Eclipse Rich Client Platform

Wayne Beaton, Evangelist
http://eclipse.org/evangelism
The Eclipse Foundation

July 4, 2006
Agenda

- What is Eclipse?
- What is a Rich Client?
- What is Eclipse Rich Client Platform?
- 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
  - 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, 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’re 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 Project

- Ten (10) top level projects
  - The Eclipse Project
  - Tools
  - Web Tools Platform
  - Test & Performance Tools Platform
  - Business Intelligence and Reporting Tools
  - Eclipse Modeling Project
  - 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
- 137 members, including major Java, Linux and Embedded vendors
  - BEA, Borland, Computer Associates, IBM, Nokia, Intel, Wind River, Zend, …
- Dozens of 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
“Callisto” Release Train

- Ten (10) projects released simultaneously on June 30, 2006
  - Business Intelligence and Reporting Tools (BIRT) Project
  - C/C++ IDE (CDT)
  - Data Tools Platform (DTP)
  - Eclipse Modeling Framework (EMF)
  - Graphical Editing Framework (GEF)
  - Graphical Modeling Framework (GMF)
  - Eclipse Project
  - Eclipse Test and Performance Tools Platform Project (TPTP)
  - Eclipse Web Tools Platform Project (WTP)
  - Visual Editor (VE)

- Improve the cohesion across projects and developer teams
- Unified update site
- Not a “one step” installer

“Oh release train sounding louder
Glide on the release train
Come on now release train
Yes, release train holy roller”
What is a Rich Client?
What is a Rich Client?

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

http://udig.refractions.net/confluence/display/UDIG/Home
Example: Plum Canary - Chirp

Example: iMedic

Example: RSS Solutions

Evolution of the Client: Fat Client

- All application logic on the client
- Rich user experience
- Platform dependent
- Monolithic application
- Relatively difficult to update
Evolution of the Client: Thin Client

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

- Thin client
  - Casual users don’t want to install software
  - Online banking, auction browsing

- Rich client
  - Power users demand responsive, rich, powerful user interfaces
  - Teller, bank manager, power auctioneer

I like all my clients to be rich!
What is Eclipse Rich Client Platform?
Eclipse Rich Client Platform? (1/3)
Eclipse Rich Client Platform? (2/3)
Eclipse Rich Client Platform? (3/3)
Example: ForeFlight

- Displays critical information graphically and prominently
- Displays alerts when conditions are near or exceeding the user's preferred limits
- Connects via the web to weather and information services
- Multiple ergonomic views of the weather that affects the go/no-go flight decision

http://www.foreflight.com/
Why Use Eclipse Rich Client Platform?

- A consistent and native look and feel across applications and features
- Provides common application services
  - Native look and feel
  - Window management
  - Standardized component model (Equinox)
    - Pervasive extensibility – Extension registry
    - Update Manager
  - Help system
- First-class development tools
- Middleware for building rich client applications!
  - Allows programmers to focus on core application not the plumbing
  - Don’t reinvent the wheel
Building Platforms (1/2)

- It all starts with plug-ins
  - Eclipse RCP applications are composed of components that plug into the 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’s natural for RCP development to spawn one or more “platforms”
  - A custom base for multiple development teams to build their applications upon
  - All of the examples in this presentation all have an underlying domain-specific platform
Example: Maestro – NASA Space Mission Management

**Equinox (1/2)**

- Equinox is the Eclipse component model
  - Based on OSGi R4 specification
  - Standard Java 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
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
Example: eBay Auction Browser
Overview of the Example

EBay Application

Auction Platform
- Auction Views
- Auction Models
- Eclipse RCP

Java EE Application Server
- Web Service
- Servlet
- Gory details

EBay Models
Another Example

**Organizer Platform**

- **Models**
- **Views**

- **Outlook DnD**
- **Thunderbird DnD**
- **Text DnD**
- **Workspace backend**
- **Derby backend**

**Eclipse RCP**

**Table:**

<table>
<thead>
<tr>
<th>Due Date</th>
<th>Subject</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>10-Apr-2006</td>
<td>Review Open Healthcare Framework</td>
<td>No</td>
</tr>
<tr>
<td>10-Apr-2006</td>
<td>Modify hotel booking for 2006 JavaOne</td>
<td>No</td>
</tr>
<tr>
<td>31-Mar-2006</td>
<td>Get Eclipse Installed stickers for LinuxWorld</td>
<td>Yes</td>
</tr>
</tbody>
</table>
**RCP Architecture (1/2)**

- Classic best-practices apply
  - Layered architecture
  - Organizer layers into components/plug-ins
- Model-view-controller
  - Model: objects representing data or activities
  - View: visualization of the state of the model
  - Controller: mediates between the view and model
RCP Architecture (2/2)

- Explicit dependencies between components
- Explicit visibility to implementation

```
org.apache.derby

org.eclipse.organizer.task.ui

org.eclipse.organizer.core

org.eclipse.ui

org.eclipse.core.runtime

org.eclipse.organizer.ide

org.eclipse.emf.ecore.xmi
```
Wrap up
Tools and Frameworks

- Many tools and frameworks available to help you build your application
  - Java Development Tools (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)

  … and more!
More information…

- Project web site
  - http://eclipse.org/rcp
- Usenet newsgroup
  - news://news.eclipse.org/eclipse.platform.rcp
- http://eclipseplugincentral.com/
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