



RAP (The Rich Ajax Platform)

Frank Appel
RAP Tech lead
fappel@innoopract.com

Jochen Krause
RAP Project lead
jkrause@innoopract.com

Why RAP?



RAP enables
**component oriented development and assembly
of web applications**

using
SWT, JFace and Workbench technology

What we like about Eclipse



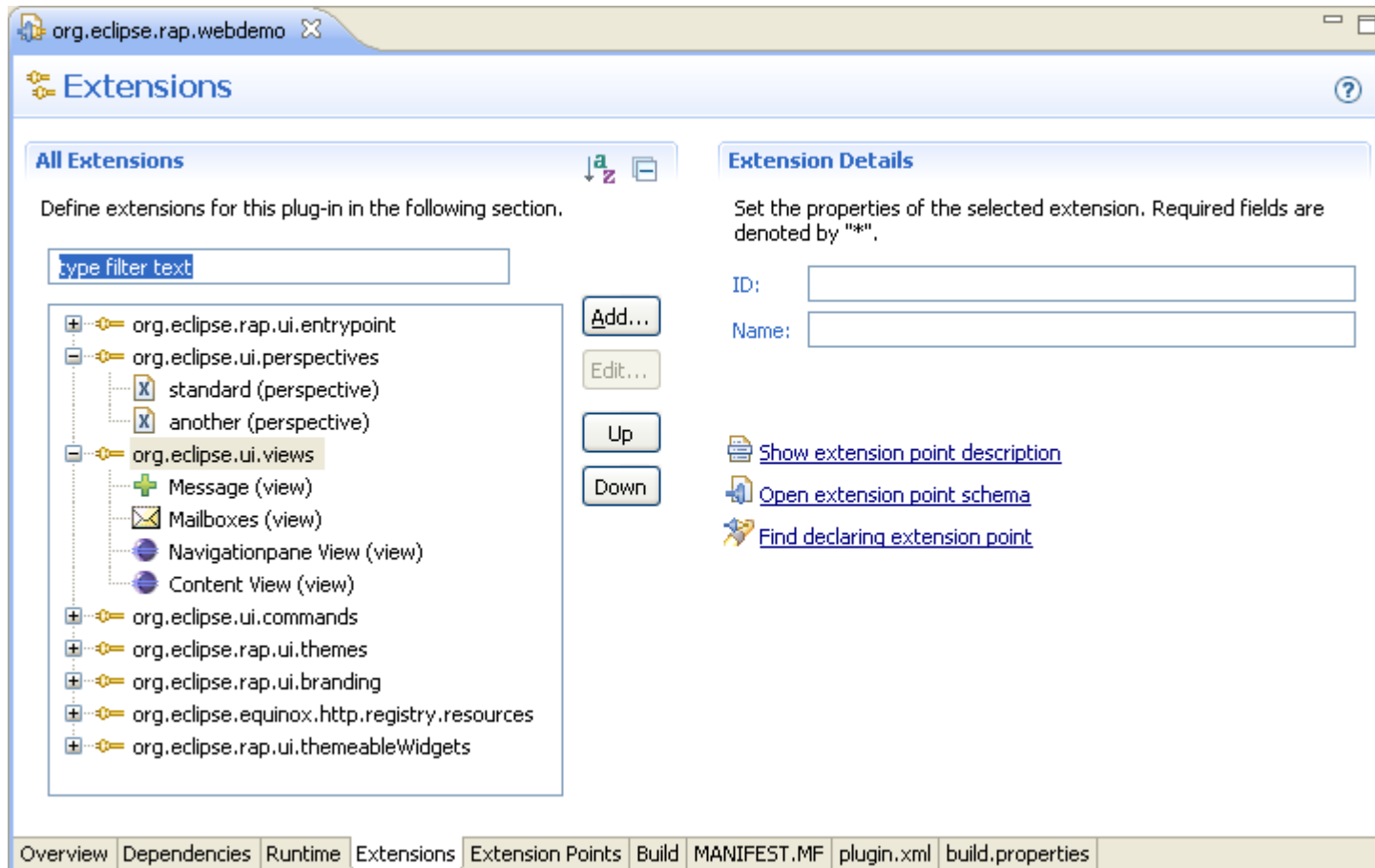
plug-ins, plug-ins, plug-ins – bundles too ...

- dependency management
- extension points
- life cycle management

contribution to a common ui (workbench)

- this is called “mashup” in web 2.0

what we like about Eclipse (cont'd)



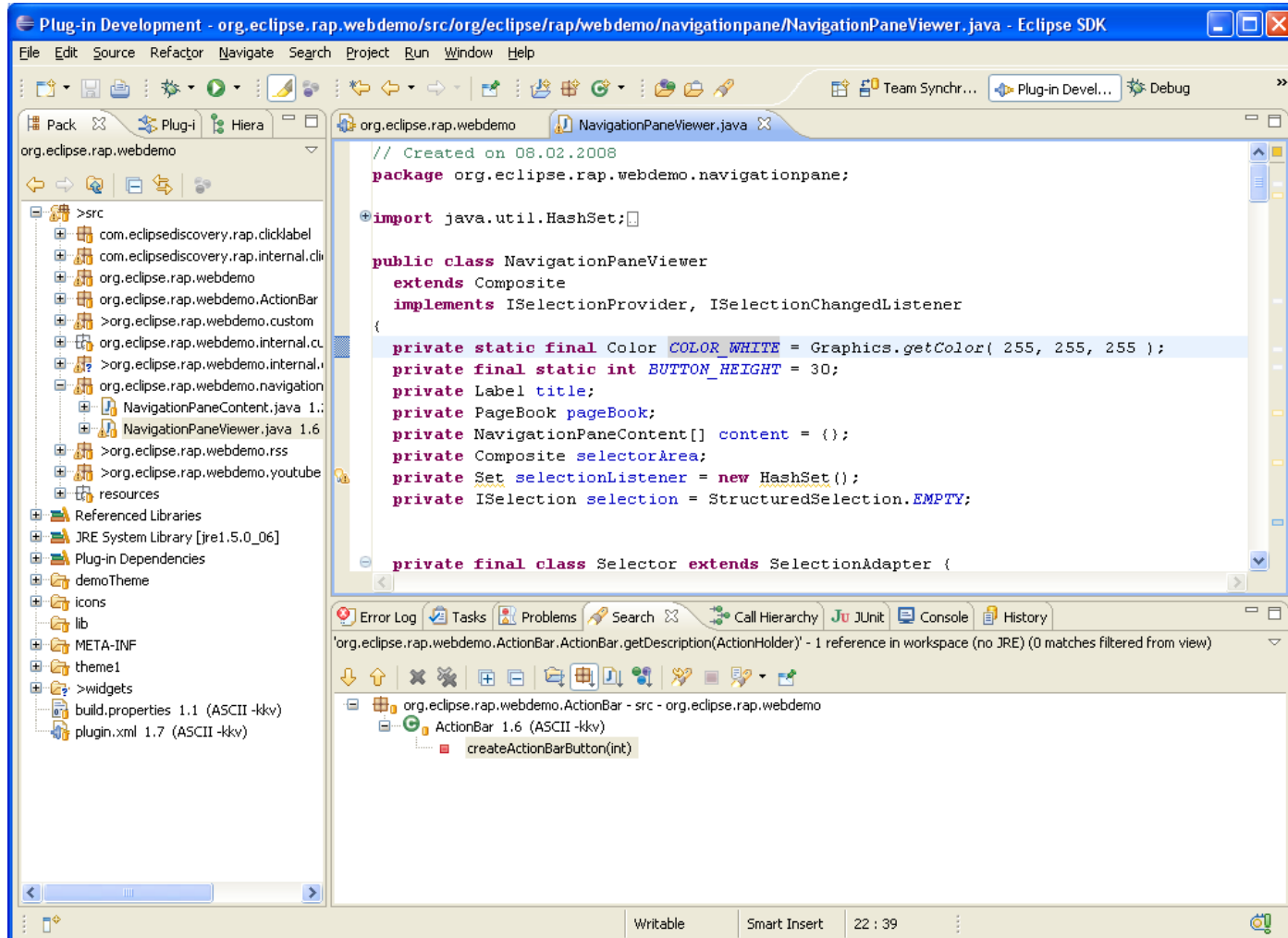
The screenshot shows the Eclipse IDE's "Extensions" dialog box. The window title is "org.eclipse.rap.webdemo". The dialog is divided into two main sections: "All Extensions" and "Extension Details".

All Extensions: This section contains a search box with the text "type filter text". Below it is a tree view of extension categories. The "org.eclipse.ui.views" category is selected and expanded, showing sub-items: "Message (view)", "Mailboxes (view)", "Navigationpane View (view)", and "Content View (view)". To the right of the tree are buttons for "Add...", "Edit...", "Up", and "Down".

Extension Details: This section is titled "Set the properties of the selected extension. Required fields are denoted by ***". It contains two input fields: "ID:" and "Name:". Below these are three links: "Show extension point description", "Open extension point schema", and "Find declaring extension point".

At the bottom of the dialog is a tabbed interface with the following tabs: Overview, Dependencies, Runtime, Extensions (selected), Extension Points, Build, MANIFEST.MF, plugin.xml, and build.properties.

what we like about Eclipse (cont'd)



How does that translate to the web?



RAP - enabling plug-in reuse

- 70% - 90% reuse is possible
- RAP provides only a subset of RCP!
- needs separation of code that is not compatible
 - can be addressed with fragments
- application needs to become multi-user enabled

getting all the things we like about Eclipse

Nice idea – but I don' like RCP in a browser



The screenshot shows the Eclipse RAP 1.0 interface running in Mozilla Firefox. The browser address bar shows `http://localhost:9090/rap`. The main window is titled "Workbench Demo" and contains several views:

- View I:** A tree view showing a hierarchy starting with "Root", including "EclipseCon location", "Eclipse Foundation", "Innoopract Inc", "Parent 2", and "Child X - filter me!".
- View II:** A 3D pie chart titled "Revenue (in Millions)". The chart is divided into seven segments, each representing a different vehicle type. The values for each segment are: 3.85 (Classic Cars), 1.12 (Motorcycles), 0.95 (Planes), 0.66 (Ships), 0.19 (Trains), 1.80 (Trucks and Buses), and 1.02 (Vintage Cars).
- View III:** A simple tree view showing "Root".
- View IV:** A table with 7 columns and 4 rows of data.
- View V:** A table with 7 columns and 4 rows of data.

Column0	Column1	Column2	Column3	Column4	Column5	Column6
Item0-0	Item0-1	Item0-2	Item0-3	Item0-4	Item0-5	Item0-6
Item1-0	Item1-1	Item1-2	Item1-3	Item1-4	Item1-5	Item1-6
Item2-0	Item2-1	Item2-2	Item2-3	Item2-4	Item2-5	Item2-6
Item3-0	Item3-1	Item3-2	Item3-3	Item3-4	Item3-5	Item3-6

o.k. - but it does not have to be this way






Banner | click me 0 | click me 1 | click me 2




In | Out | Over | Under | Through | Perspective 1 | Perspective 2




Content

- This is Text for chapter 0.
- This is Text for chapter 1.
- This is Text for chapter 2.
- This is Text for chapter 3.
- This is Text for chapter 4.

The content goes here

0  **recep ivedik fragman**
XD 7uUQKER0  2y 
youtube.com
Recep ?vedik'in beklenen filminin fragman?.

0  **amazing guitar player**
m3qMqK7h-BA  2y 
youtube.com
...one of those that will make you say...holy %\$#^

0  **Peanut and Jeff # 2**
EpRW8jh8AqY  2y 
youtube.com
Ventriliquist

Web L&F and workbench technology



The screenshot shows a web browser window with the address bar containing `http://127.0.0.1:2869/rap?startup=mail`. The page features a blue banner with the text "Banner" and "commands" in red. Below the banner are navigation tabs: "In", "Out", "Over", "Under", "Through", "Perspective 1", and "Perspective 2". The main content area is divided into two columns. The left column is titled "Content" and lists four items: "This is Text for chapter 0.", "This is Text for chapter 1.", "This is Text for chapter 2.", and "This is Text for chapter 3.". The right column is titled "The content goes here" and displays three video thumbnails. Each thumbnail has a "rap it" button below it. The videos are: "recep ivedik fragman" (video ID: X0_7uJQKERO), "amazing guitar player" (video ID: m3qMqK7h-BA), and "Peanut and Jeff # 2" (video ID: EpRW8ih8AqY). The text "selection service" is overlaid in red on the left side of the content area.

commands

perspectives

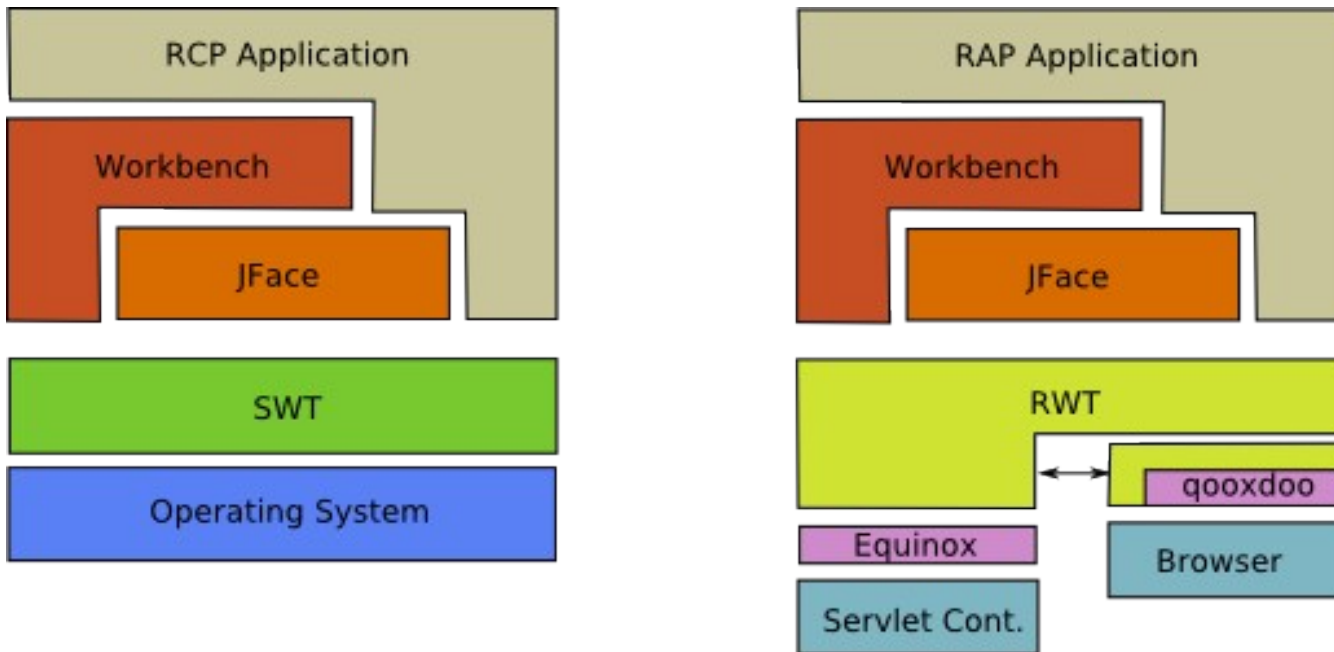
selection service

Web L&F and workbench technology (cont'd)

editors

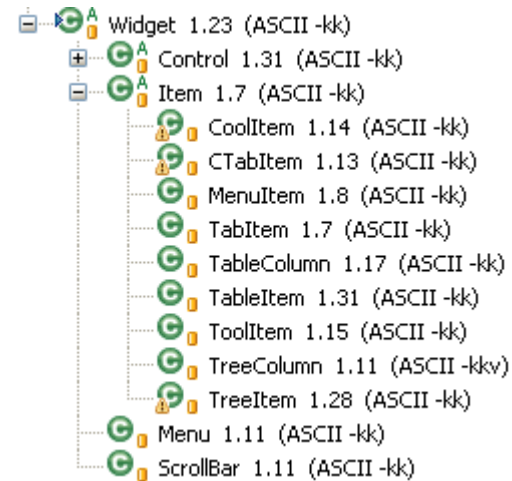
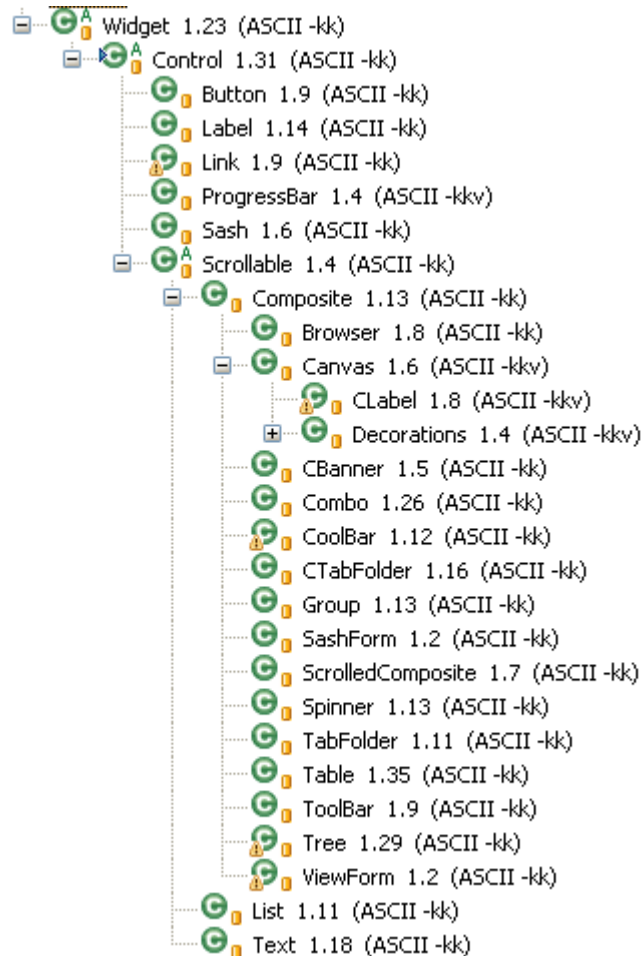
views

How does it work?



- replacing SWT with an implementation that can render to browsers
- everything else is pretty much the same
- RWT uses qooxdoo Javascript library to render widgets in the browser
- differences:
 - multi-user environment: OSGi bundles shared between sessions
 - Additional API for web specifics

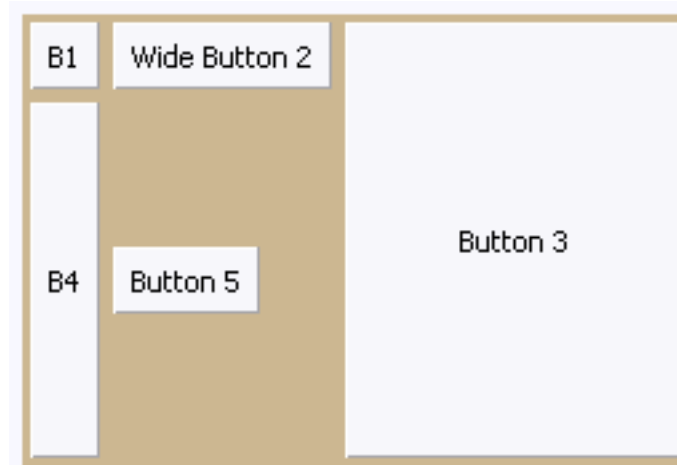
RWT widgets



RWT Layouts



- All usual layouts:
 - GridLayout,
 - RowLayout
 - FillLayout
 - FormLayout
 - StackLayout
 - and a lot more ...

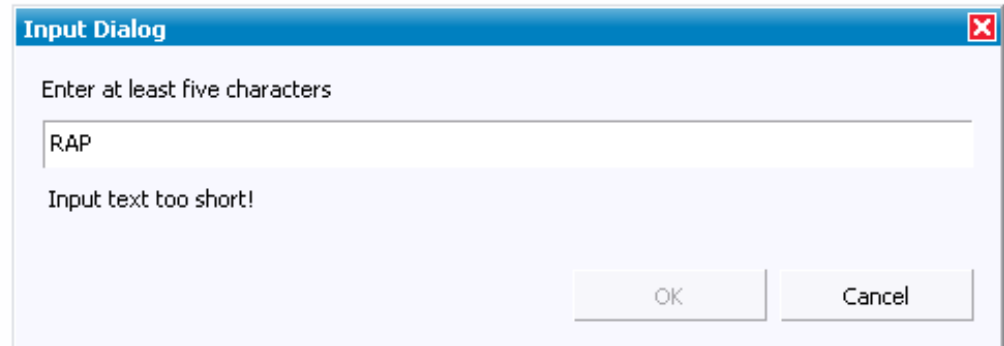


- mostly verbatim copies of SWT (OS independent)
- Layout algorithms work exactly as in SWT
- Layouts are computed on the server, e.g. after a Shell has been resized

RWT Events and Listeners



- Typed and untyped Listeners are supported
- Many Listeners implemented:
 - SelectionListener
 - ControlListener
 - ShellListener (supports doit flag now)
 - MenuListener
 - ModifyListener
 - VerifyListener
- More in future:
 - MouseListener
 - KeyListener
 - Drag/DropListener



Developer's View on a RAP Application

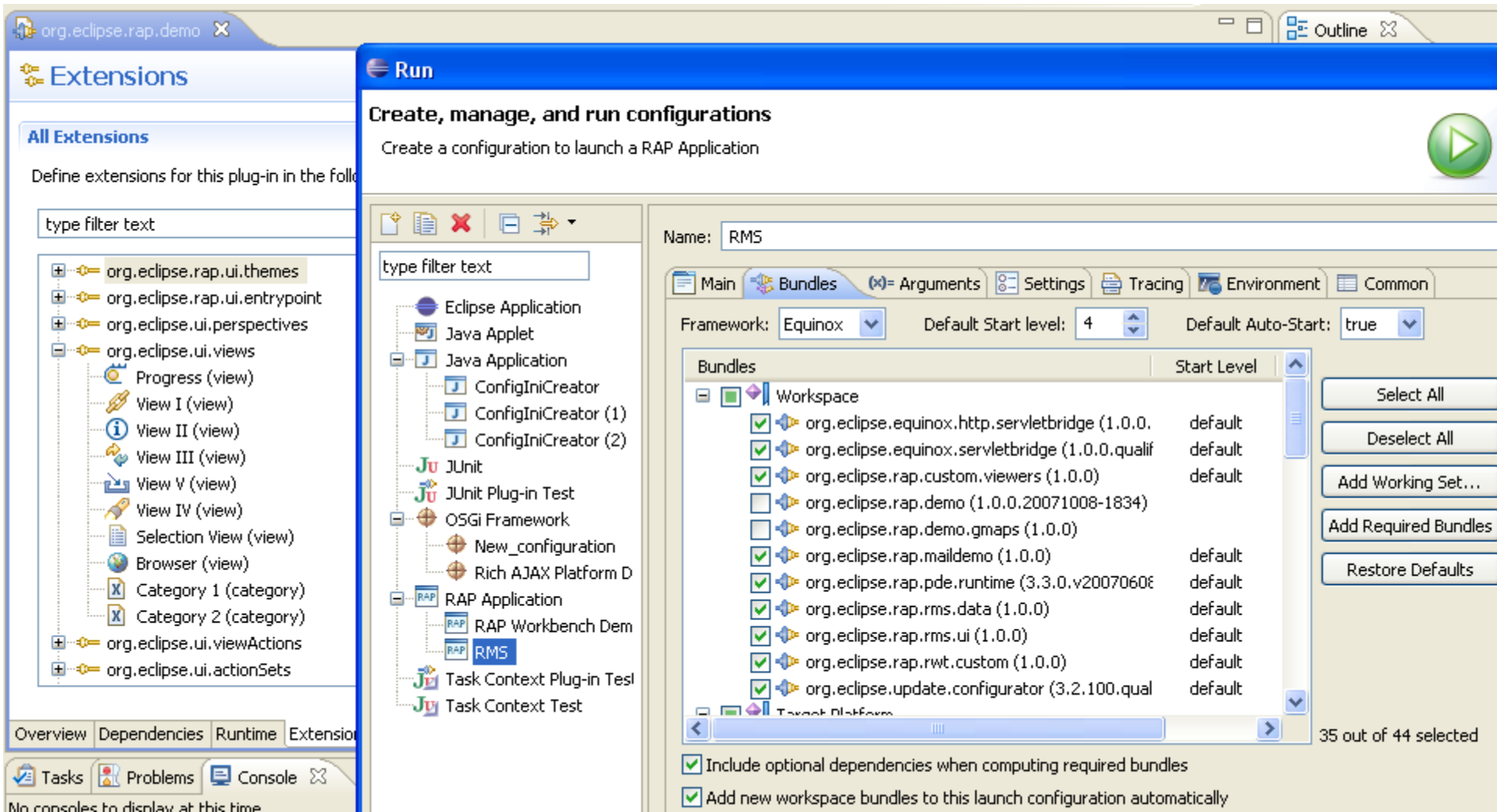


```
org.eclipse.rap.demo *DemoTreeViewPart.java x
+ * Copyright (c) 2002-2006 Innoopract Informationssysteme GmbH.
package org.eclipse.rap.demo;
+ import java.util.ArrayList;
public class DemoTreeViewPart extends ViewPart implements IDoubleClickListener {
    private TreeViewer viewer;
- public void createPartControl( final Composite parent ) {
    viewer = new TreeViewer( parent );
    viewer.setLabelProvider( new DecoratingLabelProvider( new LabelProvider(),
                                                         new LeafStarLabelDecorator() ) );
viewer.setC
viewer.
viewer.
getSite
)
- private f
- public
    if(
Press 'Ctrl+Space' to show Template Proposals
```

The AbstractTreeViewer implementation of this method checks to ensure that the content provider is an ITreeContentProvider.

- JDT: content assist, refactoring, etc., PDE, Javadoc available
- Developer does not get in touch with Javascript, CSS, HTTP ...

Developer's View on a RAP Application cont'd



The screenshot displays the Eclipse IDE interface with the 'Run' dialog box open. The dialog is titled 'Create, manage, and run configurations' and is used to launch a RAP application. The configuration name is 'RMS'. The 'Bundles' tab is selected, showing a list of bundles to be included in the launch configuration. The bundles are organized into a tree structure under the 'Workspace' folder. The 'RMS' bundle is highlighted in blue. The 'Start Level' column shows the default start level for each bundle. The 'Include optional dependencies' and 'Add new workspace bundles' checkboxes are checked.

Run Configuration Details:

- Name: RMS
- Framework: Equinox
- Default Start level: 4
- Default Auto-Start: true

Bundles	Start Level
Workspace	
<input checked="" type="checkbox"/> org.eclipse.equinox.http.servletbridge (1.0.0)	default
<input checked="" type="checkbox"/> org.eclipse.equinox.servletbridge (1.0.0.qualif	default
<input checked="" type="checkbox"/> org.eclipse.rap.custom.viewers (1.0.0)	default
<input type="checkbox"/> org.eclipse.rap.demo (1.0.0.20071008-1834)	
<input type="checkbox"/> org.eclipse.rap.demo.gmaps (1.0.0)	
<input checked="" type="checkbox"/> org.eclipse.rap.malldemo (1.0.0)	default
<input checked="" type="checkbox"/> org.eclipse.rap.pde.runtime (3.3.0.v20070606)	default
<input checked="" type="checkbox"/> org.eclipse.rap.rms.data (1.0.0)	default
<input checked="" type="checkbox"/> org.eclipse.rap.rms.ui (1.0.0)	default
<input checked="" type="checkbox"/> org.eclipse.rap.rwt.custom (1.0.0)	default
<input checked="" type="checkbox"/> org.eclipse.update.configurator (3.2.100.qual	default

35 out of 44 selected

Include optional dependencies when computing required bundles

Add new workspace bundles to this launch configuration automatically

Looks familiar, so what are the differences?



- No GC
 - Determining text size is possible by using the Graphics class
 - If you want to draw you need to implement a custom widget
- No constructors, dispose for Font, Image
 - For performance & memory consumption reasons the same fonts and Images are shared between all sessions
 - Using higher level API (JFace) abstracts the problem away, Graphics class provides factory methods

More differences – multi user



- SINGLETONS in RCP are shared between ALL users in RAP
 - RAP provides a class called `SessionSingletonBase`, that can be subclassed to provide Singletons by Session
 - `MySessionSingleton` extends `SessionSingletonBase`

```
getInstance() {  
    super.getInstance( MySessionSingleton.class );  
}
```
 - Access to `SessionSingletons` is simple within the UI thread, but need special care in background processes (jobs)

RWT Theming



- Objective: allow for a custom look of web applications
- Predefined properties of widgets can be customized
- Dimensions, Colors, Borders, Fonts, Images
- Simple Java .properties file
- Themeable custom widgets



Content
Sample Tree
Second Tree
Trallala

Content
Sample Tree
Second Tree
Trallala

NEW: individual styles



SWT Code:

```
button1.setData( WidgetUtil.CUSTOM_VARIANT, "mybutton" );
```

Theme file:

```
mybutton/button.border: 2px #169531  
mybutton/button.background: #9dd044
```

Result:



button1

Composed Widgets – just like SWT



- Composition of existing widgets
- Custom layouts, event handling
- Application developers uses composition as a control

Custom Widgets – the shortcut



Use the browser widget!

```
public YoutubeShell( final Display display ) {
    this.display = display;
    bgColor = display.getSystemColor( SWT.COLOR_BLACK );
    ...
    createShield();
    createShell();
}

public void setId( final String id ) {
    browser.setText( getHtml( id ) );
}

private static String getHtml( final String id ) {
    String html = "<html><body>"
        + "<object width=\"425\" height=\"373\">"
        + "<param name=\"movie\" value=\"http://www.youtube.com/v/\" + id"
        + "&rel=1&border=1\"></param>"
        + "<param name=\"wmode\" value=\"transparent\"></param>"
        + "<embed src=\"http://www.youtube.com/v/\" + id + "&rel=1&border=1\""
```

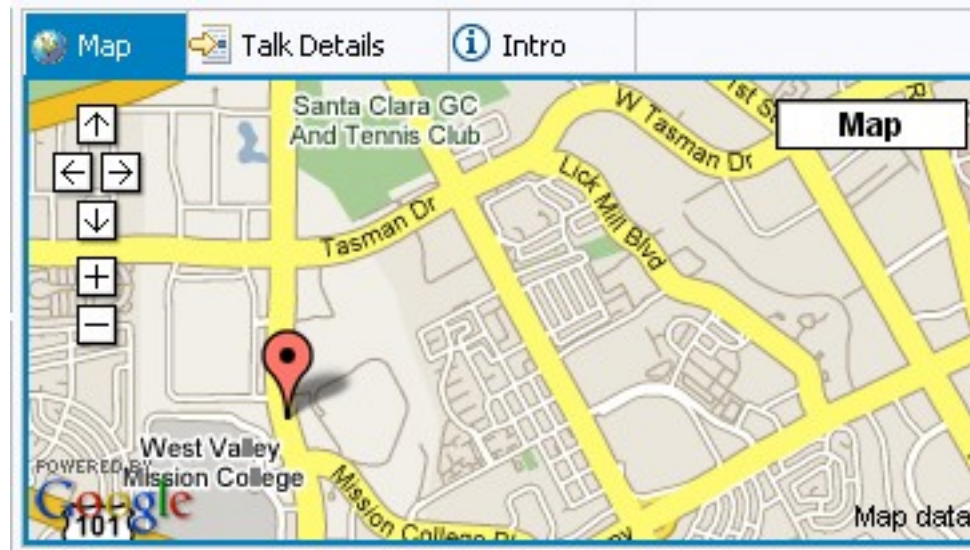
Custom Widgets – the shortcut



Custom Widgets (the real thing)



- Like in SWT, requires good knowledge of the platform
- Component developer needs Javascript, qooxdoo and RAP knowledge
- Allows do embed all sort of client side technologies: JS frameworks, Flash, Applets, ...
- Application developer simply uses Java API
- Tutorial in RAP Help



```
GMap map = new GMap( shell, SWT.NONE );  
map.setAddress( "5001 Great America Pkwy, Santa Clara" );
```


Demo



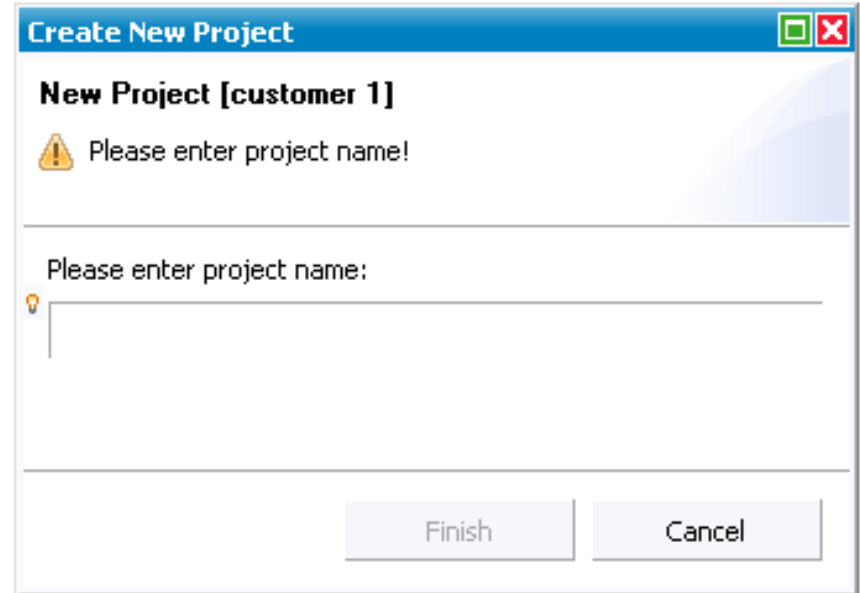
Name	First name	Company
		Focus Magazin
		Dr. Eckenstein Consult GmbH
Albrecht	Sofie	CAS Software AG
Alt	Marc	Marketing Corporation
Balducci	Graziella	Westautomatik GmbH
Baldur	Edwin-Ralf	
Baldur	Erwin-Ralf	
Bau	Tobias	Franke Unternehmensberatung
Bauer	Markus	CAS Software AG

JFace



- Support for all JFace viewers
 - TableView
 - TreeView
 - ...
- Support for most Viewer concepts
 - Provider (Content, Label, Color...)
 - Sorter
 - Filter
 - Decorator (new: Image decoration)
 -
- Support for Field Decorations
- **No Cell-Editors (yet)**

- Dialogs
 - All standard JFace dialogs like
 - ErrorDialog
 - TrayDialog
 - Support for own Dialogs
- Wizards



Workbench: Parts, Perspectives & Interaction

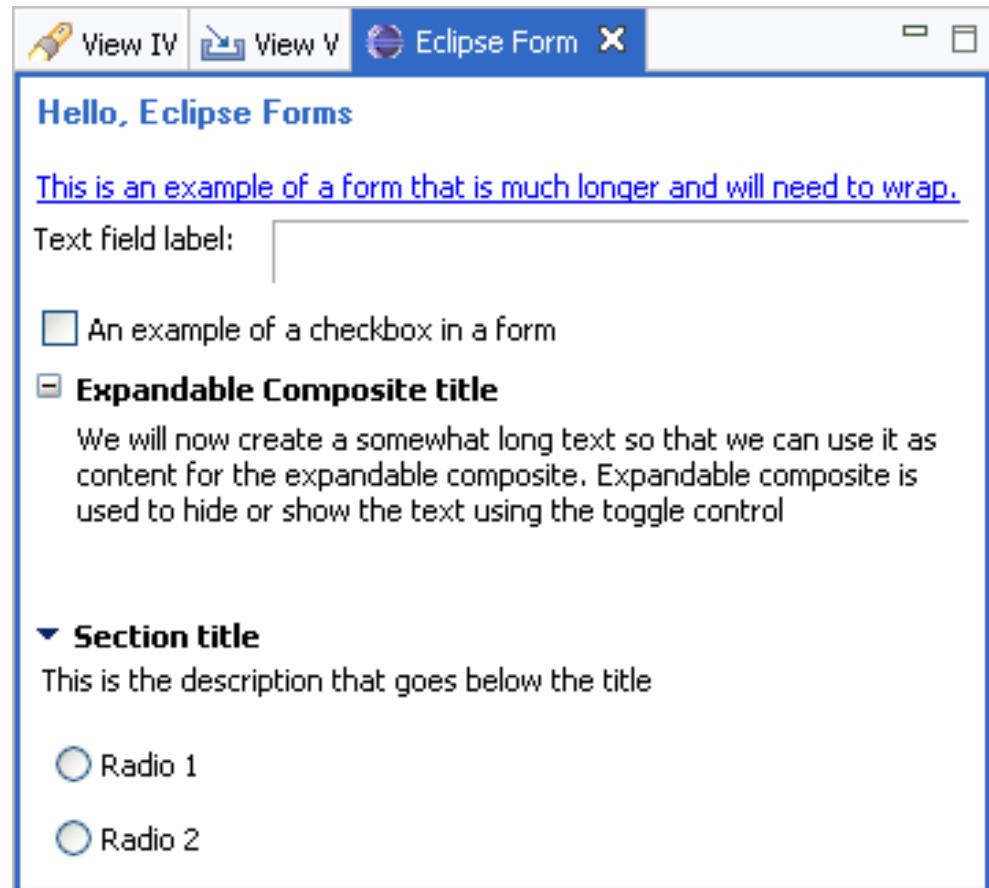


- Full support for views
 - Additional views
 - Outline
 - Properties
 - Editor support
 - Multi-page editors available
 - ISaveablePart available
 - SelectionService
 - Jobs (UIJobs, ProgressView)
 - Support for perspectives
 - Perspective Switcher
 - Extensions
 - Eclipse 3.3 Menus Framework
 - Commands & Handler
 - Expression support for visibleWhen and enabledWhen
- New: ScopedPreferenceStore**
New: Activity Support

Additional bundles



- Data Binding
- Eclipse UI Forms
- Help
- **Sandbox:**
 - Search
 - PDE Runtime





Outlook

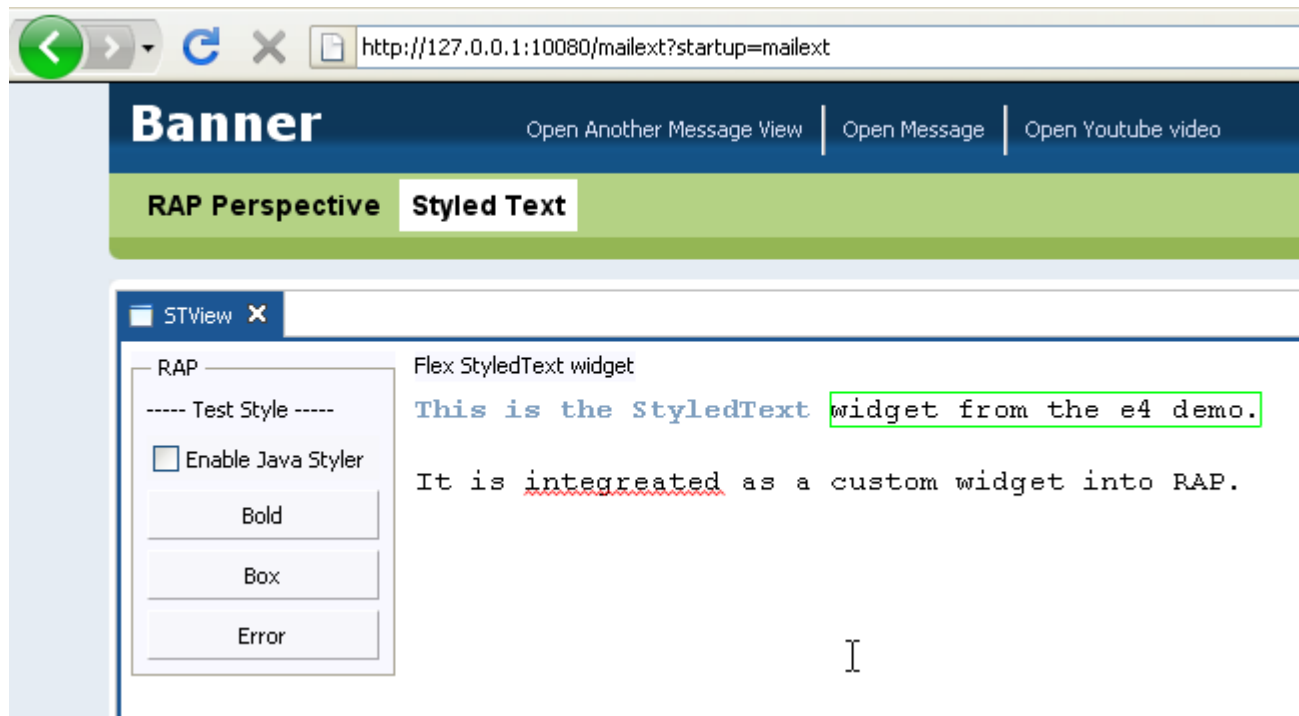
Moving to the new Eclipse RT project



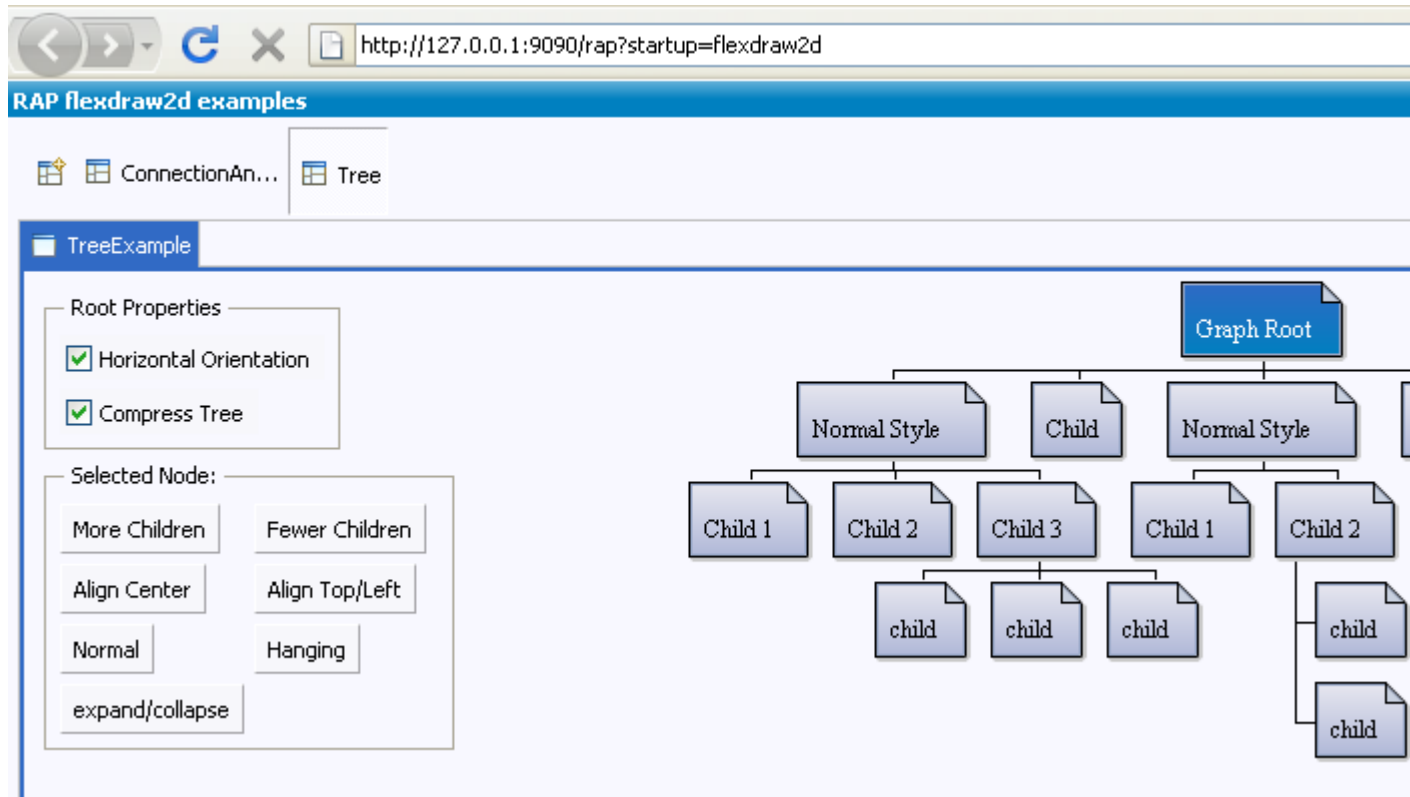
- The new Eclipse RT (Runtime, not Realtime) top level project helps to coordinate the runtime efforts
- The Equinox Community will foster and promote runtime technologies at Eclipse – independently of their location in a top level project
- RAP is one of the initial six projects of RT
- Integration with other runtime technologies is key

<http://eclipse.org/equinox-portal/>

- Platform and RAP team will work together on next generation of Eclipse platform, community is invited to join



- prototype using Flash in the browser



Browser address bar: <http://127.0.0.1:9090/rap?startup=flexdraw2d>

Page title: RAP flexdraw2d examples

Tree structure:

- Graph Root
 - Normal Style
 - Child 1
 - Child 2
 - Child 3
 - Child
 - Normal Style
 - Child 1
 - Child 2
 - child
 - child

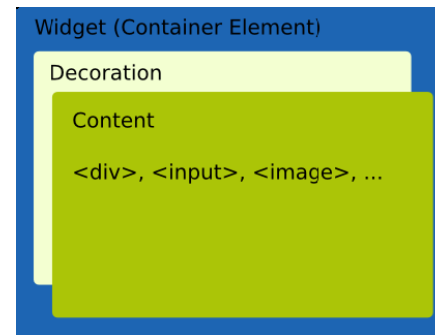
Root Properties:

- Horizontal Orientation
- Compress Tree

Selected Node:

- More Children
- Fewer Children
- Align Center
- Align Top/Left
- Normal
- Hanging
- expand/collapse

- Decoration
 - Customization mechanism of the widget's look & feel
 - support of rounded corners, shadow, etc.
 - widgets can have different decoration renderers



Get the RAP - <http://eclipse.org/rap>



Demos

See some demos here

Downloads

Get the latest RAP release

The RAP project enables developers to build rich, Ajax-enabled Web applications by using the Eclipse development model, plug-ins with the well known Eclipse workbench extension points, JFace, and a widget toolkit with SWT API (using **qooxdoo** for the client-side presentation). The project has graduated from incubation and released its 1.0 release.

[Learn more ...](#)

References



- <http://www.eclipse.org/rap> - RAP project page
- <http://wiki.eclipse.org/RAP> - RAP project wiki
- <http://www.qooxdoo.org> - qooxdoo js library