Development with the Embedded Rich Client Platform (eRCP)

Chris Aniszczyk, IBM Lotus, Austin
Gorkem Ercan, Nokia, Finland
Mark Rogalski, IBM Lotus, Austin
Agenda

- Exercise Tracker Exercise 😊
- Overview
- eSWT
  - Introduction
  - Mobile Extensions
  - Developing Mobile applications
- Working with Devices
- Troubleshooting and Debugging
- Conclusion
- Q&A
Agenda

- **Exercise Tracker Exercise 😊**
- Overview
- eSWT
  - Introduction
  - Mobile Extensions
  - Developing Mobile applications
- Working with Devices
- Troubleshooting and Debugging
- Conclusion
- Q&A
Exercise Tracker

- **Requirements**
  - Eclipse 3.2.2
    - [www.eclipse.org/downloads](http://www.eclipse.org/downloads)
  - eRCP
    - [www.eclipse.org/ercp](http://www.eclipse.org/ercp)
  - Sample Plug-ins
    - [http://eclipsezilla.eclipsecon.org/attachment.cgi?id=428](http://eclipsezilla.eclipsecon.org/attachment.cgi?id=428)
Exercise Tracker
Agenda

- Exercise Tracker Exercise 😊
- **Overview**
- eSWT
  - Introduction
  - Mobile Extensions
  - Developing Mobile applications
- Working with Devices
- Troubleshooting and Debugging
- Conclusion
- Q&A
Relating eRCP to RCP

- Action sets
  - Editors
  - Perspectives
  - Views
  - Workbench

- UI
- Runtime
- OSGi
- Bundles
  - Services

- JFace
- SWT
- Extension
  - Applications
  - Products
- Widgets
Relating eRCP to RCP

- eRCP
  - Views Workbench
    - eWorkbench
      - eCore
        - OSGi
      - eJFace
        - eSWT
          - Viewers
          - Widgets
        - Bundles Services
          - Extensions Applications Products
Architecture

- OSGi underpinnings
- eRCP applications run in a workbench similar to Eclipse IDE plugins
- Also supports stand-alone eSWT applications
- Applications provide a View extension which the eWorkbench instantiates on demand
- Applications and services run in the same JVM
  - Consumes fewer resources than separate JVMs
  - Allows sharing of services
  - Enables variety of life cycle choices
Application Model

- eSWT UI Applications
- SWT
  - Core eSWT (required)
  - Expanded eSWT (optional)
- Mobile Extensions for SWT (optional)
- Java Virtual Machine
- Native Operating System
eWorkbench

- Applications provide a View extension which the eWorkbench instantiates on demand
- Applications and services run in the same JVM
  - Consumes fewer resources than separate JVMs
  - Allows sharing of services
  - Enables variety of life cycle
- Owns eSWT UI thread
- Applications are registered using
  org.eclipse.ercp.eworkbench.applications extension point
Agenda

- Exercise Tracker Exercise 😊
- Overview
- eSWT
  - Introduction
  - Mobile Extensions
  - Developing Mobile applications
- Working with Devices
- Troubleshooting and Debugging
- Conclusion
- Q&A
Introduction to eSWT

- Subset of SWT and additional mobile specific widgets
- Provides efficient, portable access to the user interface facilities of the operating system
- Consists of 3 packages
  - SWT subset divided into two
    - core package
    - expanded package
  - mobile package - new eSWT components defined for mobile world
Class Diagram
Mobile package - MobileShell

- A device tailored Shell that can change the trimmings dynamically
  - Top-level shell
  - Full screen mode
  - Allows key press polling
- Introduces status pane styles
  - NO_STATUS_PANE
  - SMALL_STATUS_PANE
  - LARGE_STATUS_PANE
Mobile package - Basic controls

- ConstrainedText
- DateEditor
- HyperLink
- TextExtension
- CaptionedControl
- SortedList
- TaskTip
  - Suitable for providing info on long running tasks
  - Text and optional ProgressBar
Mobile package - Advanced Controls

- **ListBox**: A list control with enhanced capabilities to display icons
- **ListView**: Selectable control that can display items in a multi-column way
- **MultipageDialog**: A tabbed dialog
Mobile package – Dialogs

- **QueryDialog**: Several query types
  - **STANDARD**: alphanumeric input
  - **NUMERIC**
  - **TIME**
  - **DATE**
  - **PASSWORD**

- **TimedMessageBox**:
  - 4 Different system Icons (working, information, warning, error)
  - Icons can be replaced
Mobile package – Commands

- Maps to a specific mechanism depending upon device capabilities
- Has logical types that are typically mapped to Soft keys (GENERAL, SELECT, OK, CANCEL, DELETE, BACK, EXIT, STOP, HELP)
  - COMMANDGROUP can contain other sub-commands
  - S60 implementation maps DELETE commands to ‘C’ clear key.
- Bound to focus context
Mobile package – MobileDevice, Screen & Input

- **MobileDevice**
  - Discover device capabilities and characteristics
  - Register listeners for changes in input, screen capabilities
- **Screen**
  - Query the capabilities of the screen(s)
- **Input**
  - Determine input device capabilities
Tips: Building mobile applications using eSWT

- Do not rely on Shell trimmings some platforms do not support trimmings like SWT.CLOSE
- Do not use too deep menu hierarchies.
- Use Commands in relation with the focus context to avoid the soft keys getting crowded
- Use MobileDevice, Screen, and Input to adjust the behavior at runtime
  - Active screens, active input methods may change
- Always use layout managers
- Check the computed layout size and add use scrollbars if greater than screen size
- Tailor your application according to aspect ratios
Tips: Choosing the right input widget

<table>
<thead>
<tr>
<th></th>
<th>Multiple Lines</th>
<th>Numeric</th>
<th>Decimal</th>
<th>Phone Number</th>
<th>DateTime</th>
<th>Duration/Offset</th>
<th>E-Mail</th>
<th>URL</th>
<th>Initial Case</th>
<th>Initial Input Mode</th>
<th>Turn off Prediction</th>
<th>Latin Input Only</th>
<th>Password</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>TextExtension</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>ConstrainedText</td>
<td></td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>DateEditor</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>QueryDialog</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>
Agenda

- Exercise Tracker Exercise 😊
- Overview
- eSWT
  - Introduction
  - Mobile Extensions
  - Developing Mobile applications
- Working with Devices
- Troubleshooting and Debugging
- Conclusion
- Q&A
Installing eRCP on a Device (1/3)

- Download the latest eRCP runtime for your device from [http://www.eclipse.org/ercp](http://www.eclipse.org/ercp)
- Unzip the run-time and copy it over to the root directory of the device
  - Make sure you device has a JRE installed!
Installing eRCP on a Device (2/3)

- The eRCP directory should be copied to the root of the device
- Browse to the eRCP directory on the actual device
  - Notice the familiar eclipse directory structure
Installing eRCP on a Device (3/3)

- Run the j9foun-hello shortcut
- You should see something similar to what’s picture on the left
- You have successfully installed eRCP on your device!
Running Applications

- All eRCP applications are Eclipse-based applications
  - org.eclipse.core.runtime.applications
  - IPlatformRunnable's
- If you look at one of the sample shortcuts included, they simply launch Eclipse
  - i.e., `<jvm> -cp <startup.jar> -application <app>`
Agenda

- Exercise Tracker Exercise 😊
- Overview
- eSWT
  - Introduction
  - Mobile Extensions
  - Developing Mobile applications
- Working with Devices
- Troubleshooting and Debugging
- Conclusion
- Q&A
Troubleshooting

- Make sure the target platform is set to the eRCP target platform
- The application ID extension point isn’t unique, check the ID in the `org.eclipse.ercp.eworkbench.application` extension point
Debugging

- **Tips**
  - Develop as much as possible on the device runtime
  - Take advantage of a resizable workbench window to test different screen sizes and proportions
  - Copying plug-ins directly to the device will work if you have the osgi.checkConfiguration property set to “true”
Agenda

- Exercise Tracker Exercise 😊
- Overview
- eSWT
  - Introduction
  - Mobile Extensions
  - Developing Mobile applications
- Working with Devices
- Troubleshooting and Debugging
- **Conclusion**
- Q&A
Conclusion

- eRCP brings the RCP paradigm to mobile devices
- Website
  - http://www.eclipse.org/ercp
- Newsgroups
  - news://news.eclipse.org/eclipse.dsdp.ercp
- Always looking for contributors!
Questions & Answers