

Eclipse on Cell phones?

An Introduction to the eRCP

~ Chris Aniszczyk <zx@us.ibm.com> ~



Agenda

- Overview (~10 mins)
- eSWT (~20 mins)
 - Introduction
 - Mobile Extensions
- Working with Devices (~10 minutes)
- Troubleshooting and Debugging (~5 minutes)
- Conclusion (~5 minutes)
- Q&A & Demo(?)

Agenda

- **Overview**
- eSWT
 - Introduction
 - Mobile Extensions
- Working with Devices
- Troubleshooting and Debugging
- Conclusion
- Q&A

eRCP Mission Statement

- *“The intent of this project is to extend the **Eclipse Rich Client Platform (RCP)** to embedded devices. eRCP is largely a set of components which are subsets of RCP components. It basically **enables the same application model used on desktop machines to be used on devices.**”*

What exactly is eRCP?

- Embedded version of RCP
- Utilizes RCP application framework model
- Reduces RCP size/function to fit on devices
- Pushes changes back to core components to enable running those components on J2ME CDC/Foundation Profile
 - eRCP requires CDC 1.0
 - eSWT can run on CLDC 1.1
- Adds component to enable application binary compatibility across a range of devices with different input mechanisms and screen types/sizes

Why is eRCP important?

- Next generation of mobile Java development!
- Extensive rich UI (RCP) capabilities
 - Integration of native platform look & feel
- Higher level of device abstraction
- Delivers OSGi service oriented features to devices
 - Dynamic install/uninstall/update
 - Service sharing
 - Heck, OSGi is just *cool*
- Puts Eclipse programming model on devices!

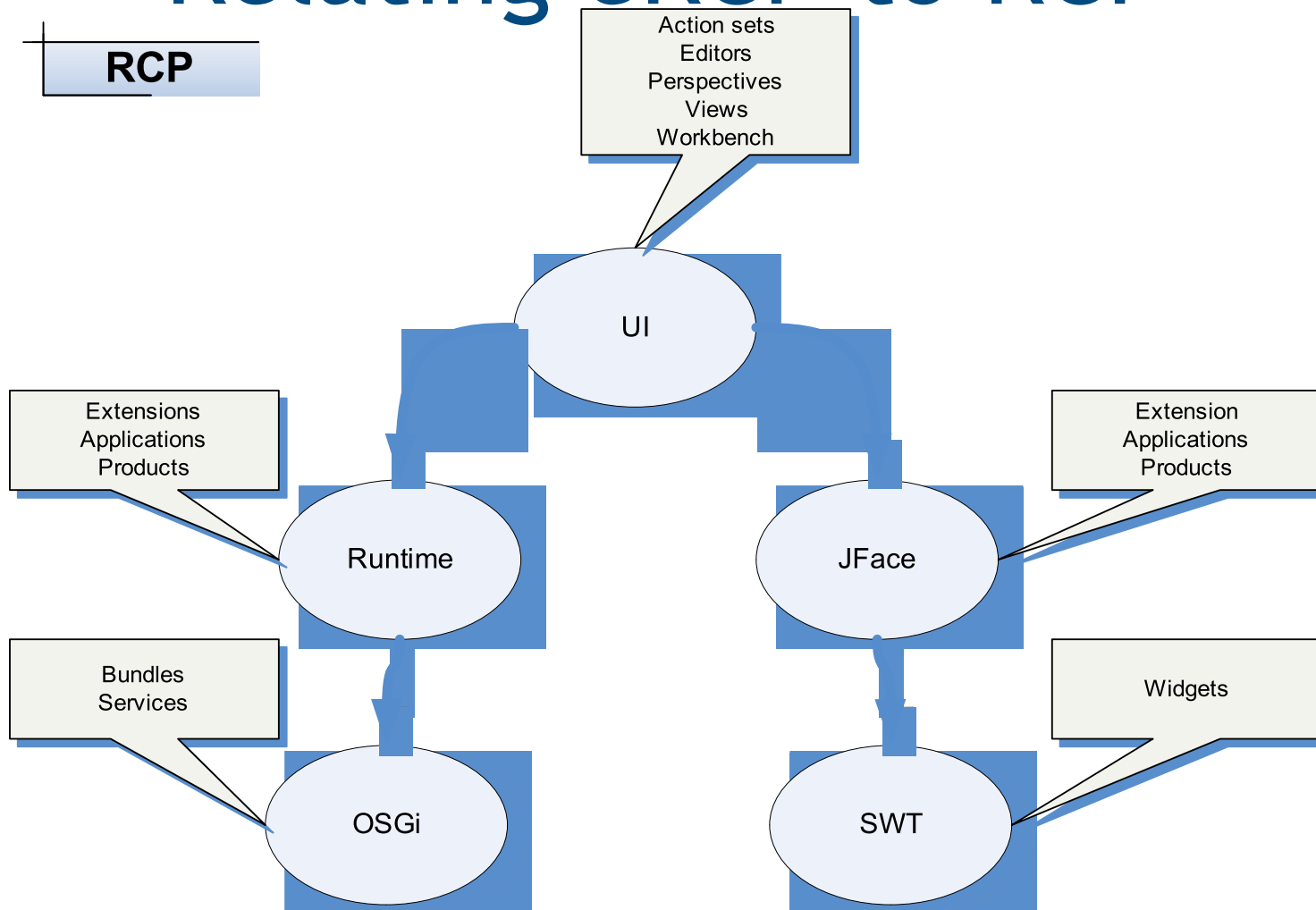
Platform Implementations

- Current
 - Windows Mobile 2003/2005
 - Windows Desktop
 - Windows CE (eSWT only)
 - Nokia Series 60*/80

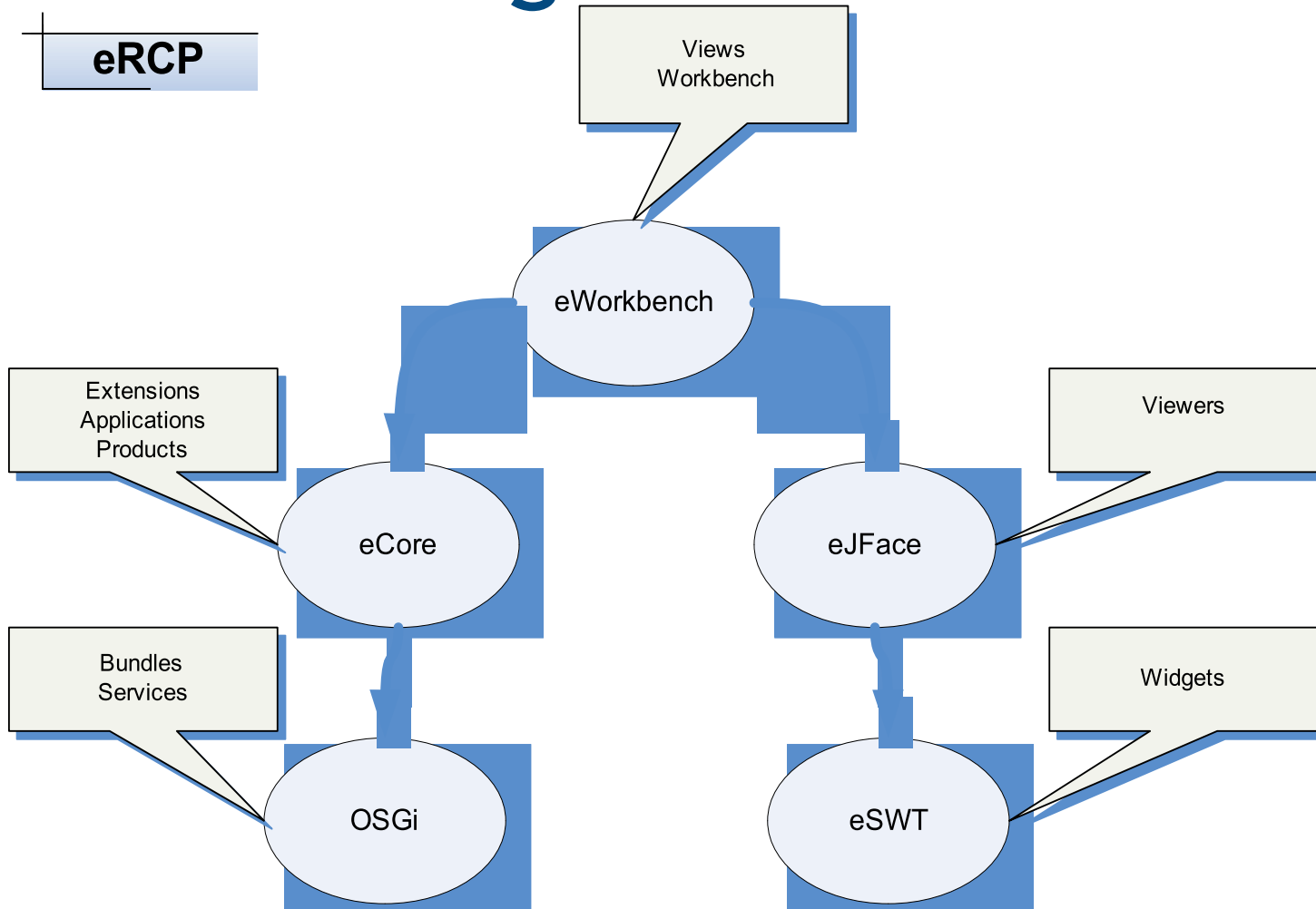
Architecture Overview

- OSGi underpinnings
- eRCP applications run in a workbench similar to Eclipse IDE plugins
- eRCP supports stand-alone eSWT applications
- Applications and services run in the same JVM
 - Consumes fewer resources than separate JVMs
 - Allows sharing of services
 - Enables variety of life cycle choices

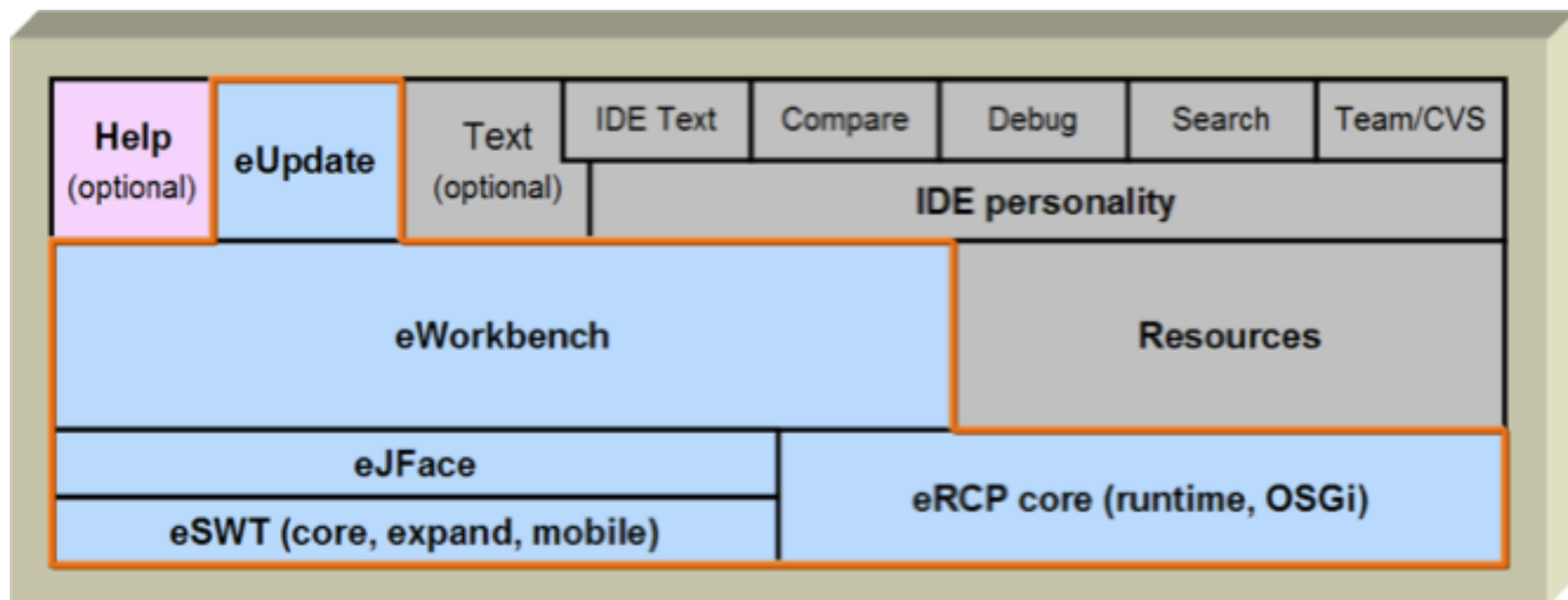
Relating eRCP to RCP



Relating eRCP to RCP



Quick Architecture Diagram



RCP Subsets

- **eSWT**
 - Core - Base function and widgets
 - Expanded - Adds Browser, Table, Tree, more layouts
- **eJFace**
 - Subset provides Table/Tree Viewers, Resource handling
- **eUpdate**
 - Allows end-user to browse to update site and download features
 - Provides subset of API for applications to use update services

New Components

- **Mobile Extensions**
 - For eSWT - provides device abstraction
 - For Desktop SWT - provides upward compatibility with desktops
- **eWorkbench**
 - Provides GUI for launching/switching among eRCP apps
 - Customizable by device provider to support devices features
- **microXML**
 - Provides full SAX and DOM support in very small footprint
- **Test Harness**
 - Provides test delivery to device, execution, and reporting

Application Model

- Applications provide a View extension which the eWorkbench instantiates on demand
- Applications and services run in the same JVM
- Owns eSWT UI thread
- Applications are registered using *org.eclipse.ercp.eworkbench.applications* extension point

Agenda

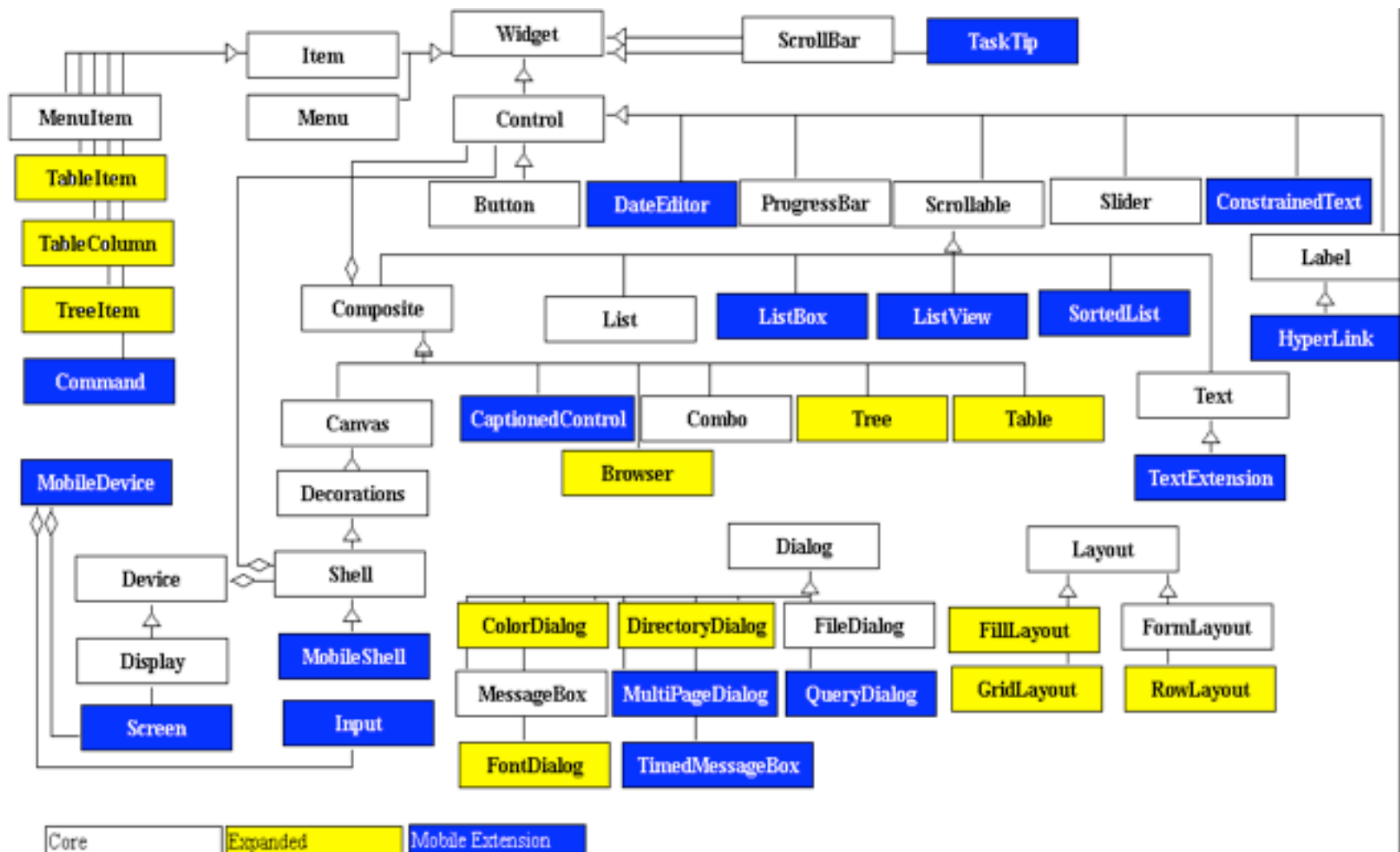
- Overview
- **eSWT**
 - Introduction
 - Mobile Extensions
- Working with Devices
- Troubleshooting and Debugging
- Conclusion
- Q&A

Introduction to eSWT

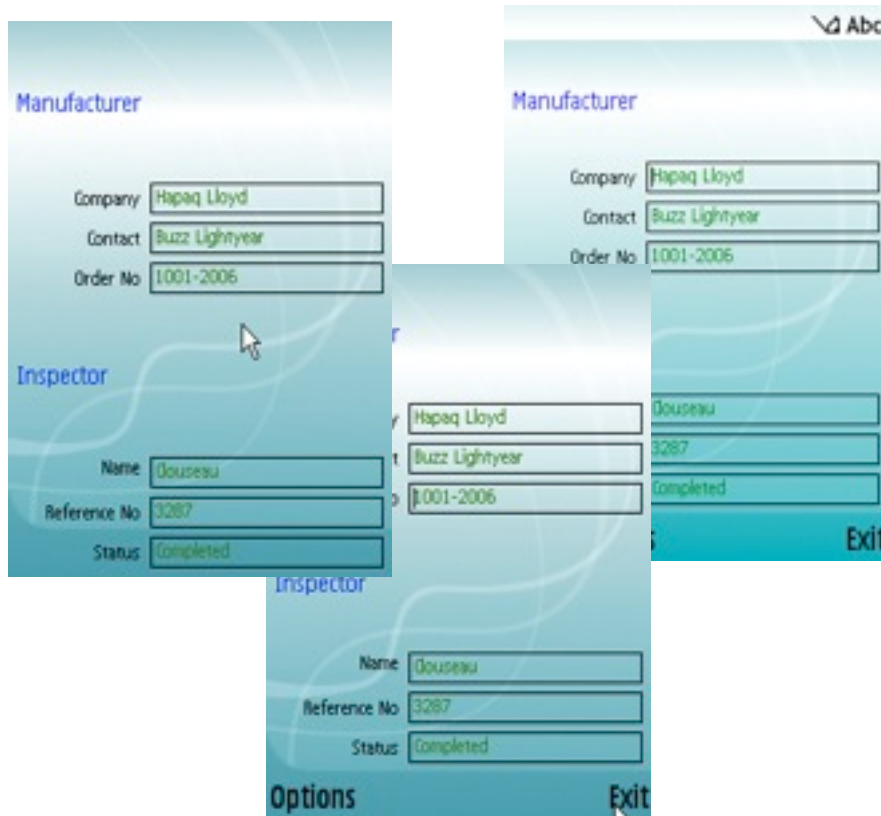
- Subset of SWT and additional mobile specific widgets
- Provides efficient, portable access to the user interface facilities of the operating system
- Consists of 3 packages (core, expanded, mobile)
 - SWT strict subset divided into two
 - *core* package
 - *expanded* package
 - *mobile* package
 -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

eSWT Components

- **eSWT Core**
 - Contains fundamental user interface elements, including low-level graphics, events, and basic widget infrastructure. This portion of eSWT is device-agnostic and applicable to a wider range of embedded devices. This component is a **strict subset** of SWT.
- **eSWT Expanded**
 - An *optional package* that provides more sophisticated user interface elements that are found commonly on high-end mobile devices and PDAs. This component is a subset of SWT.
- **eSWT Mobile Extension**
 - An *optional component* that provides user interface elements that are found commonly on mobile devices. Although many of the mobile device requirements are fulfilled already by eSWT, this package focuses on critical features that are not defined currently by eSWT.



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 - CaptionedControl

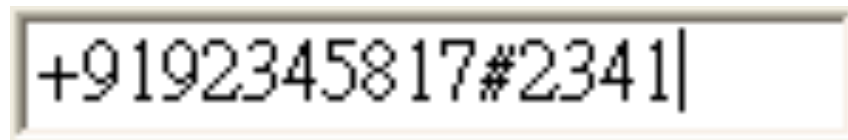
A CaptionedControl is used to display a label (caption) in front of a control.

Name: John Marshall Software Engineer

Determining which control has focus can be difficult on mobile devices where lighting conditions are often less than optimal. With *CaptionedControl*, a label shows focus highlighting whenever the control has focus. With an entire label highlighted, it is easy to locate the focus.

Mobile package - ConstrainedText

ConstrainedText is a single-line Text control that constrains the user input by styles. This control limits the characters that users can enter by styles to make data entry more efficient. This capability is usually implemented by the underlying platform.



+9192345817#2341|

Style	Description
DECIMAL	Allows the input of numeric values with optional decimal fractions (for example, -123, 0.123, or .5 are all valid input)
NUMERIC	Allows the input of numeric values
PHONENUMBER	Allows the input of numeric values with optional phone specific characters such as a plus sign (+), an asterisk (*), and a number sign (#)

Mobile package - Hyperlink

Instances of this class represent a selectable user interface object that launches other applications when activated by the user. This class represents several types of hyperlinks associated with certain functionalities. The user can activate the associated program determined by the style.

john@foo.com



Style	Description
EMAIL	Opens the implementation-dependent e-mail client when activated
PHONE	Shows the implementation-dependent dialer interface when activated
URL	Launches a implementation-dependent Web browser when activated



Mobile package - DateEditor



Style	Description
COMPACT	A hint that the widget should be displayed in a format which is smaller or less full featured
DATE	A date entry type for year, month and day
DATE_TIME	An entry type for date and time
DURATION	An entry type for a period of time in hours, minutes and seconds
FULL	A hint that the widget should be displayed in a format which is full featured and emphasizes ease of use rather than compactness
OFFSET	An entry type for a period of time in hours, minutes, and seconds which can be subtracted or added to another time value
TIME	A time entry type for hours, minutes and seconds

Mobile package - TextExtension

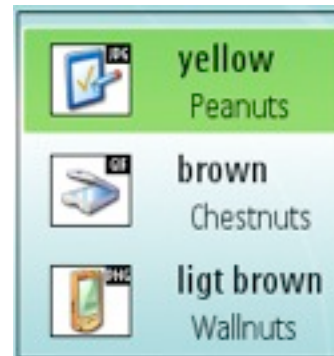
test@ibm.com
http://

Content Styles	Description
EMAILADDRESS	An e-mail address
URL	A Web address

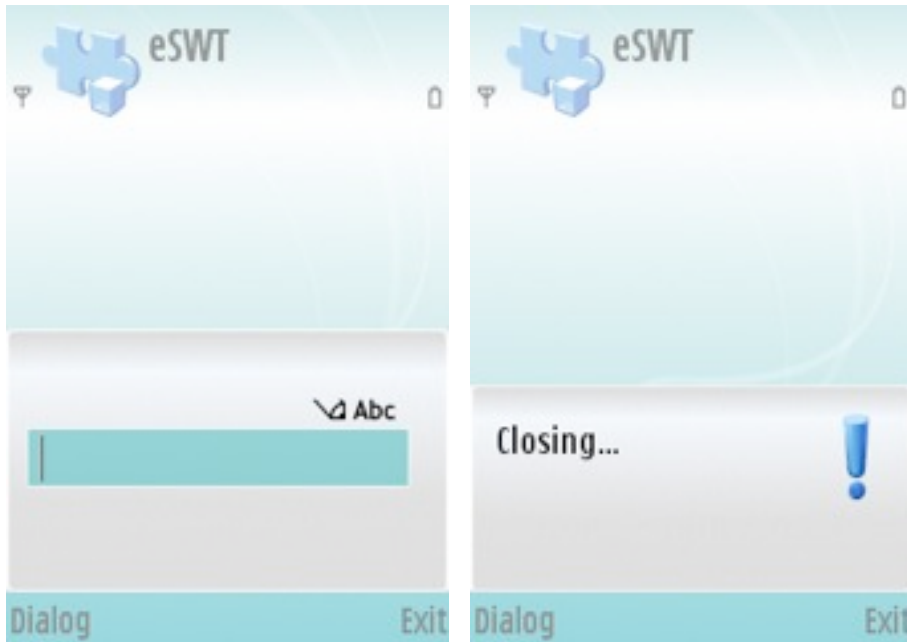
Input Modifier Styles	Description
NON_PREDICTIVE	A hint for turning off possible predictive text input. By default any predictive input facilities should be turned on if available
LATIN_INPUT_ONLY	Forces that locale specific input modes should not be available. This is used in some situations when only Latin characters are allowed, for example, password text field

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 (soft keys) 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 eSWT Apps

- 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

Tips: Choosing the right widget

	Multiple Lines	Numeric	Decimal	Phone Number	Date/Time	Duration/Offset	E-Mail	URL	Initial Case	Initial Input Mode	Turn off Prediction	Latin Input Only	Password
Text	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓
TextExtension	✓	✓	✓	✓	✓	✓	⊙	⊙	⊙	⊙	⊙	⊙	✓
ConstrainedText	✗	⊙	⊙	⊙	✗	✗	✗	✗	✗	✗	✗	✗	✗
DateEditor	✗	✗	✗	✗	⊙	⊙	✗	✗	✗	✗	✗	✗	✗
QueryDialog	✗	⊙	✓	✓	⊙	✓	✓	✓	✗	✗	✗	✗	⊙

Agenda

- Overview
- eSWT
 - Introduction
 - Mobile Extensions
- **Working with Devices**
- Troubleshooting and Debugging
- Conclusion
- Q&A

Installing eRCP on a Device (1 / 3)

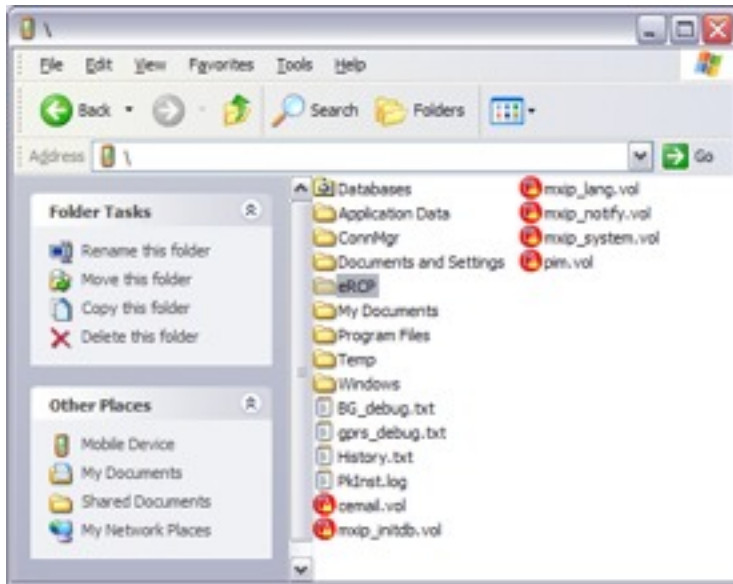
embedded Rich Client Platform (eRCP)

The intent of this project is to extend the Eclipse Rich Client Platform (RCP) to embedded devices. eRCP is largely a set of components which are subsets of RCP components. It basically enables the same application model used on desktop machines to be used on devices.

Downloads	Community
<ul style="list-style-type: none">→ eRCP 1.0.1 Now Available! → eSWT bundles for WSDD (from M7)→ eSWT 1.0 Spec - updated Aug/03/06 (zip)→ eJFace JavaDoc - updated Dec/07/05 (zip)	<ul style="list-style-type: none">→ Documents→ eRCP newsgroup (Password Required) - Web Interface→ dsdp-ercp-dev mail list (archive)→ Bugzilla→ CVS

- Download the latest eRCP runtime for your device from <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

- Overview
- eSWT
 - Introduction
 - Mobile Extensions
- Working with Devices
- **Troubleshooting and Debugging**
- Conclusion
- Q&A

Troubleshooting & 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

- Overview
- eSWT
 - Introduction
 - Mobile Extensions
- 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!
 - QTe / GTK mobile
 - Blackberry!?

Agenda

- Overview
- eSWT
 - Introduction
 - Mobile Extensions
- Working with Devices
- Troubleshooting and Debugging
- Conclusion
- Q&A

Questions & Answers

Backup Slides for Exercise

Exercise 1: Setup the Environment

- Download
 - Grab latest eRCP (Windows Desktop) latest
 - <http://www.eclipse.org/ercp>
 - Eclipse 3.3
 - <http://www.eclipse.org>
- Run the hello application



Exercise 2: eSWT

```
public static void main(String[] args) {  
    Display display = new Display();  
    Shell shell = new Shell(display);  
    shell.setText("Hello World");  
    shell.setLayout(new FillLayout());  
    Label label = new Label(shell, SWT.CENTER);  
    label.setText("Howdy");  
    shell.open();  
    while (!shell.isDisposed())  
    {  
        if (!display.readAndDispatch())  
            display.sleep();  
    }  
    display.dispose();  
}
```

Exercise 3: eWorkbench (1/3)

```
public class AppPlugin extends AbstractUIPlugin {  
    private static AppPlugin plugin;  
  
    public AppPlugin() {  
        super();  
        plugin = this;  
    }  
  
    public void stop(BundleContext context) throws Exception {  
        super.stop(context);  
        plugin = null;  
    }  
  
    public static AppPlugin getDefault() {  
        return plugin;  
    }  
}
```

Exercise 3: eWorkbench (2/3)

```
public class NormalView extends ViewPart {  
    Label label;  
  
    public void createPartControl(Composite parent)  
    {  
  
        label = new Label(parent, SWT.NONE);  
        label.setText("Hello World");  
        label.setSize(parent.getSize());  
    }  
  
    public void setFocus() {}  
}
```

Exercise 3: eWorkbench (3/3)

```
<extension
```

```
point="org.eclipse.ercp  
.eworkbench.applicatio  
ns">
```

```
  <application  
    id="sample.app"  
    name="Sample  
Application">
```

```
    <views
```

```
<extension  
point="org.eclipse.ui.views">
```

```
  <view
```

```
category="org.eclipse.ercp.ework  
bench.viewCategory"
```

```
class="sample.app.views.Normal  
View"
```

```
  icon="icons/sample.gif"
```

```
id="sample.app.views.normal"  
  name="Sample Normal  
View">
```

```
  </view>
```

```
</extension>
```


Exercise 4: Deployment

- **Packaging**
 - Feature Creation
 - Update-site Creation
- **Deployment**
 - Launch eWorkbench
 - Run the Application Manager
 - Install application from previous exercise
 - Restart eWorkbench
 - Run the newly installed application