

# Eclipse 4.0 and e4

Boris Bokowski, IBM



# Background



# eclipse


## 4.0 EARLY ADOPTER RELEASE

(c) Copyright Eclipse contributors and others, 2000, 2010. All rights reserved. Eclipse is a trademark of the Eclipse Foundation, Inc. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective



# Deep Dive

# Background

- 
- e4 is a place for incubation, not a product
  - Platform needs to innovate to remain viable

# Competition



Microsoft®  
Silverlight™

# Why change?

“We’ve already built all our plug-ins.

The most important thing is  
**don’t break us.”**

(Yes, there is a compatibility layer.)



Francois Schnell, <http://www.flickr.com/photos/frenchy/30217773/>

# Goals of e4 Project

- Place for innovation
- Use e4 technologies as the basis of Eclipse 4.0 (and some will show up / are showing up in 3.x)
- Engage open source community
- Build a better Eclipse
  - Make it more flexible and easier to use
  - Prepare for web technologies



# Themes

- Making it easier to develop UI components
  - Programming model (DI, Services)
  - OpenSocial Gadgets
  - Declarative UI (XWT)
- Making it easier to assemble and style applications
  - Modeled Workbench
  - CSS styling
- Others: Flexible Resources (in 3.6), JavaScript debugging (in JSDT), ...

# Work Areas

- Modeled Workbench 4.0
- Renderer Framework 4.0
- CSS Styling 4.0
- Dependency Injection 4.0
- Application Services 4.0, work ongoing
- OpenSocial Gadgets not graduated, in 4.1 plan
- Flexible Resources 3.6
- Declarative UI (XWT) not graduated
- Declarative UI (TM) not active
- REST-based Resources not graduated
- Deep Linking, Easy Install recent contribution

# Eclipse 4.0 SDK Early Adopter Release



Java - org.eclipse.e4.ui.workbench/src/org/eclipse/e4/ui/internal/workbench/handlers/SaveHandler.java - Eclipse SDK

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access

Package Explorer

- ResourceHandler.java 1.5
- SelectionLookupFunction
- SelectionServiceCreationF
- SelectionServiceImpl.java
- Trackable.java 1.1
- UIEventPublisher.java 1.1
- Util.java 1.1
- WorkbenchLogger.java 1
- XMLModelReconciler.java
- > org.eclipse.e4.ui.internal.wc
  - ExitHandler.java 1.5
  - SaveAllHandler.java 1.5
  - > SaveHandler.java 1.6
  - UnimplementedHandler.j
- org.eclipse.e4.ui.workbench
  - IExceptionHandler.java 1.
  - IModelResourceHandler.j
  - IPresentationEngine.java
  - IResourceUtilities.java 1.1
  - IWorkbench.java 1.1
  - UIEvents.java 1.5
- org.eclipse.e4.ui.workbench.l
- org.eclipse.e4.ui.workbench.r
- META-INF
- OSGI-INF
- schema
- about.html 1.1
- build.properties 1.7
- plugin.properties 1.1

\*SaveHandler.java

```

* Copyright (c) 2009, 2010 IBM Corporation and
package org.eclipse.e4.ui.internal.workbench.ha

import javax.inject.Named;

/**
 *
 */
public class SaveHandler {

    @CanExecute
    boolean canExecute(@Named(IServiceConstants
        return dirtyable == null ? false : dirt
    }

    @Execute
    void execute(EPartService partService, @Nam
        partService.savePart(part, false);
    }
}

```

Outline

- org.eclipse.e4.ui.inte
- import declarations
- > SaveHandler 1.6
  - canExecute(MDi
  - execute(EPartSei

Problems @ Javadoc Declaration

Description	Resource	Path	Location	Type
Warnings (100 of 393 items)				

Writable Smart Insert 22:4

Package Explorer

- MyProject
  - JRE System Library [J2SE-1.5]
  - Plug-in Dependencies
  - src
    - myproject
      - Activator.java
    - myproject.editors
      - ColorManager.java
      - IXMLColorConstants.java
      - NonRuleBasedDamagerRepa
      - TagRule.java
      - XMLConfiguration.java
      - XMLDocumentProvider.java
      - XMLDoubleClickStrategy.jav
      - XMLEditor.java
      - XMLPartitionScanner.java
      - XMLScanner.java
      - XMLTagScanner.java
      - XMLWhitespaceDetector.java
  - icons
  - META-INF
  - build.properties
  - plugin.xml

```

package myproject;

import org.eclipse.jface.resource.ImageDescriptor;

/**
 * The activator class controls the plug-in life cycle
 */
public class Activator extends AbstractUIPlugin {

    // The plug-in ID
    public static final String PLUGIN_ID = "MyProject"; //$NON-NLS-1$

    // The shared instance
    private static Activator plugin;

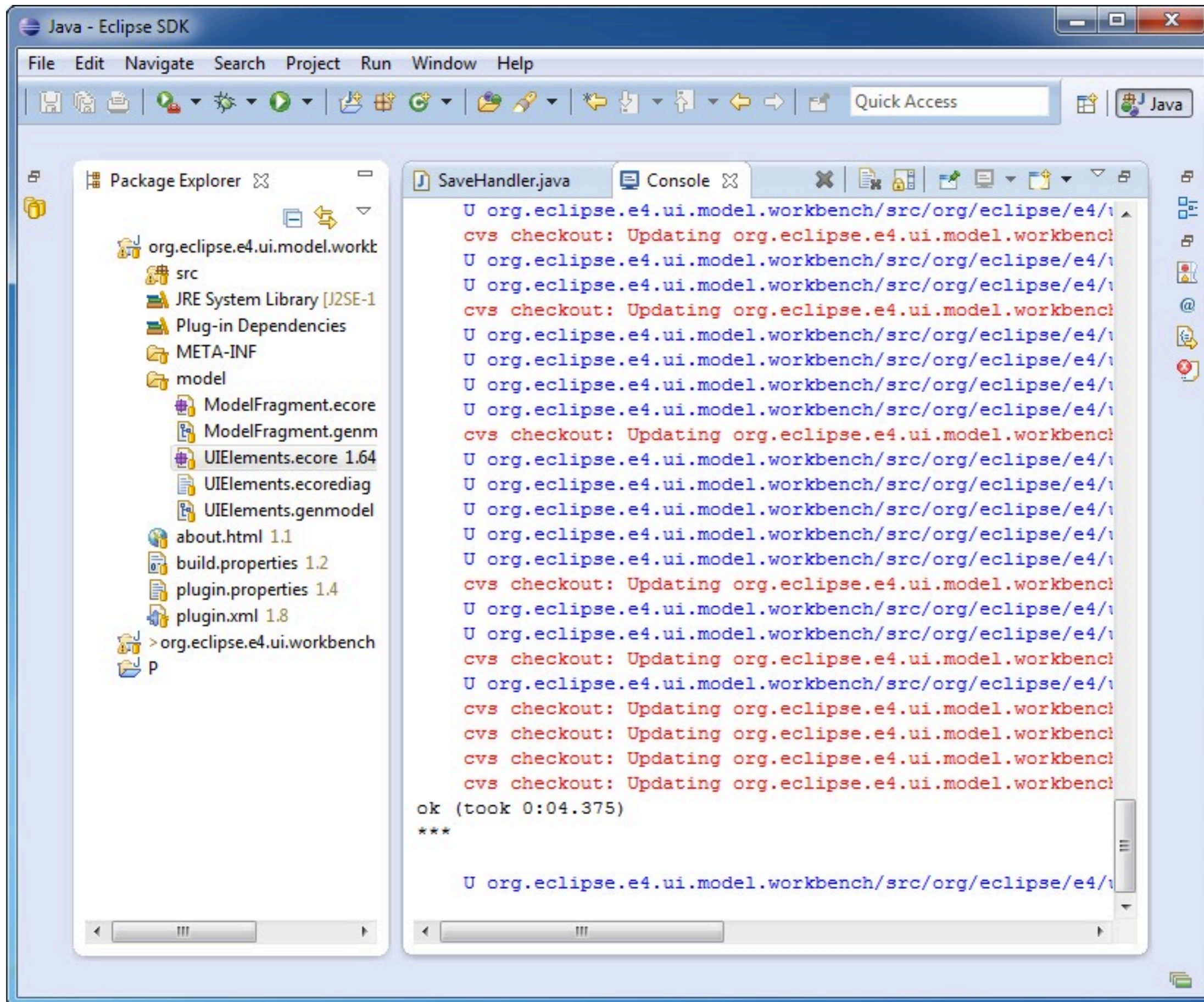
    /**
     * The constructor
     */
    public Activator() {
    }

    /**
     * (non-Javadoc)
     * @see org.eclipse.ui.plugin.AbstractUIPlugin#start(org.osgi.framework.BundleContext)
     */
    public void start(BundleContext context) throws Exception {
        super.start(context);
        plugin = this;
    }
}
    
```

Problems | Javadoc | Declaration

Description	Resource	Path	Location	Type
Warnings (4 items)				

# Demo



# Eclipse 4.0 SDK Early Adopter Release

Rest of Platform:  
Debug,  
Text,  
Team,  
...

JDT:  
Java Editor,  
Java Debugging,  
Refactoring,  
...

PDE:  
Manifest Editor,  
Launching,  
API Tooling,  
...

Workbench (Compatibility, provides 3.x APIs)

Modeled UI, CSS styling, Dependency Injection, Application Services

Eclipse 4  
Application Platform

Equinox



EMF Core



SWT, JFace




Java Virtual Machine

OS (Windows, Mac, Linux)



# Deep Dive

A diver in a black wetsuit and mask is shown underwater. The diver is wearing a black wetsuit with a yellow patch on the left chest. The diver's mask is clear, and they are looking directly at the camera. The background is a hazy, greenish-blue underwater environment. Overlaid on the image is a quote in white text.

“Think of e4 as RCP 2.0, simplified”  
-- Chris Aniszczyk

star5112, <http://www.flickr.com/photos/johnjoh/131767717/>

Main

File Theme

Contacts List

First Name	Last Name
Chris	Aniszczyk
Boris	Bokowski
Peter	Friese
Kevin	McGuire
Tom	Schindl
Hallvard	Traetteberg
Kai	Tödter
Yves	YANG
Angelo	ZERR

Details


**General**

Full Name: Kai Tödter

Company: Siemens AG, CT SE 2

Job Title: Dipl.-Inform.

Note: Eclipse RCP Expert and OSGi Enthusiast



**Business Address**

Street: Otto-Hahn-Ring 6

City: Munich

ZIP: 81739

State/Prov:

Country: Germany

**Business Phones**

Phone: +49 (89) 63641064

Mobile: +49 (173) 7068910

**Business Internet**

Email: kai.toedter@siemens.com

Web Page: http://www.ct.siemens.com



Library Create Album

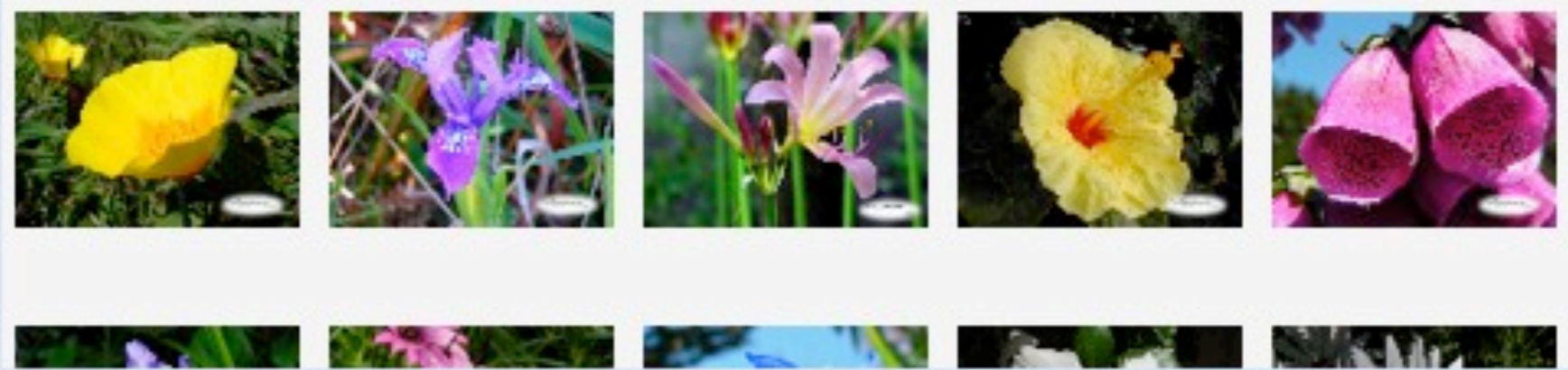
- ▼ Album 1 (0,10 KB)
  - .project (0,20 KB)
  - californiapoppies2.jpg (111 KB)
  - glowingiris.jpg (148,05 KB)**
  - lilies3.jpg (98,58 KB)
  - paleyellowhibiscus.jpg (94 KB)
  - purplebells2.jpg (142,70 KB)
  - purpleiris4.jpg (123,24 KB)
  - purplestar.jpg (151,62 KB)
  - spikyblueflower.jpg (122,5 KB)
  - whitehibiscus2.jpg (94,29 KB)
  - whitestar1.jpg (133,78 KB)
- ▶ Album 2 (0,10 KB)



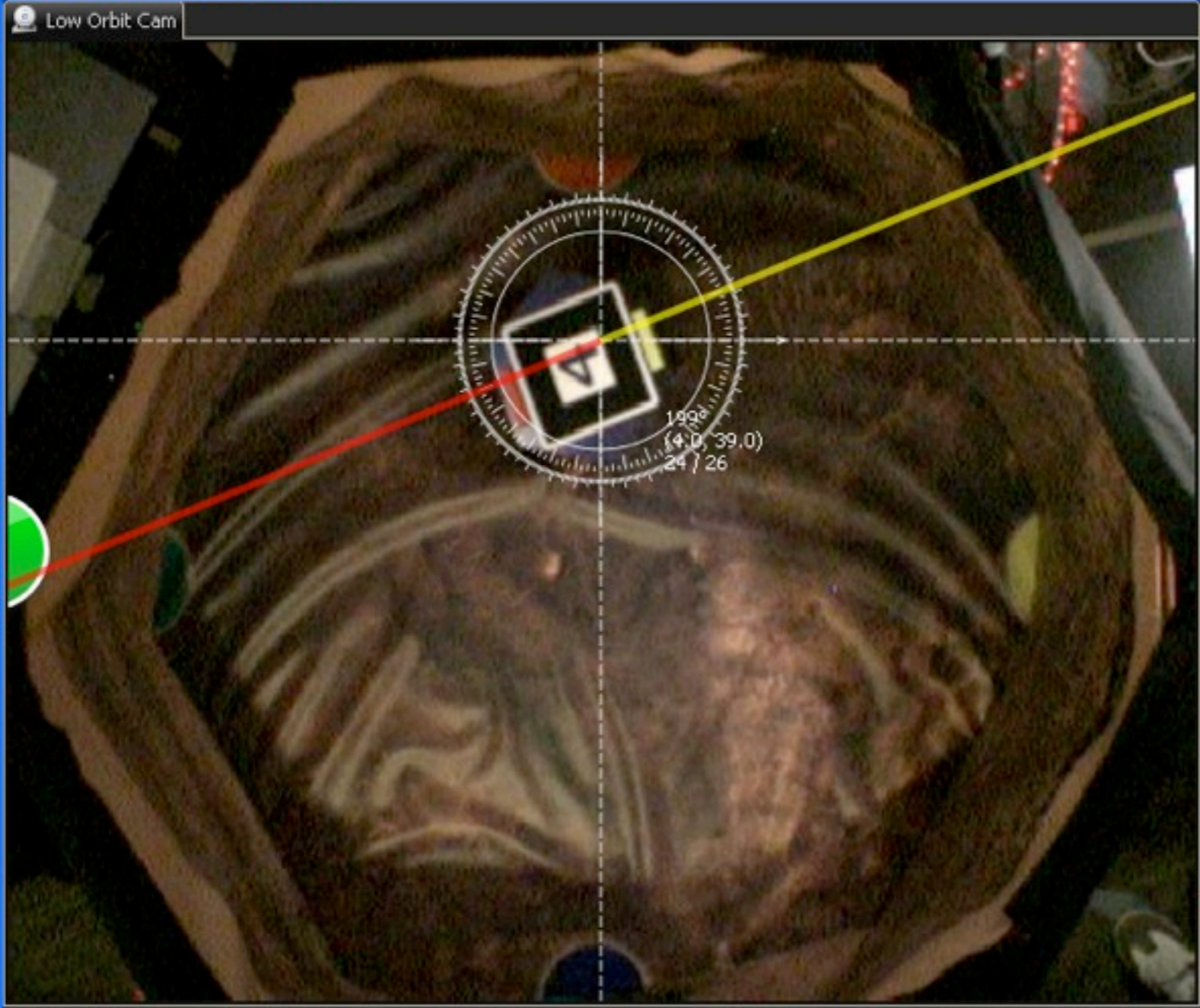
Location



Thumbnails Exif



# Mars Rover Mission Control



Control Controller Queue

L Wheel 24 R Wheel 26

L Cmd: 24 R Cmd: 26 Duration: 4.6

This control panel features a diamond-shaped grid of cells. The top half of the grid is green, and the bottom half is red. A green arrow points up from the top, and a red arrow points down from the bottom. A red 'X' is located in the center of the grid. On the left and right sides of the grid, there are circular icons with arrows.

Telemetry

Orientation		Wheel Velocities		Systems
(4, 39) Position	199° Heading	24 L Velocity	26 R Velocity	7353 Battery

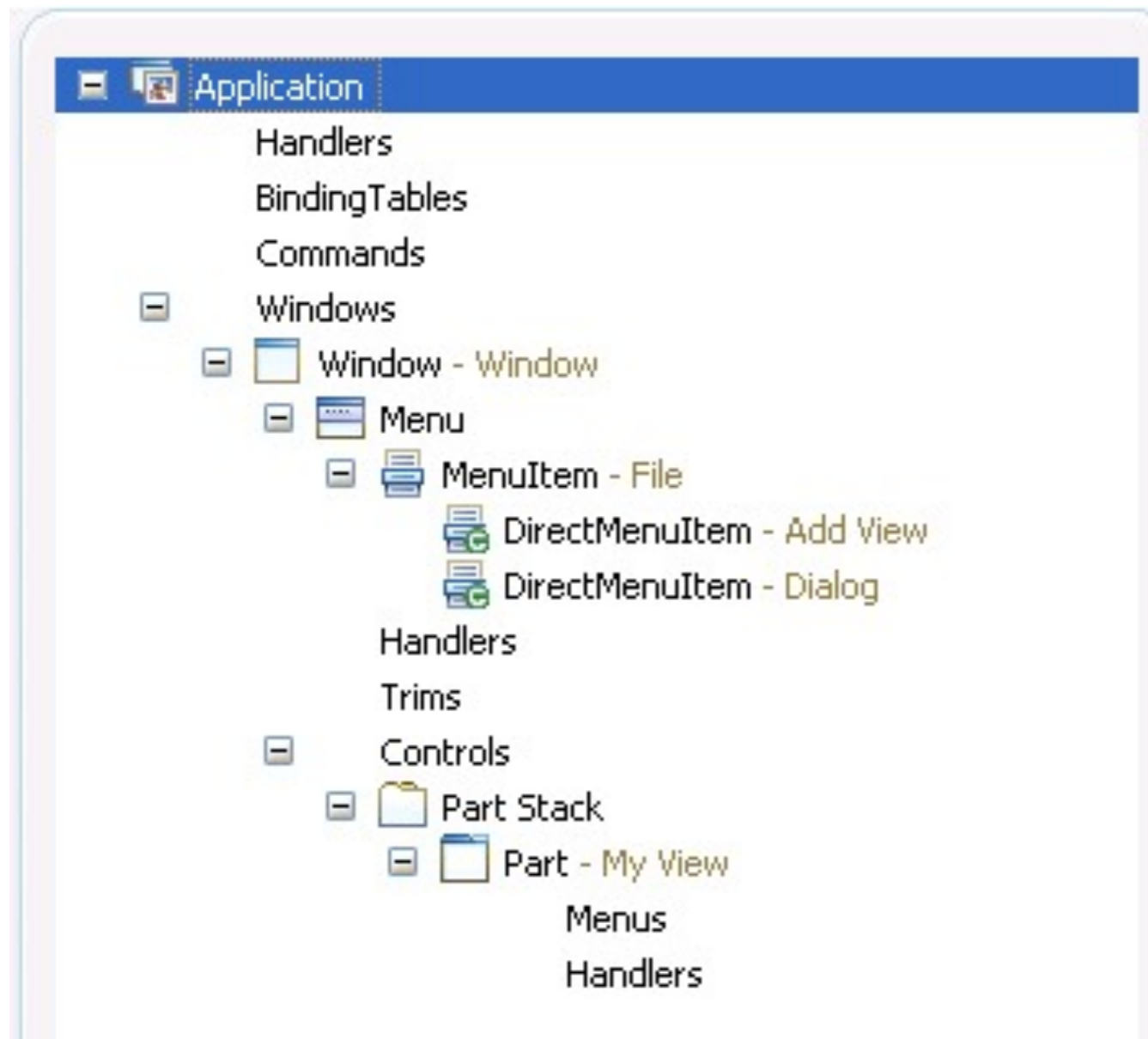
The telemetry section contains three main panels: Orientation, Wheel Velocities, and Systems. Each panel has a representative icon and a numerical value.

Mission

Instrument	Target		Mission Status	
			Controller:	roggram
Drill	Humphrey	Mimi	Controller score:	0
			Mission potential:	81
Spectro	Mazatzal	Adirondack	Seconds remaining:	77

The mission panel displays various instrument and target icons. The 'Mission Status' section shows the current controller, score, potential, and remaining time.

# Workbench Model



- Workbench window
  - Menu with menu items
  - Window Trim, e.g. toolbar with toolbar items
  - Parts Sash Container
    - Parts
  - Part Stack (CTabFolder)
    - Parts
  - Handlers
  - Key Bindings
  - Commands

# Models Application Structure

The image shows two parts of an IDE interface. On the left is a project explorer showing a tree structure under 'Application'. The 'Part Stack' folder is expanded, and 'Part - My View' is highlighted with a red box. A callout bubble points to it with the text 'view in model'. On the right is a properties view for the selected 'Part'. The 'Class URI' field is highlighted with a red box and has a callout bubble pointing to it with the text 'URL points to implementation'. Other fields include 'Id' (de.vogella.e4.rcp.modelcontribution.parts.View1), 'Label' (My View), 'Icon URI', 'Closeable' (checkbox), and 'Bindings'. A callout bubble points to the 'Label' field with the text 'attributes of view'.

view in model

attributes of view

URL points to implementation



**CODE EXAMPLES**

rs-photo, <http://www.flickr.com/photos/rs-foto/2129343084/sizes/l/>



# CSS

```
.MPartStack {
    tab-renderer: url('platform:/plugin/.....');
    unselected-tabs-color: #FFFFFF #FFFFFF #FFFFFF 100% 100%;
    outer-keyline-color: #FFFFFF;
        inner-keyline-color: #FFFFFF;
        font-size: 9;
        font-family: 'Segoe UI';
        simple: true;
}

.MTrimBar {
    background-color: #E1E6F6;
}

.MTrimBar#org-eclipse-ui-main-toolbar {
    background-image: url(../win7.PNG);
}

.MPartStack.active {
    unselected-tabs-color: #F3F9FF #D0DFEE #CEDDED #CEDDED #D2E1F0 #D2E1F0 #FFFFFF 20% 45%
60% 70% 100% 100%;
    outer-keyline-color: #B6BCCC;
```

# Parts (Views/Editors) in e4

```
public class MyView {  
    @Inject Composite parentComposite;  
  
    @Inject ...  
  
    @PostConstruct void init() {  
        // create part controls  
    }  
  
    @EventHandler(TOPIC_NAME) {  
        ...  
    }  
}
```

# Handlers in e4

```
public class MyHandler {  
    @CanExecute  
    boolean canExecute(...) {  
        ...  
    }  
    @Execute  
    void execute(...) {  
        ...  
    }  
}
```

# Dependency Injection

- Use JSR-330 annotations and interfaces
  - @Inject
  - @Qualifier, @Scope
  - Provider
  - plus @PostConstruct, @PreDestroy  
(from javax.annotation)
- and custom annotations such as
  - @Preference
  - @EventHandler
  - @CanExecute

# Access preference values

```
IPreferenceStore store =  
    IDEWorkbenchPlugin.getDefault()  
        .getPreferenceStore();  
  
boolean saveBeforeBuild = store  
    .getBoolean(SAVE_BEFORE_BUILD);
```

3.x

```
@Inject  
@Preference(SAVE_BEFORE_BUILD)  
boolean saveBeforeBuild;
```

4.x

# Print a message on the status line

```
getViewSite()  
    .getActionBars()  
        .getStatusLineManager()  
            .setMessage(msg);
```

**3.x**

```
@Inject  
IStatusLineManager statusLine;  
...  
statusLine.setMessage(msg);
```

**4.x**

# Associate help context with control

```
getSite()  
    .getWorkbenchWindow()  
    .getWorkbench()  
    .getHelpSystem().setHelp(  
        viewer.getControl(), some_id);
```

3.x

```
@Inject  
IWorkbenchHelpSystem helpSystem;  
...  
helpSystem.setHelp(  
    viewer.getControl(), some_id);
```

4.x

# React to errors/exceptions

```
catch(Exception ex) {  
    Istatus status = new Status(  
        Istatus.ERROR, "plugin-id",  
        "Error while writing file", ex);  
    StatusManager.getManager()  
        .handle(status, StatusManager.SHOW);  
}
```

3.x

@Inject

```
StatusReporter statusReporter;
```

...

```
catch(Exception ex) {  
    statusReporter.show(  
        "Error while writing file", ex);  
}
```

4.x



# Services

- All OSGi services are available through DI
  - e.g.: IWorkspace
  - Use OSGi DS to add your own:
    - Database connection
- Eclipse Application Services, e.g.
  - StatusReporter
  - IStatusLineManager
  - IWorkbenchHelpSystem

# Eclipse Application Services (“Twenty Things”)

Editor lifecycle

Long-running operations

Receiving input

Progress reporting

Producing selection

Error handling

Standard dialogs

Navigation model

Persisting UI state

Resource management

Logging

Status line

Interface to help system

Drag and drop

Menu contributions

Undo/Redo

Authentication

Accessing preferences

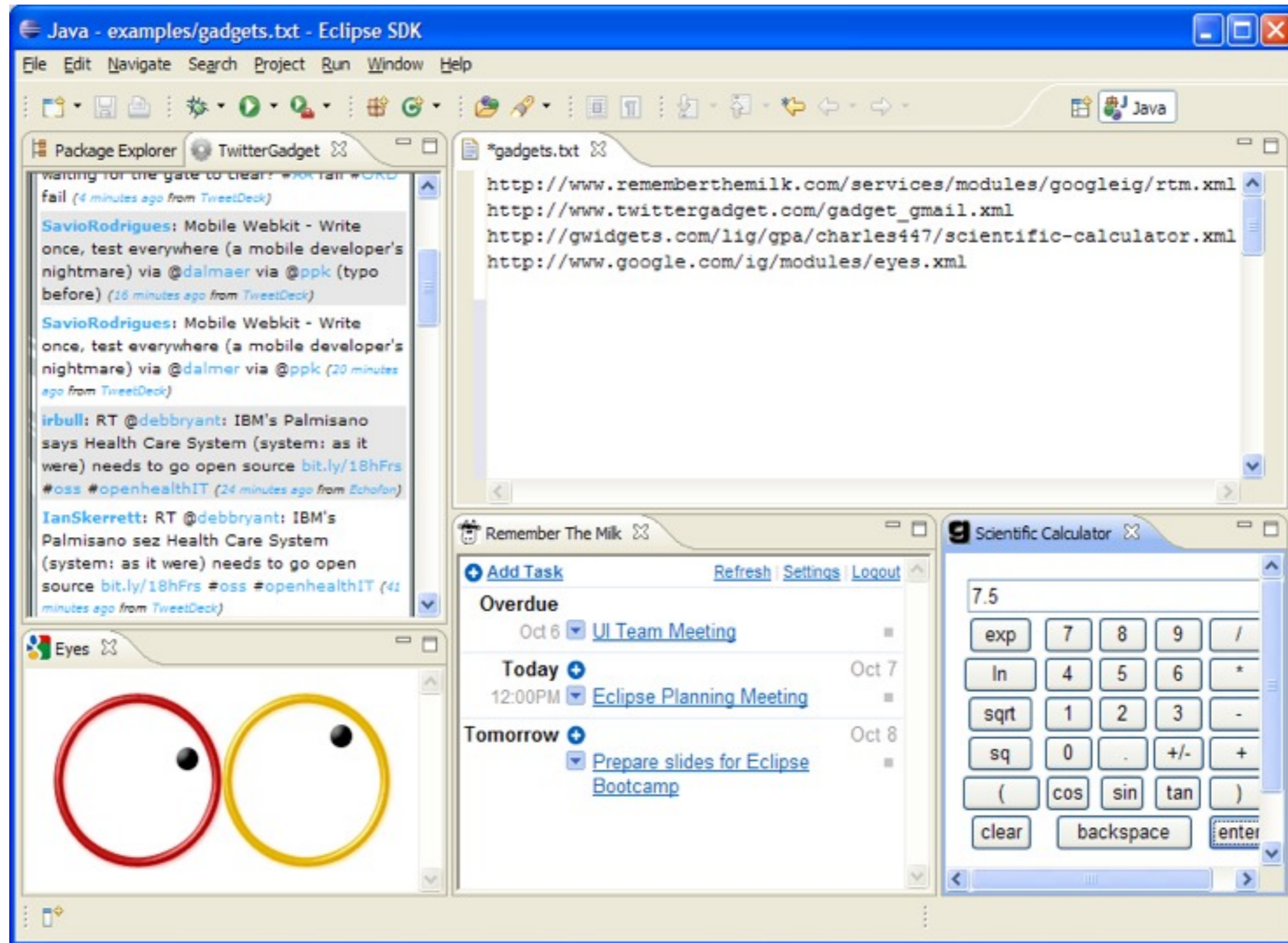
Authorization

# XWT

- Remains in Incubator for now
- Similar to Microsoft's XAML

```
<TableView input="{Binding Path=employees}">
  <TableView.columns>
    <TableViewColumn width="150" text="Name"
                      displayMemberPath="name"/>
    <TableViewColumn width="150" text="Age"
                      displayMemberPath="age"/>
  </TableView.columns>
  <TableView.control.layoutData>
    <GridData horizontalAlignment="FILL"
              grabExcessHorizontalSpace="true"/>
  </TableView.control.layoutData>
</TableView>
```

# OpenSocial Gadgets



# Questions?

[www.eclipse.org/eclipse4](http://www.eclipse.org/eclipse4)

# Legal Notice

- Copyright © IBM Corp., 2007-2010. All rights reserved. This presentation and the source code in it are made available under the EPL, v1.0.
- Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- Eclipse and the Eclipse logo are trademarks of Eclipse Foundation, Inc.
- IBM and the IBM logo are trademarks or registered trademarks of IBM Corporation, in the United States, other countries or both.
- THE INFORMATION DISCUSSED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, AND IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, SUCH INFORMATION. ANY INFORMATION CONCERNING IBM'S PRODUCT PLANS OR STRATEGY IS SUBJECT TO CHANGE BY IBM WITHOUT NOTICE