How to develop eRCP Apps on Eclipse 3.1

Target Audience:

High-end mobile device application developers

About this Guide:

This guide describes the development process for the eRCP M3 build and will change for future milestone builds as the eRCP plug-in app model is more fully supported

Development Environment:

Windows 2000, XP
Java 1.4.x or higher

Target Environment:

Mobile Device capable of running J2ME – CDC/Foundation Profile
3MB of available storage
6MB of available RAM

Information about eRCP:

www.eclipse.org/ercp

Installation on Windows development machine:

Download Eclipse 3.1 SDK – www.eclipse.org/downloads
Download WSDD 5.7.1 – www.ibm.com/embedded
Download eRCP M3 package for Windows – www.eclipse.org/ercp

Install Eclipse by unzipping the download file
Install WSDD
Unzip eRCP milestone package

Run Eclipse on development machine:

Run %EclipseHome%\eclipse.exe
Import eRCP Hello sample into workspace
- Click File, then Import
- Select Checkout projects from CVS
- Select Create a new repository location and fill in following information:
  - Server: dev.eclipse.org
  - Path: /home/technology
  - User: anonymous
- Choose Use specified module name: org.eclipse.ercp/org.eclipse.ercp.example.ercpHello

In order to build and run against the correct libraries, point target platform at eRCP directory
- Click Window, then Preferences
- Expand Plug-in Development and click Target Platform
- Enter location of eRCP (i.e. eRCP-v20050727\x86\eRCP-M3) and click OK

Switch to plug-in perspective (via tab on upper right of IDE)

Setup workspace for development on Foundation Profile:

Click Window->Preferences, then select Java->Installed JREs
- Add a new JRE
  - Choose "Standard VM"
  - Provide an appropriate name such as "Foundation"
  - For JRE home, point to a standard J2SE JVM
  - Uncheck "Use default system libraries"

Delete all libraries listed
Add new libraries that point to your desired configuration. I.e.:
%WSDDhome%\wsdd5.0\ive-2.2\lib\charconv.zip
%WSDDhome%\wsdd5.0\ive-2.2\lib\jclFoundation10\classes.zip
%WSDDhome%\wsdd5.0\ive-2.2\lib\jclFoundation10\locale.zip
%WSDDhome%\wsdd5.0\ive-2.2\lib\jclFoundation10\map.zip

Note: Setting the default JRE affects the entire workspace except for those projects that override the default with their own preference. You can set the workspace default to Foundation and still keep your "Run..." configurations as J2SE.

Set sample project to use Foundation JRE

Right click on ercpHello project
Select Properties and Java Build Path
Select Libraries tab
Remove JRE System Library
Click Add Library and select JRE System Library
Check Alternate JRE, and select Foundation JRE, then click Finish

Running sample on desktop machine:

Right click on ercpHello project
Click on Run As and select Eclipse Application
Sample should appear in a new window

Sample Description:

The sample demonstrates an application provided as an OSGi bundle. However, the M3 build only supports one GUI application per JVM. Therefore, the current app model most closely resembles the RCP application model but without the access to workbench features. The next (M4) build will support the eRCP plug-in application model described below.

Application Model Details:

There are many similarities, but also some differences among Eclipse Application/Plugin Models. This section describes these models for reference:

Eclipse IDE Plug-in Model – used primarily by the Eclipse Integrated Development Environment (IDE) tooling. Various tooling plug-ins may contribute perspectives and/or views to the IDE. All plug-ins are OSGI bundles that extend workbench extension points and are controlled via the IDE workbench. The plug-ins all run with in the workbench's JVM.

RCP Application Model – provides the power of the Eclipse workbench functionality to non-IDE applications. The application assumes the role of workbench allowing it to consist of multiple independent parts. Each RCP application runs within its own JVM.

eRCP Plug-in Application Model – is most similar to the Eclipse IDE plug-in model. A generic or device specific workbench uses a *Perspective* to allocate screen space and control the display of eRCP Applications which are composed of various *Views*. A plug-in eRCP app does not have a main method, but instead implements an eWorkbench extension which allow it to be discovered and run on the workbench's thread. The workbench shares its execution thread amongst all eRCP applications, thus allowing multiple eRCP apps to run within a single JVM

HelloApplication.java Code:

This class is required to fit the sample into the RCP application model. It extends Plugin which lets it be discovered and loaded by the OSGi runtime. It implements IPlatformRunnable which allows it to be executed as an Eclipse Application.

ErcpHello.java Code:

This class implements the application GUI by making eSWT calls and then going into the standard SWT event loop.

Manifest Description:

The manifest file records information about the plug-in needed for it to execute. For convenience, I'll refer to sections of the manifest as shown by the Eclipse Manifest Editor.

Under Overview, the full plug-in class name must be specified.

Under Dependencies, Required Plug-ins, org.eclipse.core.runtime is required. Under Imported Packages are listed any eSWT or eJFace packages that are used by the application, such as:

org.eclipse.swt
org.eclipse.ercp.swt.mobile
org.eclipse.jface

These Java packages are specified by *package name* versus *plug-in ID* because the deployment of these packages may be done through a variety of plug-in configurations. For instance, on the Pocket PC platform, Core eSWT, Expanded eSWT and SWT Mobile Extensions may all be packaged together in a single plug-in.

Under Extensions, org.eclipse.core..runtime.applications is required. The ID is a name you may choose to uniquely identify this application.

Under Build, Runtime Information, the dot path specifies that classes can be found by searching from the root of the plug-in. The items checked under Binary Build determine what else will be included when the sample plug-in is built.

The remaining tabs show the actual XML built from information provided in the previous tabs and usually does not need to be altered.

Deploying eRCP to a WM2003 target device:

Download eRCP M3 package for WM2003 – www.eclipse.org/ercp
Unzip download file
Follow the platform read.me for installation instructions
Run the pre-built Hello sample by launching file explorer and clicking on j9foun-hello

Exporting the sample plug-in:

Right click on ercpHello project
Click on Export
Select Deployable plug-ins and fragments and click Next
Select the ercpHello project
Under Export Destination, select Directory and choose eRCP-v20050727\wm2003\eRCP-M3
Click Next and then Finish
The plug-in is now deployable

Running sample:

Copy the ercpHello jar from the eRCP-M3\plugins directory to the same directory on your target device
Run the ercpHello sample you built by launching file explorer and clicking on j9foun-hello

Using the sample as a template:

Right click on the ercpHello project and select Team, then Disconnect
Choose Remove CVS information
Rename your project as desired