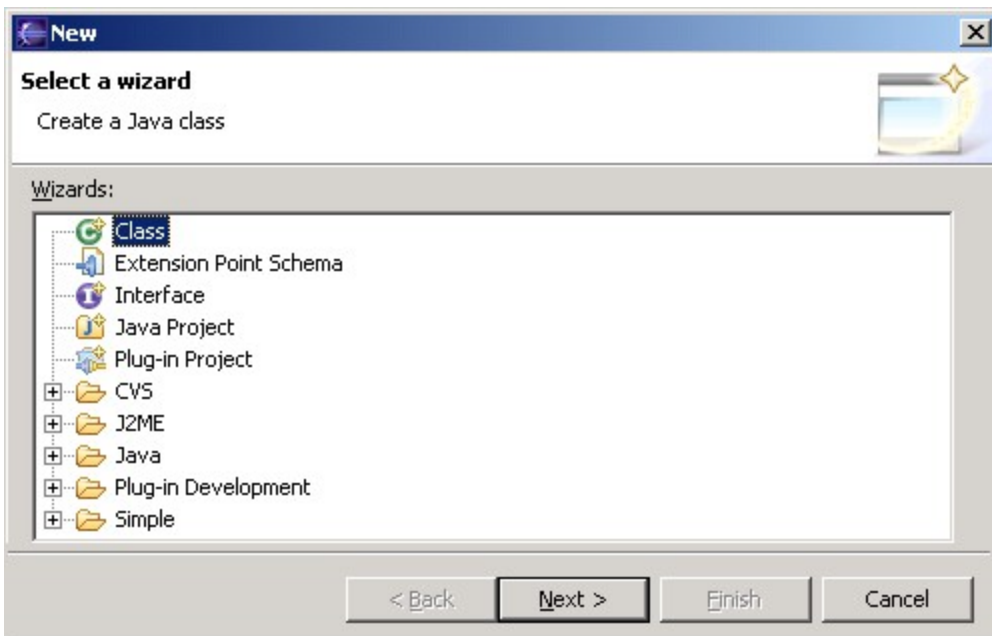
from EclipseME tutorial. You have to have /src and /res folders. Click **Finish**.

Then right-click /src folder.

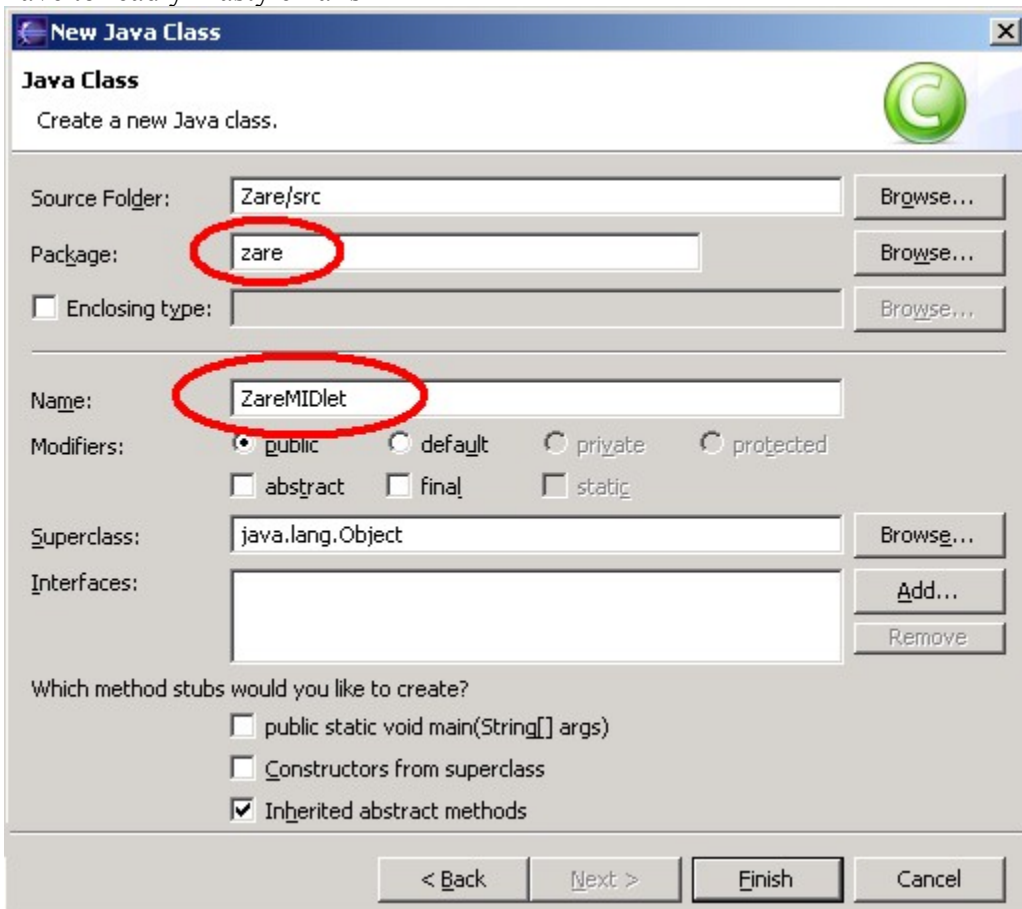
You should be prompted with an amazing **New** dialog



Choose **New** and then **Other...** and then **Class** cause you aint gonna playing karaoke but makin some hot J2ME Apps, remember?



After you clicked **Next** define yr MIDlet class and package. I think it's the best thing for you to copy my settings so I don't have to read yr nasty emails



Click Finish. Now Eclipse beats like hell, and it creates new file called **ZareMIDlet.java**. Also Eclipse opens it in its editor. Delete all the stuff it wrote in **ZareMIDlet.java** and copy/paste this src:

```
/*
 * Created on May 1, 2005
 */
package zare;
```

```

/**
 * @author Zarko Acimovic, BMF
 */
import javax.microedition.lcdui.Display;
import javax.microedition.midlet.MIDlet;
import javax.microedition.lcdui.Displayable;

public class ZareMIDlet extends MIDlet
{

    public ZareMIDlet()
    {
    }

    public void startApp()
    {
        Displayable current = Display.getDisplay(this).getCurrent();
        PolyCanvas pc = new PolyCanvas();
        Display.getDisplay(this).setCurrent(pc);
    }

    public void pauseApp()
    {
    }

    public void destroyApp(boolean flag)
    {
    }
}

```

Now be smart and create new class **PolyCanvas** the same way you created **ZareMIDlet** class. That class should be in the same package **zare** as **ZareMIDlet** is. Don't forget to mention that in class dialog. And don't think too much. The source for **PolyCanvas** is:

```

/*
 * Created on May 1, 2005
 */
package zare;

/**
 * @author Zarko Acimovic, BMF
 */

import javax.microedition.lcdui.*;

public class PolyCanvas extends Canvas
    implements Runnable
{

    PolyCanvas()
    {
        x = 0;
        t = new Thread(this);
    }
}

```

```

        t.start();
    }

    public void paint(Graphics g)
    {

        g.drawLine(x, 56, 100, 100);
    }

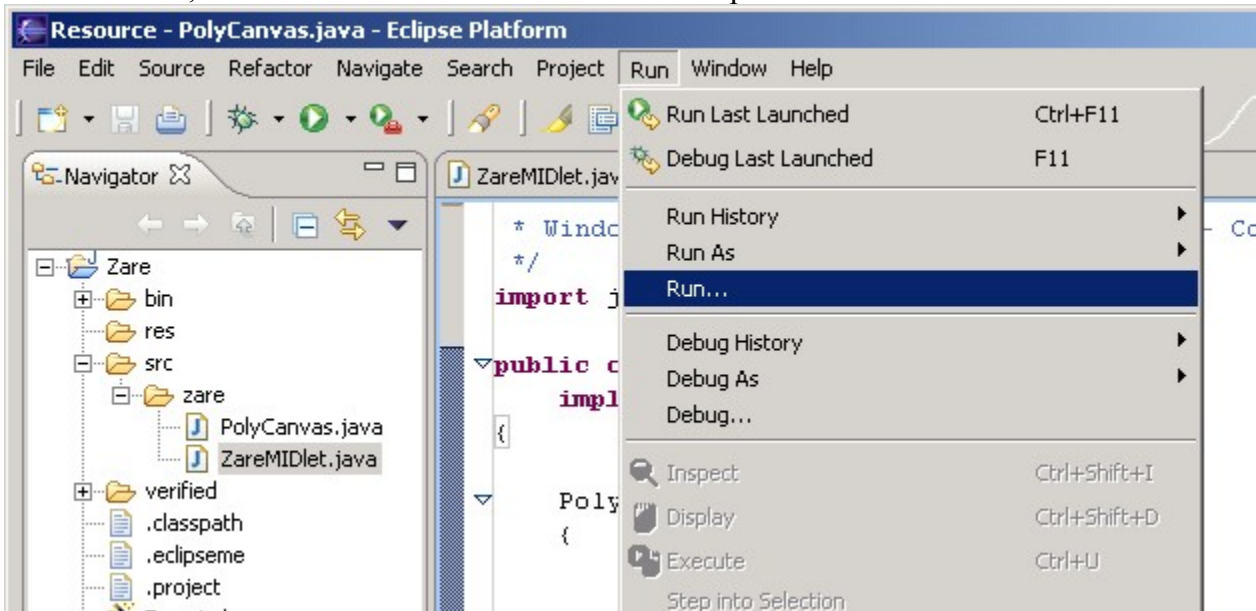
    public void keyPressed(int k)
    {
        x += 5;
        repaint();
    }

    public void run()
    {
        do
        {
            x += 5;
            repaint();
            try
            {
                Thread.sleep(5000);
            }
            catch(Exception exception) { }
        } while(true);
    }

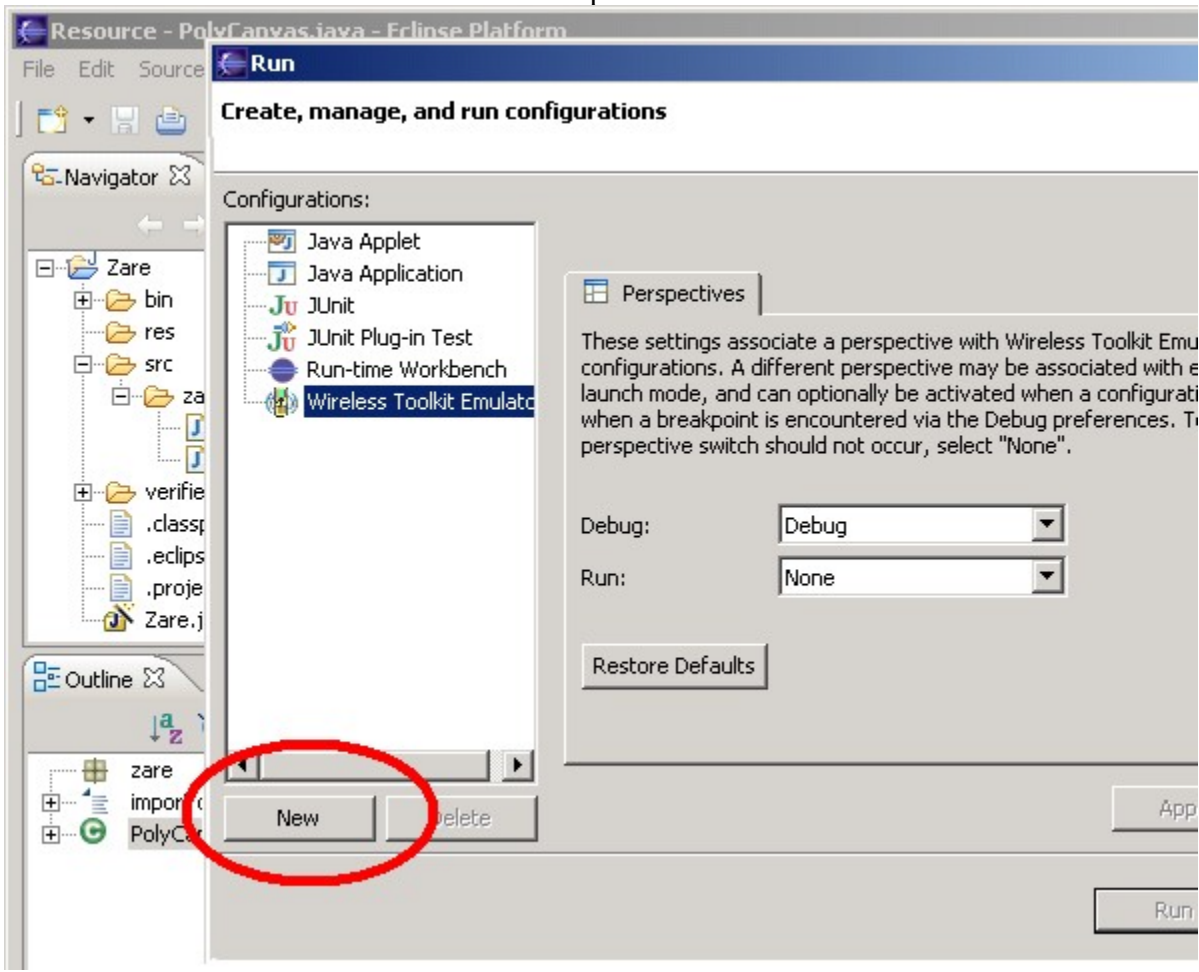
    int x;
    Thread t;
}

```

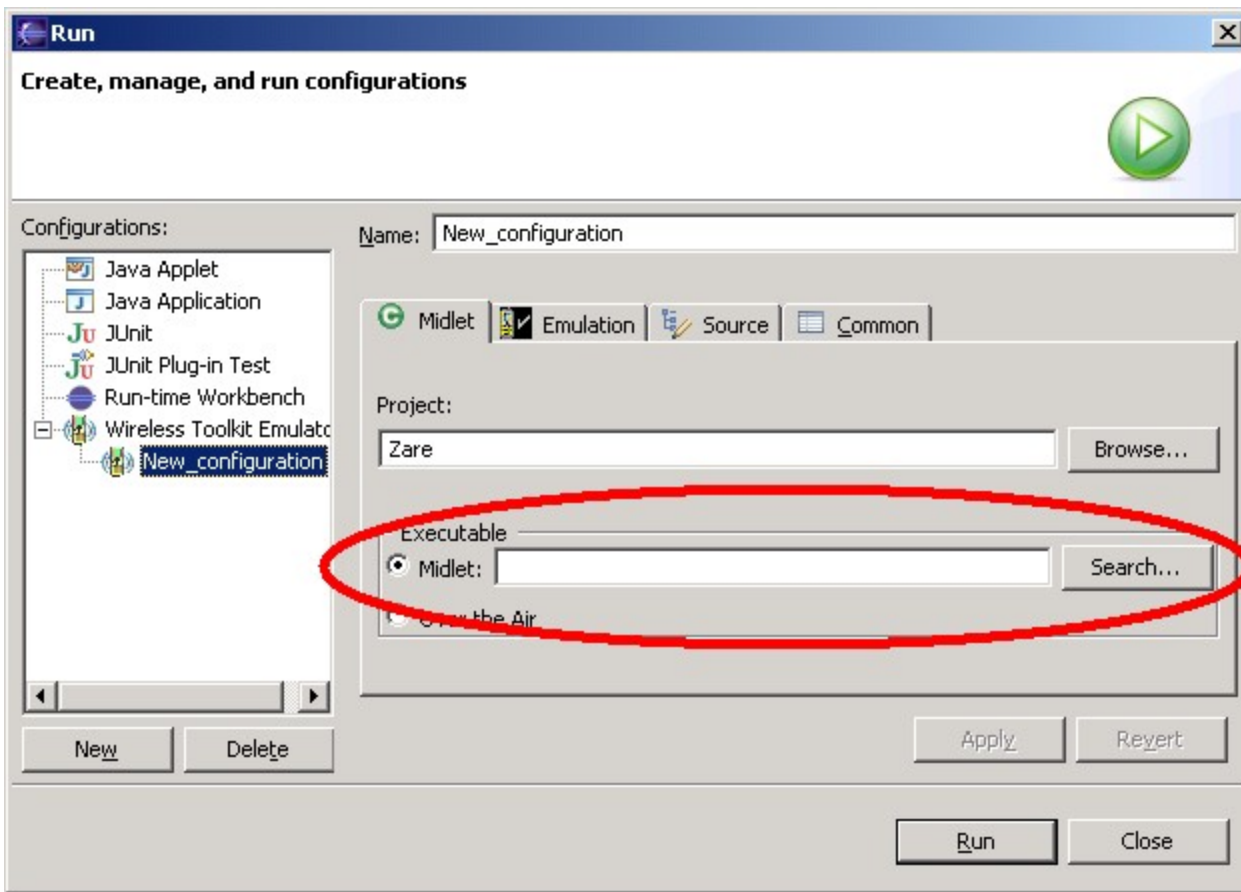
That's it. When ya save **PolyCanvas** and **ZareMIDlet** at the same time Eclipse compiles 'em. Now it's time to RUN yr app in emulator. This is where the magic comes, and problems too. But don worry, this tut covers it all. Click **Run**, and choose **Run...** like I showed in next pic



Next select **Wireless Toolkit Emulator** and press **New**



This dialog should show up:



You definitely should check **Midlet:** and then click button **Search** on the right to find out our **ZareMIDlet**. Next dialog will prompt, so you just select **ZareMIDlet**



Click **OK** and then previous dialog will show up with defined **Midlet** space. Click **Run** and application starts. It's a silly one: line is drawn every 5 secs on screen, but if ya eager to wait, you can click by mouse any button and line will be drawn immediately.



Emulator rolls over....



Feel free to send yr comments and requests for clarification at [zareac@gmail.com](mailto:zareac@gmail.com)