

J2ME applications development using Eclipse IDE

(For the Internet and Mobile Services course)

Faculty of Computer Science
Free University of Bozen-Bolzano
Oct. 2009

Overview

- Eclipse IDE installed together with appropriate plug-ins can allow you to develop J2ME applications
- You also need one plug-in
 - Eclipse MTJ (Mobile Tools for Java) (v.1.0.1)
 - <http://www.eclipse.org/dsdp/mtj/>

Eclipse IDE - Setup

- Prerequisites:
 - Sun Java 1.5.x or greater
 - Eclipse 3.3 or greater is required.
 - A supported wireless toolkit
 - Download the **Eclipse for Java** installation package
 - <http://www.eclipse.org/downloads/>
 - Unzip the downloaded file to a folder
 - Create a shortcut on the desktop (i.e., to <Unzipped folder>\eclipse.exe) to simplify the launching of Eclipse IDE
 - Launch Eclipse IDE
 - At the first time of the Eclipse IDE is launched, the system asks you to specify the workspace (i.e., your working folder)
-

EclipseME – Install (1)

- Prerequisites
 - Eclipse IDE installed
 - A supported wireless toolkit
 - **Sun Wireless Toolkits**
 - **MOTODEV SDK for JavaME**
 - **Nokia SDK S40/S60**
 - **Sony Ericsson J2ME SDK**
 - **Sprint PCS Wireless Toolkit for Java**
 - **Siemens SMTK for Series 60**
 - **mpowerplayer**
 - **Microemulator**

EclipseME – Install (2)

- Two options for installation of EclipseMTJ
 - Use Eclipse's built-in updates feature to directly install/update via Eclipse MTJ's update site
 - <http://download.eclipse.org/dsdp/mtj/updates/1.0.1/stable/>
 - From the Eclipse Help menu, select Software Updates and then Find and install...
 - Download an archive (.zip) file manually, and then install from it
 - <http://www.eclipse.org/dsdp/mtj/>
 - Restart Eclipse IDE
 - See more details on the installation instructions at <http://www.eclipse.org/dsdp/mtj/development/tutorial/installation.php>
-

EclipseME – Configuring (1)

- Import the Device Definitions, you must configure at least one Device Definition
 - Select the *Preferences* menu item from Eclipse's *Window* menu
 - Expand the J2ME item in the pane to the left and click on *Device Management*
 - Select *Import...* (Try with the Java ME SDK 3.0)
 - ... (*continue with the wizard*)

EclipseME – Configuring (2)

- Change Eclipse's Debug Settings
 - Select the *Preferences* menu item from Eclipse's *Window* menu
 - Expand the *Java* item in the left pane and click on the *Debug* entry
 - Ensure that both *Suspend execution on uncaught exceptions* and *Suspend execution on compilation errors* near the top of the dialog are NOT checked
 - Increase the *Debugger timeout* near the bottom of the dialog to at least 15000 ms

EclipseME – Creating new J2ME project

- From the *File* menu, select *New / Other...*
- Expand the *J2ME* item, select *J2ME Midlet Project*
- Enter the information
 - The project's name and location
 - Choose the wireless toolkit to be used (e.g., Sun WTK)
 - Choose the default device to run/debug the MIDlet
 - If your project uses external libraries (JAR files), go to the *Libraries* tab to add them to the project

EclipseME – Creating new MIDlet

- From the *File* menu, select *New / Other...*
- Expand the *J2ME* item, select *J2ME Midlet*
- Enter the information
 - Source folder
 - Full name of the package
 - Name of the MIDlet
- *...(continue with the wizard)*

EclipseME – Building/Packaging the project

- Building the project
 - In the *Package Explorer* panel, select the project
 - Right click, and select the *Build Project* menu
- Packaging the project
 - In the *Package Explorer* panel, select the project
 - Right click, and select "Mobile Tools for Java" and then "*Create Package*"

EclipseME – Creating Run/Debug config.

- Creating a Run/Debug configuration for the project
 - In the *Package Explorer* panel, select the project
 - Right click, and select *Run As / Run Configurations ...*
 - Select the Emulated JavaME MIDlet item, select the “New launch configuration” icon
 - Enter the information in the *Midlet* tab
 - Name of the Run/Debug configuration
 - Name of the project
 - The class of the MIDlet to be executed (or the location of the JAD file of the project)
 - Go to the Common tab, check both *Debug* and *Run* options (under the “Display in favorites menu” field)
-

EclipseME – Running/Debugging

- Running a J2ME project
 - In the *Package Explorer* panel, select the project
 - Select the drop-down button near the “*Run As...*” icon
 - Another way: Right click on the project name, select *Run As / Run Configurations ...*, and then select the appropriate Run configuration
- Debugging a J2ME project
 - In the *Package Explorer* panel, select the project
 - Select the drop-down button near the “*Debug As...*” icon (with the BUG!)
 - Another way: Right click on the project name, select *Debug As / Open Debug Dialog...*, and then select the appropriate Debug configuration

Structure of a J2ME project in Eclipse

- **.settings**
 - Settings for compiling/building the project
 - **bin**
 - The compiled classes of the project
 - **deployed**
 - The deployment (JAD and JAR files) of the project
 - **res**
 - The resources (e.g., icons, images, etc.) used in the project
 - **src**
 - The source codes of the project
-

Convert to Sun WTK project structure

- Copy 5 sub-folders of the project from Eclipse to Sun WTK

Eclipse		Sun WTK
src	(copied to)	src
deployed	(copied to)	bin
res	(copied to)	res
For EclipseME: .eclipseme.tmp\verified\libs For Eclipse MTJ: verified\libs	(copied to)	lib

- Create a text file called **project.properties**
 - In the root folder of the project (in Sun WTK)
 - Enter the text “**platform: MSA**” in the file, and save it

Sun WTK 2.5.2 - User's working folder

- By default, Sun WTK 2.5.2 creates a working folder (named ***j2mewtk***) for each user.
 - For Windows users: `C:\Documents and Settings\<<Windows_user_account_name>\j2mewtk\2.5.2`
 - For Linux users: `~/j2mewtk/2.5.2` (where `~` is your home directory)
 - To change the default working folder:
 - Open the ***ktools.properties*** file
 - Add the following line to the end of the file
kvem.apps.dir: <full_path_of_your_working_folder>
 - For example: to set the working folder to `D:\midlets`, use:
kvem.apps.dir: D:\midlets
-