



the
POWER
of
JAVA™



JavaOne
Part of the Oracle and Sun Microsystems

Building Applications for Multiple Platforms Using the Eclipse Rich Client Platform

Wayne Beaton
Evangelist

<http://eclipse.org/evangelism>

<http://wbeaton.blogspot.com>

The Eclipse Foundation

TS-3422

Agenda

What is Eclipse?

What is a Rich Client?

What is Eclipse Rich Client Platform?

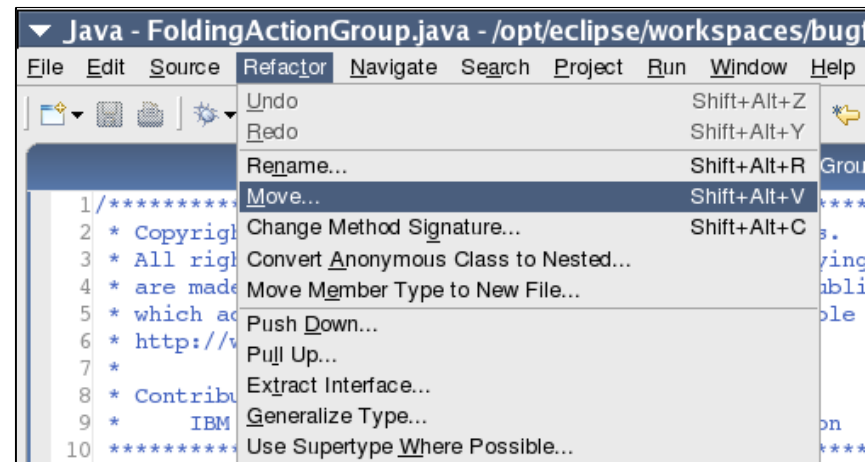
Deployment

Wrap-Up

What Is Eclipse ?

Eclipse Is a Java™ IDE

- Widely regarded as the Java development environment
- With all the bells and whistles...
 - Language-aware editors, views...
 - Refactoring support
 - Integrated unit testing and debugging
 - Incremental compilation and build
 - Team development support
 - Out of the box support for CVS

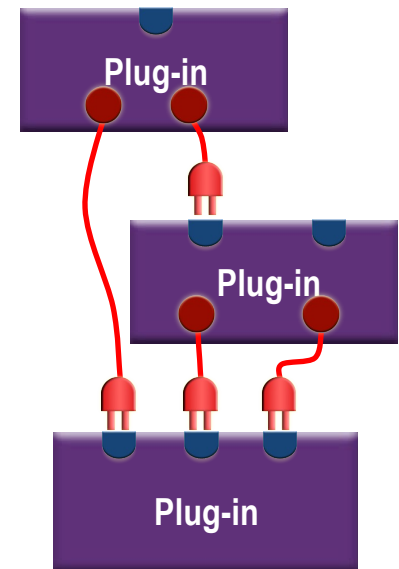


Eclipse Is an IDE Framework

- Eclipse + JDT = Java IDE
 - First class framework for Java technology
 - Language aware editor
 - Incremental build
 - Integrated debugging
- Eclipse + CDT = C/C++ IDE
 - First class framework for C/C++
 - Language aware editor
 - Refactoring, search
- Eclipse + PHP = PHP IDE
- Eclipse + JDT + CDT + PHP = Java technology, C/C++, PHP IDE

Eclipse Is a Tools Framework

- Extensibility through OSGi implementation
 - Plug-ins make Eclipse whatever you need it to be
- Focus on developing a universal platform of frameworks and exemplary tools
- Tools extend the Eclipse platform using plug-ins
 - Business Intelligence and Reporting Tools (BIRT)
 - Eclipse Communications Framework (ECF)
 - Web Tools Project (WTP)
 - Eclipse Modelling Framework (EMF)
 - Graphical Editing Framework (GEF)
 - Test and Performance Tooling Project (TPTP)



Eclipse Is a Application Framework

- Remove the IDE elements, Java language support, team development support, ...and you are left with a pretty comprehensive general application framework
 - Support for multiple platforms
 - Linux, Windows, Mac OSX, UNIX, embedded
 - Rich widget set, graphics
 - Native-OS integration (drag and drop, OLE/XPCOM integration)
- A platform for rich clients
- Some examples of this later...

Eclipse Is an Open Source Enabler

- Nine (9) top level projects
 - The Eclipse Project
 - Tools
 - Web Tools Platform
 - Test and Performance Tools Platform
 - Business Intelligence and Reporting Tools
 - Data Tools Platform
 - Device Software Development Platform
 - SOA Tools Platform
 - Technology (Incubators)
- Dozens of sub projects

Eclipse Is an Open Source Community

- All Eclipse projects are available for free download
 - No registration, no tracking, no sales calls,...
- Contributors number in the hundreds
- Hundreds of plug-ins provided by commercial companies, organizations, and individuals
- Enthusiastic blogger community
 - www.planeteclipse.org
- Numerous Eclipse portals, etc...
 - EclipseZone (<http://eclipsezone.com>)
 - EclipseWiki (<http://eclipsewiki.editme.com>)
 - Eclipse Plug-in Central (<http://eclipseplugincentral.com>)
 - Others (<http://eclipse.org/community>)

Eclipse Is an Eco-System

- Eclipse is focused on nurturing the eco-system to complement, and enhance the Eclipse Platform
- 98 members, including major Java technology, Linux and Embedded vendors
 - BEA, Borland, JBoss, IBM, SAP, RedHat, Novell, Monta Vista, Wind River, Mentor, ENEA, QNX
- 40+ open source projects
- 50 million download requests to date

Eclipse Is a Foundation

- The Eclipse Foundation
 - Was created to manage and direct the ongoing development of the Eclipse open source software project
 - Is responsible for providing IT infrastructure required by development teams
- Independent not-for-profit Foundation formed in 2004



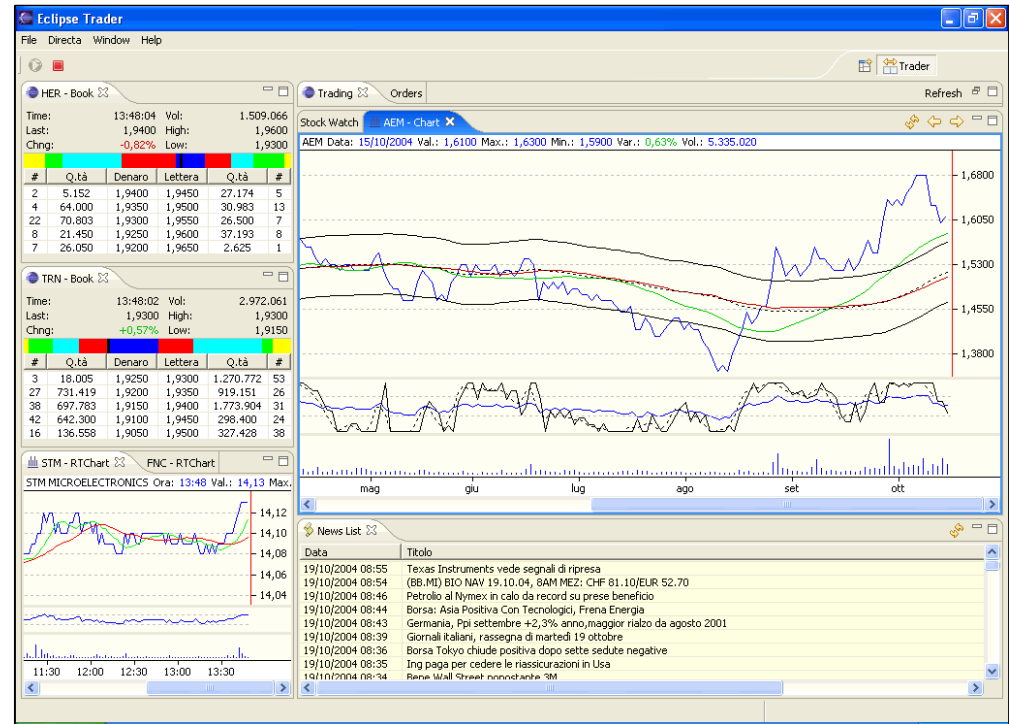
Eclipse Is All These Things...

- A Java IDE
- An IDE Framework
- A Tools Framework
- An Application Framework
- An Open Source Enabler
- A community
- An eco-system
- A foundation

What Is Rich Client?

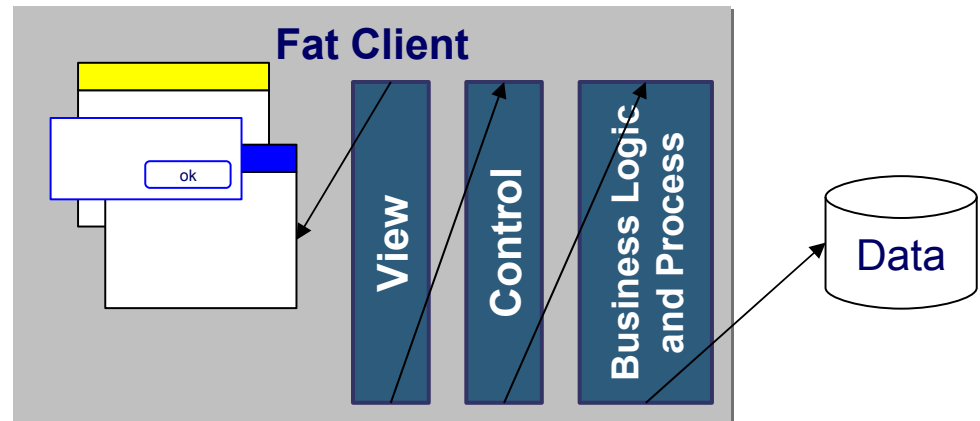
What Is a Rich Client?

- An application that uses the windowing and GUI features of the operating system
 - Native widgets
 - Drag and drop
 - Integration with platform component model
 - Much, much, more...



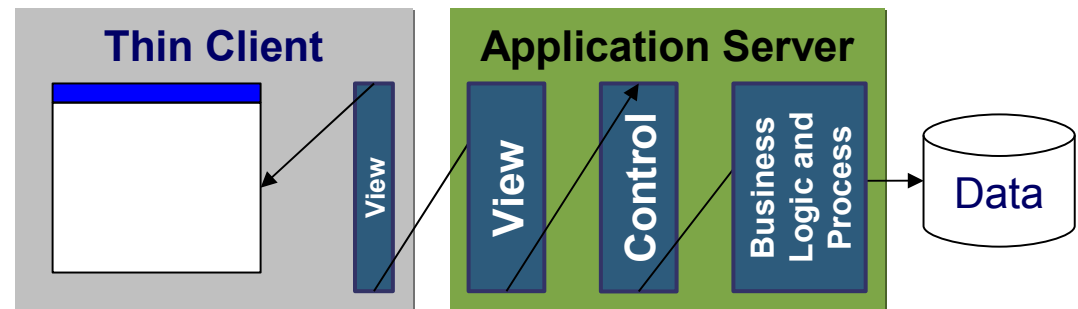
Evolution of the Client: Fat Client

- All application logic on the client
- Relatively difficult to update
- Rich user experience
- Monolithic application
- Platform dependent



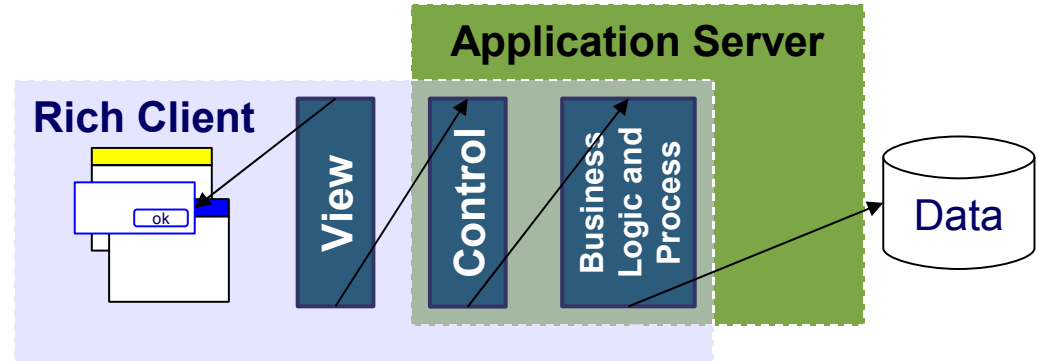
Evolution of the Client: Thin Client

- All application logic on the server
- Relatively easy to update
- Massively concurrent application
- Very simplistic user experience
- Client platform independent
 - Clients run in browser



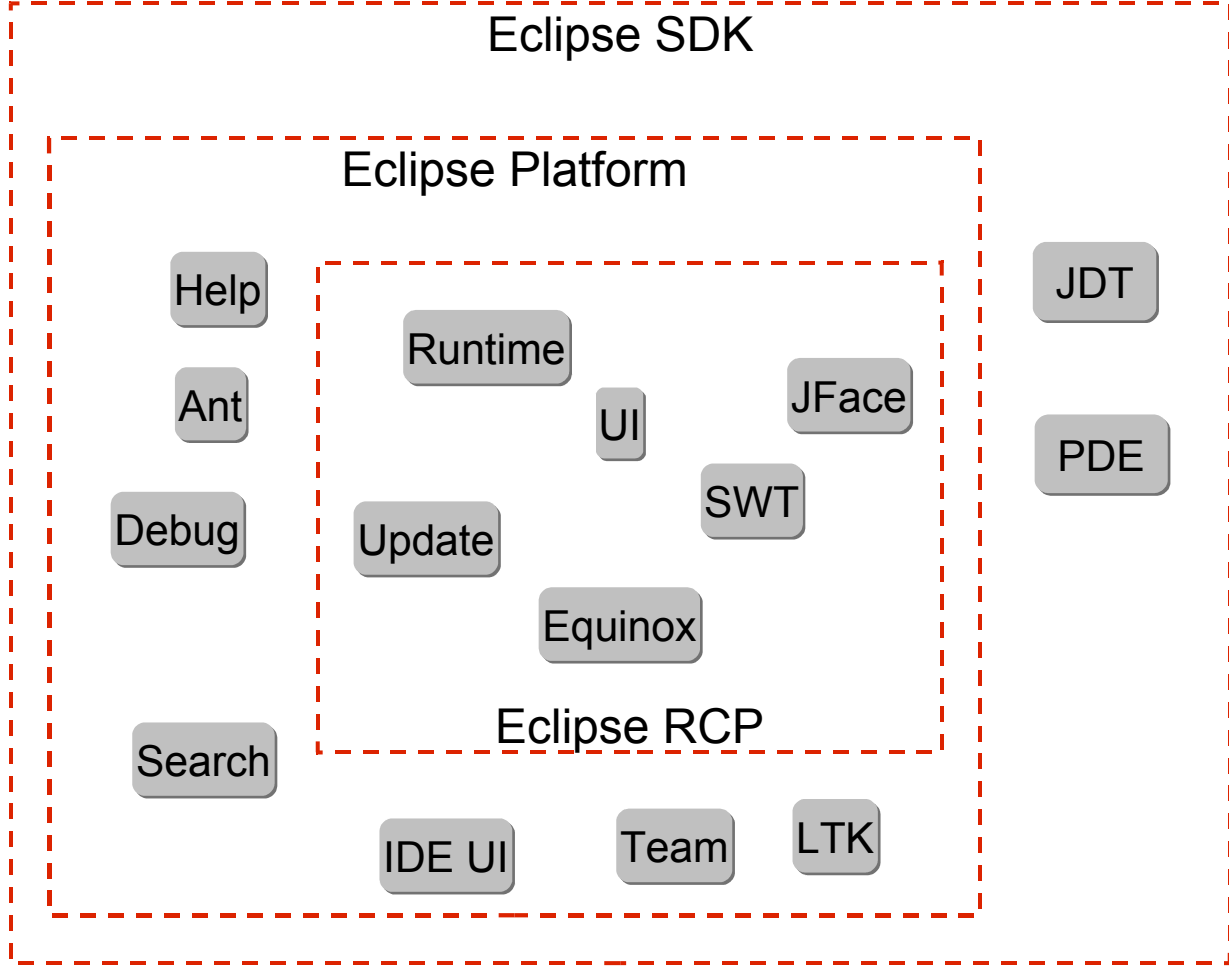
Evolution of the Client: Rich Client

- Rich user experience
- Typically (though not necessarily) a client for some backend service
- Platform independent
 - Runs with little or no modification on multiple platforms and devices
- Component model
- Integrated update mechanism
- Extensible



What Is Eclipse Rich Client Platform?

What Is the Eclipse Rich Client Platform?



RCP Example: ERP System (RPC Software)

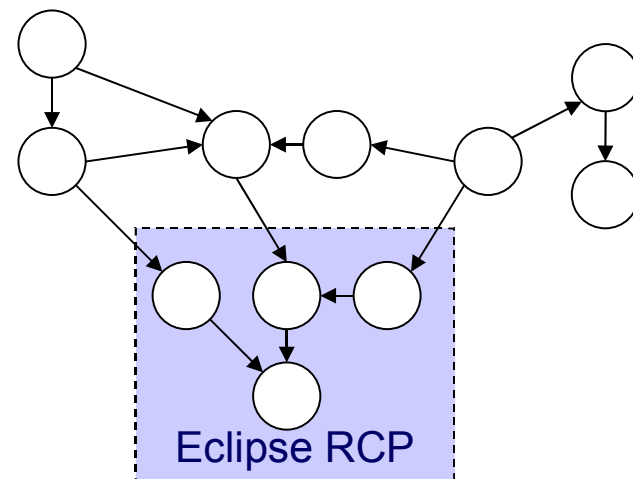
- Targets the contract furniture industry
- Embedded catalogs, accounting and EDI functionality
- Deployed at multiple customers

The screenshot displays a web-based ERP application interface. The main window is titled "Core Demo - Core - Unsaved Order - Core Desktop" and contains several panes:

- Order Header:** Shows order details for "ECLIPSE CONSORTIUM" with order number 25040355. Fields include Vendor (KNOLEG), Account, and various pricing and discount options.
- Order Line Items:** A table listing 6 line items with columns for Line, Parent, Create Date, Req Ship, Vendor, Qty, Price, Cost, Status, and Description.
- Select and configure a product:** A side panel for configuring product options, including a table for "Specify the products options" with columns for Number, Description, Finishes, and Price.
- Product Image:** A large image of a black office chair is displayed in the bottom right corner of the configuration panel.

Equinox (1/2)

- Equinox is the Eclipse component model
 - Based on OSGi R4 specification
 - Standard Java technology lacks an explicit notion of components
- Components = Bundles = Plug-in
 - Versioned
 - Defined declaratively
 - Dynamically loadable/unloadable
 - Support dynamic update and install
- Explicitly define
 - Dependencies
 - Runtime visibility
 - Interactions (extension points/extensions)



Equinox (2/2)

- Components integrate without interfering
 - Required components set explicitly
 - Unrelated components do not have direct access to one-another
- Downstream components can access upstream components through the extension mechanism
 - Downstream component registers (declaratively) an extension point
 - Dependent components register (declaratively) extensions

Extension Points...

- Are integration points for upstream plug-ins
 - Allows the defining plug-in to work with an upstream plug-in without knowing anything about it
- Are defined in the manifest (plugin.xml)
- Facilitate loosely-coupled extensions to functionality
- Are all treated equally
 - There are no **blessed** internal plug-ins
- Any plug-in can define extension points
 - A plug-in can define any number of extension points

Extension Point Examples

- **org.eclipse.ui.popupMenus**
 - Popup menus for editors and views
- **org.eclipse.ui.actionSets**
 - Menu and toolbar for the Workbench
- **org.eclipse.auction.provider**
 - Technology-specific functionality for auction application (later)
- You can define your own extension points!

Extensions...

- Are defined in the manifest (plugin.xml)
- Hook into extension points
- Provide information as required
- For example, an extension to `org.eclipse.ui.popupMenus` includes
 - The text to display in the menu
 - The type of object the menu applies to
 - An icon to display
 - The name of the class that provides behavior
 - More...

Extension Registry

- At runtime, Eclipse gathers the extension points and extension into the registry where plug-in can find them
 - Plug-in explicitly get extensions to their extension points from the extension registry
- Plug-ins are only loaded only when needed
- Example: extension to `org.eclipse.ui.popupMenus`
 - Manifest information is used to create the menu
 - When the menu is selected, the plug-in is loaded and the **behavior** class is instantiated

Eclipse RCP as an Integration Point

- Integrating completely independent components is easy
 - Views from independent components can easily share the workspace
 - Menus populated by multiple components
- Integrating loosely coupled components requires planning
 - Sharing selection information
 - Drag and drop between views from different components
 - Extension point/extension mechanism for more intimate integration
 - Not artificially restricted in any way: non-GUI extension points can be used to provide all types of extensions

RCP Example: Maestro–NASA Space Mission Management

The screenshot displays the Maestro software interface for mission management. The main window shows an orbital view of the lunar surface with a green path and 34 numbered sites. A table on the right lists mission data, and a bottom-right window shows a zoomed-in image of the lunar surface.

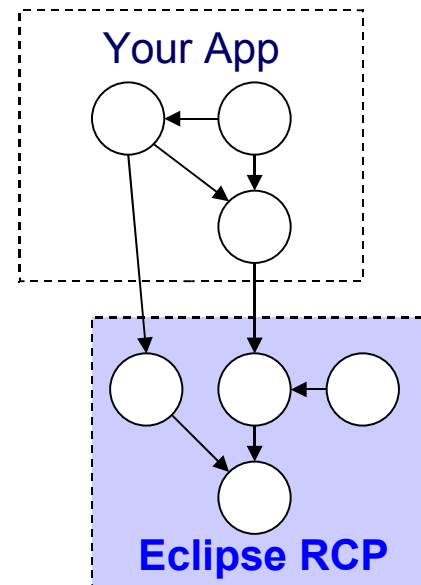
Product ID	Instrument	Sol	Seq...
2F134356767EFF2600P1212L0M1	FRONT_HAZCAM_LEFT	90	p1212
2F134449644EFF2700P1212L0M1	FRONT_HAZCAM_LEFT	91	p1212
2F134614869EFF2700P1403L0M1	FRONT_HAZCAM_LEFT	93	p1403
2F134615161EDN2700P1131L0M1	FRONT_HAZCAM_LEFT	93	p1131
2F135147950EDN2700P1131L0M1	FRONT_HAZCAM_LEFT	99	p1131
2F135148174ESF2700P1127L0M1	FRONT_HAZCAM_LEFT	99	p1127
2F135149185EDN2700P1141L0M1	FRONT_HAZCAM_LEFT	99	p1141
2F135149794EDN2700P1141L0M1	FRONT_HAZCAM_LEFT	99	p1141
2F135150380EDN2700P1141L0M1	FRONT_HAZCAM_LEFT	99	p1141
2F135150997EDN2700P1141L0M1	FRONT_HAZCAM_LEFT	99	p1141
2F135151610EDN2700P1141L0M1	FRONT_HAZCAM_LEFT	99	p1141
2F135152416EDN2700P1141L0M1	FRONT_HAZCAM_LEFT	99	p1141
2F135152602EFF2700P1212L0M1	FRONT_HAZCAM_LEFT	99	p1212
2F135153765EDN2700P1111L0M1	FRONT_HAZCAM_LEFT	99	p1111

Why Use Eclipse Rich Client Platform?

- A consistent and native look and feel across applications and features
- Provides common application services
 - Standardized component model (Equinox)
 - Pervasive extensibility–Extension registry
 - Native look and feel (SWT)
 - Window management
 - Help system
 - Update Manager
- First-class development tools
- Middleware for building rich client applications!
 - Allows programmers to focus on core application not the plumbing
 - Do not reinvent the wheel

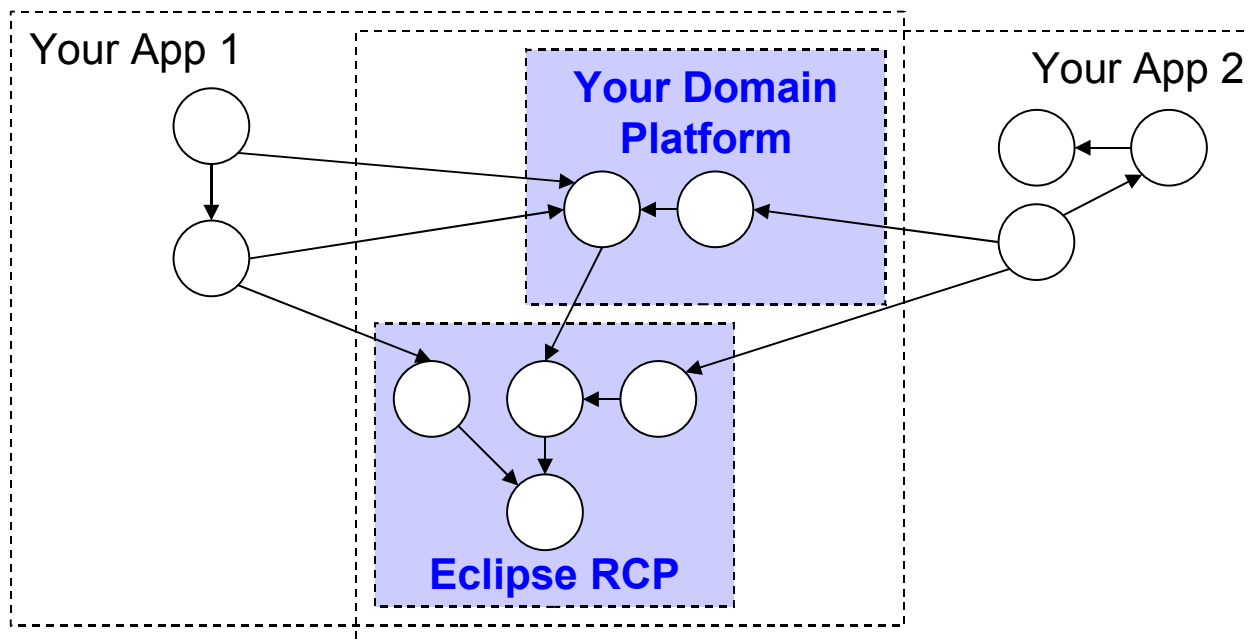
Building Platforms (1/2)

- The preceding examples all have an underlying domain-specific platform
- When starting development on RCP, it is common to provide a handful of domain-specific components that sit directly on top of RCP



Building Platforms (2/2)

- It is natural for RCP development to spawn one or more **platforms**
 - A custom base for multiple development teams to build their applications upon



Example

The screenshot shows a web browser window titled "Mocktion" displaying an auction site. The interface includes a sidebar with a tree view of categories: Toys, Models, Trains, Cars, Trucks, Die-cast, and Junk. The main content area features a table of auction items and a detailed view of a selected item.

Title	Current Price
Exc Fleischmann German N S	\$7.09
Fleischmann German N Scale	\$9.50
MINT Fleischmann German N	\$5.01

Fleischmann German N Scale DB 2nd CI Passenger Car NIB!

General information

Title: [Fleischmann German N Scale DB 2nd CI Passenger Car NIB!](#)

Primary category: Trains

Current price: \$9.50

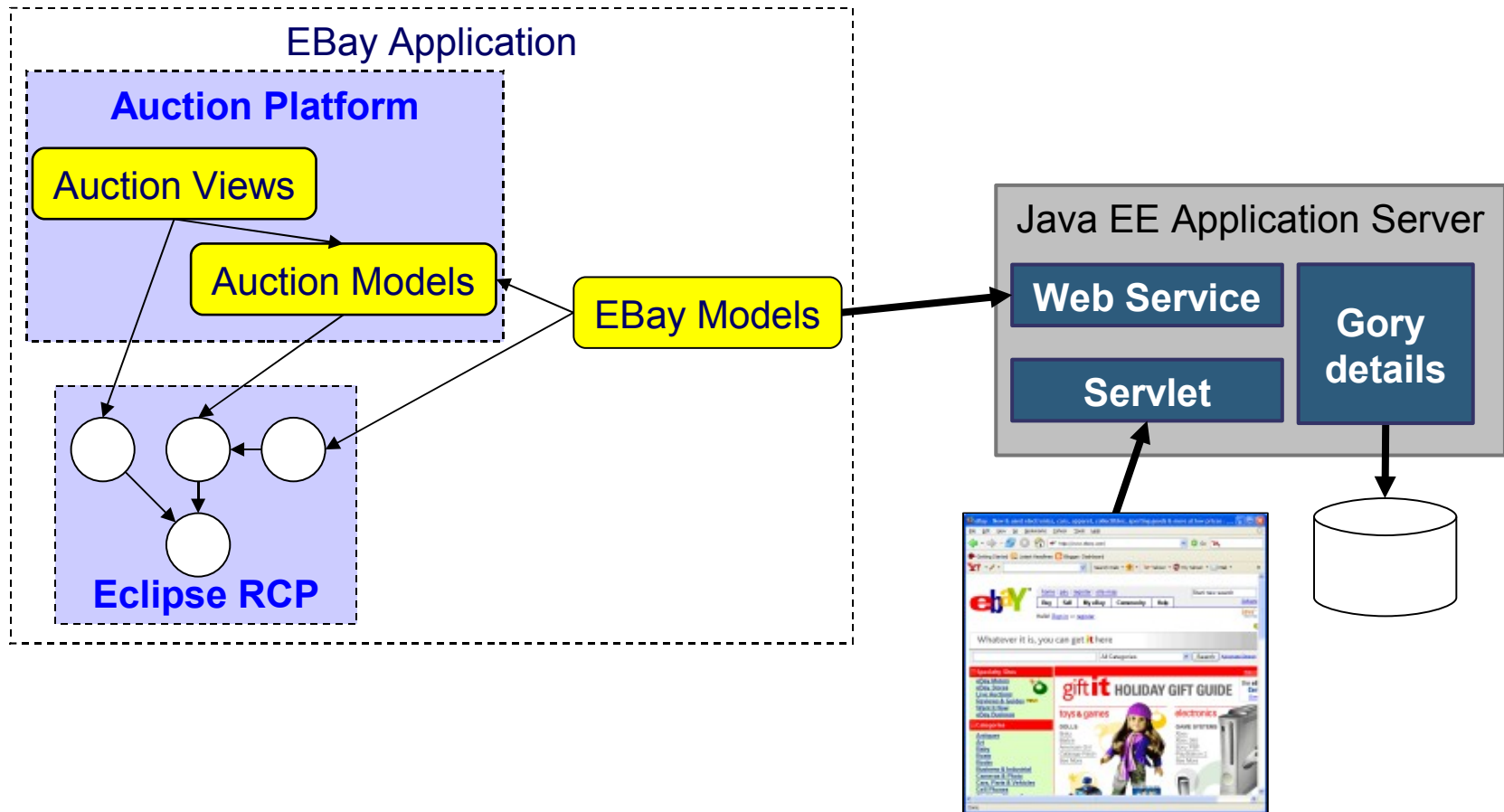
Time left:

Description

Fleishmann 2nd Class passenger car, single truck wheels for local service. Old fashioned style car, riveted steel construction probably 1920-30s vintage, with DB markings indicating 1950s era. Includes original box. Condition is NEW with no scratches or chips, no scuffs or missing paint, just perfect. Always stored in the boxes in smoke free home and with a total run time of less than an hour.

Image

Overview of the Example



Let's See Some Code...

```

class ViewLabelProvider extends LabelProvider {
    public String getText(Object object) {
        if (object instanceof Category) {
            return ((Category)object).getName();
        } else if (object instanceof IWorkbenchAdapter) {
            return ((IWorkbenchAdapter)object).getLabel(null);
        } else return object.toString();
    }

    public Image getImage(Object object) {
        if (object instanceof IWorkbenchAdapter) return null;
        if (object instanceof Category) return getCategoryImage();

        return null;
    }

    private Image getCategoryImage() {
        return PlatformUI.getWorkbench().getSharedImages().getImage(
            ISharedImages.IMG_OBJ_FOLDER);
    }
}
  
```

Tools and Frameworks

- Many tools and frameworks available to help you build your application
 - Eclipse Development Tools for the Java Platform (JDT)
 - Plug-in Development Environment (PDE)
 - Visual Editor (VE)
- ...and runtime support...
 - Business Intelligence Reporting Tools (BIRT)
 - Eclipse Modeling Framework (EMF)
 - Graphical Editing Framework (GEF)
 - Eclipse Communication Framework (ECF)

Deployment

Branding

Splash Screen

The splash screen appears when the product launches. If its location is not specified, the 'splash.bmp' file is assumed to be in the product's defining plug-in.

Specify the plug-in in which the splash screen is located:

Plug-in:

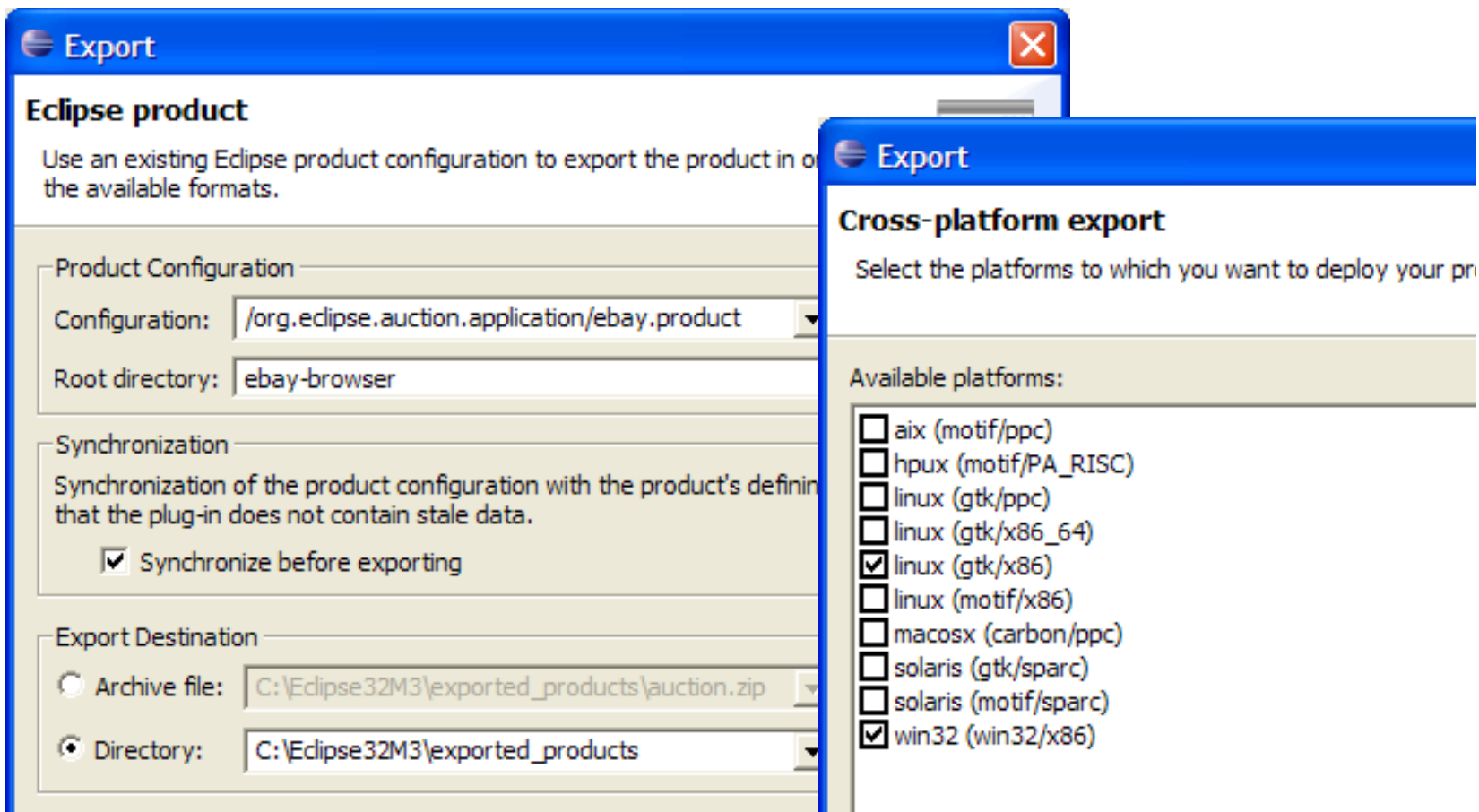
Window Images

Specify the images that will be associated with the application window. These GIF images are typically located in the product's defining plug-in.

16x16 Image:

32x32 Image:

Deployment for Multiple Platforms (1/2)



The image shows two overlapping Eclipse dialog boxes. The background dialog is titled "Export" and has the "Eclipse product" section selected. The foreground dialog is also titled "Export" and has the "Cross-platform export" section selected.

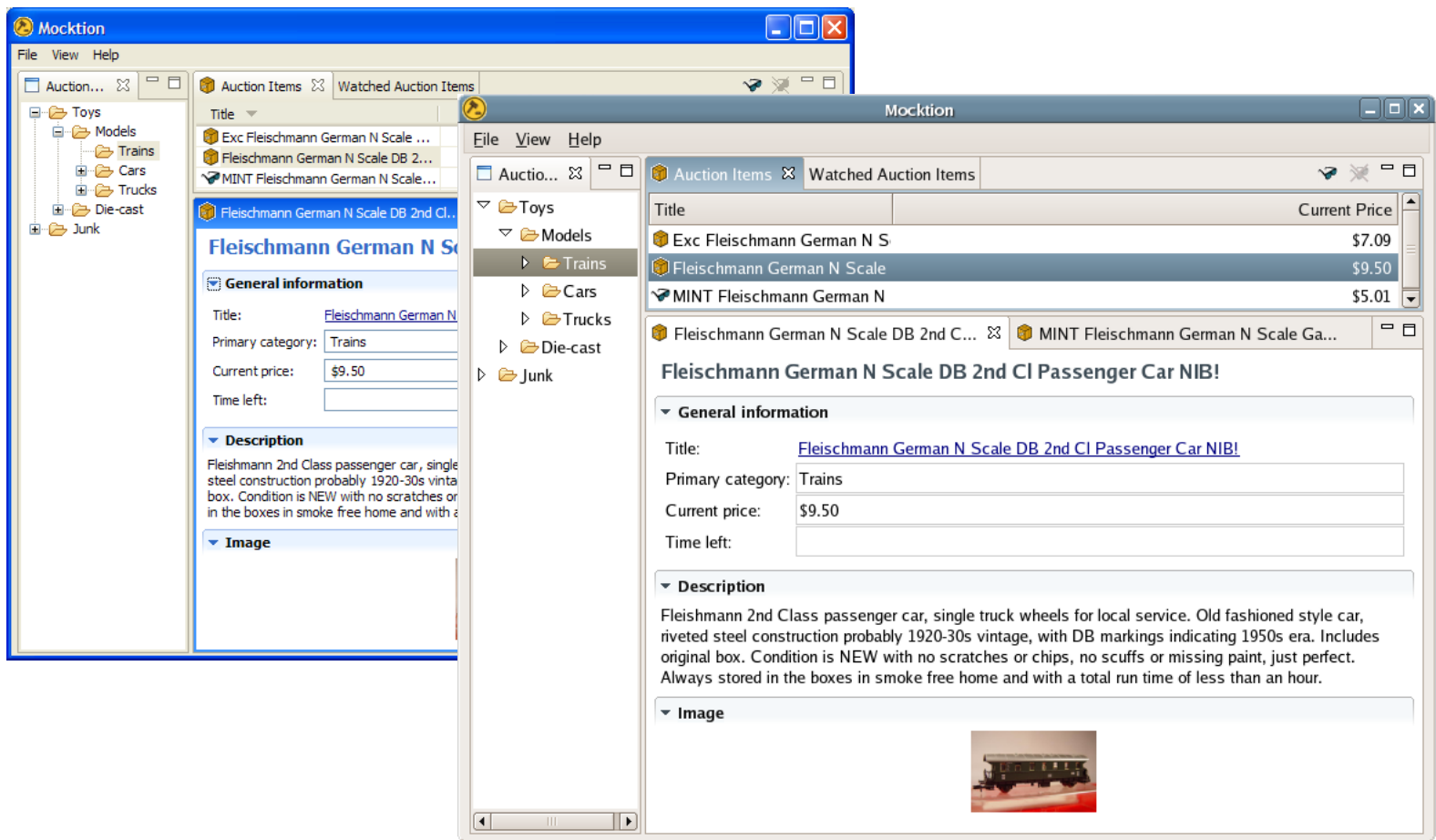
Eclipse product dialog:

- Use an existing Eclipse product configuration to export the product in one of the available formats.
- Product Configuration:
 - Configuration: /org.eclipse.auction.application/ebay.product
 - Root directory: ebay-browser
- Synchronization:
 - Synchronization of the product configuration with the product's definition that the plug-in does not contain stale data.
 - Synchronize before exporting
- Export Destination:
 - Archive file: C:\Eclipse32M3\exported_products\auction.zip
 - Directory: C:\Eclipse32M3\exported_products

Cross-platform export dialog:

- Select the platforms to which you want to deploy your product.
- Available platforms:
 - aix (motif/ppc)
 - hpux (motif/PA_RISC)
 - linux (gtk/ppc)
 - linux (gtk/x86_64)
 - linux (gtk/x86)
 - linux (motif/x86)
 - macosx (carbon/ppc)
 - solaris (gtk/sparc)
 - solaris (motif/sparc)
 - win32 (win32/x86)

Deployment for Multiple Platforms (2/2)





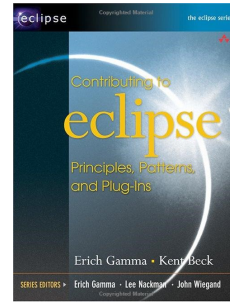
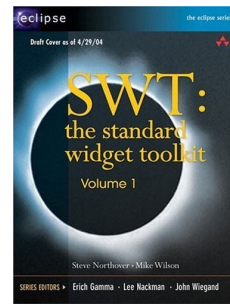
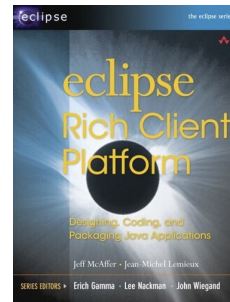
Wrap Up

More Information

- Project web site
 - <http://eclipse.org/rcp>
 - http://wiki.eclipse.org/index.php/Rich_Client_Platform
- Examples and Case Studies
 - <http://www.eclipse.org/community/rcp.php>
- RCP for PHB
 - http://eclipsercp.org/book/chapters/RCP_Foreward2.pdf
- Usenet newsgroup
 - <news://news.eclipse.org/eclipse.platform.rcp>
- <http://eclipseplugincentral.com/>
- <http://osgi.org>

Recommended Reading

- Eclipse Rich Client Platform
 - By Jeff McAffer and Jean-Michel Lemieux
 - Addison-Wesley Professional
 - ISBN: 0321334612
- SWT: The Standard Widget Toolkit, Volume 1
 - By Steve Northover, Mike Wilson
 - Addison-Wesley Professional
 - ISBN: 0321256638
- Contributing to Eclipse: Principles, Patterns, and Plugins
 - By Erich Gamma, Kent Beck
 - Addison-Wesley Professional
 - ISBN: 0321205758



Q&A





the
POWER
of
JAVA™



JavaOne
Part of the Network for Business Success

Building Applications for Multiple Platforms Using the Eclipse Rich Client Platform

Wayne Beaton
Evangelist

<http://eclipse.org/evangelism>

<http://wbeaton.blogspot.com>

The Eclipse Foundation

TS-3422