



# Eclipse for Smalltalkers

Eric Clayberg  
Sr. Vice President of Product Development  
Instantiations, Inc.  
April 22, 2002

[clayberg@instantiations.com](mailto:clayberg@instantiations.com)  
<http://www.instantiations.com>  
978-750-3621

# Who Am I?



- First used Smalltalk in late '80s; full-time since 1991; First used Java in 1996; Eclipse in 2000
- Co-Founder of (the original) ObjectShare in 1992; V.P. of Product Development for ParcPlace-Digitaltalk 1996-97 V.P. of Product Development for Instantiations since 1997
- Developer & Chief Architect of WindowBuilder Pro and over a dozen other commercial Smalltalk add-on products (VA Assist Pro, WidgetKits, etc.)

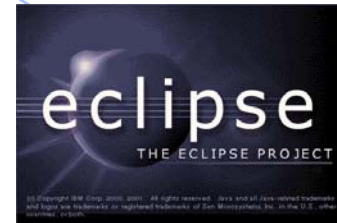
## Instantiations

- Multi-faceted Software Company - founded in 1997 (out of the ashes of ParcPlace, Digitaltalk and ObjectShare)
- Advanced Tier IBM Business Partner; Winner of the 2001 IBM Solutions Excellence Award for “Cool Tool” (for a product built with Smalltalk)
- VisualAge Smalltalk product line: VA Assist Pro, WindowBuilder Pro, WidgetKits, VSE to VAST Translation Tool, GF/ST; Also known for: JOVE, VA Assist/J, jFactor, jKits, CodePro Studio for WebSphere & JBuilder
- Built some of the very first Eclipse demos used by IBM



# Agenda

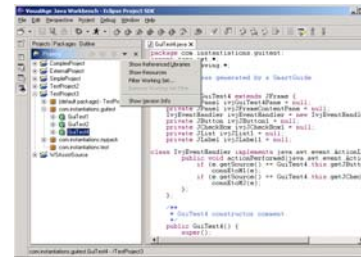
- What is Eclipse?



- Eclipse vs. Smalltalk



- Eclipse Demo



# What is Eclipse?

“An IDE for anything, and for nothing in particular”

*Eclipse Platform Technical Overview, OTI 2001*



**The Eclipse Project** is an open source software development project dedicated to providing a robust, full-featured, commercial-quality, industry platform for the development of highly integrated tools.

**The Eclipse Platform** is an open extensible IDE. “Eclipse” provides building blocks and a foundation for constructing and running integrated software development tools. It is composed of three primary sub-components: Platform, Java Development Tools (JDT), and Plug-in Development Environment (PDE).

# What is WebSphere Studio Workbench?



**IBM WebSphere Studio Workbench** is based on the Eclipse platform and is fully supported by IBM. This is the basis for all of IBM's Eclipse-based commercial tools



**IBM WebSphere Studio Application Developer** is a J2EE application development tool optimized for the WebSphere software platform. Application Developer represents the next generation of IBM offerings and is the successor to VisualAge for Java Enterprise.

# Eclipse Overview

- Created by OTI & IBM – the same folks who brought you **ENVY** and **VisualAge Smalltalk**
- The technology is organized around a **workbench** into which any number of **tools** can be installed.
- The workbench provides the **run-time** context in which these tools run.
- Tools generally operate on the tree of **resources** managed by the workbench.
- Resources are regular files and folders, arranged into larger units called **projects**.
- The **desktop** is built on top of the platform to provide a general-purpose graphical user interface (GUI) to the **workbench** and its resources.
- **Tools** integrate seamlessly into the platform by contributing buttons, menus, viewers, and other GUI elements to the desktop.

# Key Components

- Platform Services
  - Resource Management
  - Team Programming Model
  - Interoperability with non-WebSphere Tools
  - Debugging
  - Extensibility Mechanism
- Common Frameworks
  - Standard Widget Toolkit (SWT)
  - User Interface Framework (JFace)
  - Source Editing Framework (SEF)
- Platform Extensions
  - Desktop
  - Java Tools
  - Web Tools

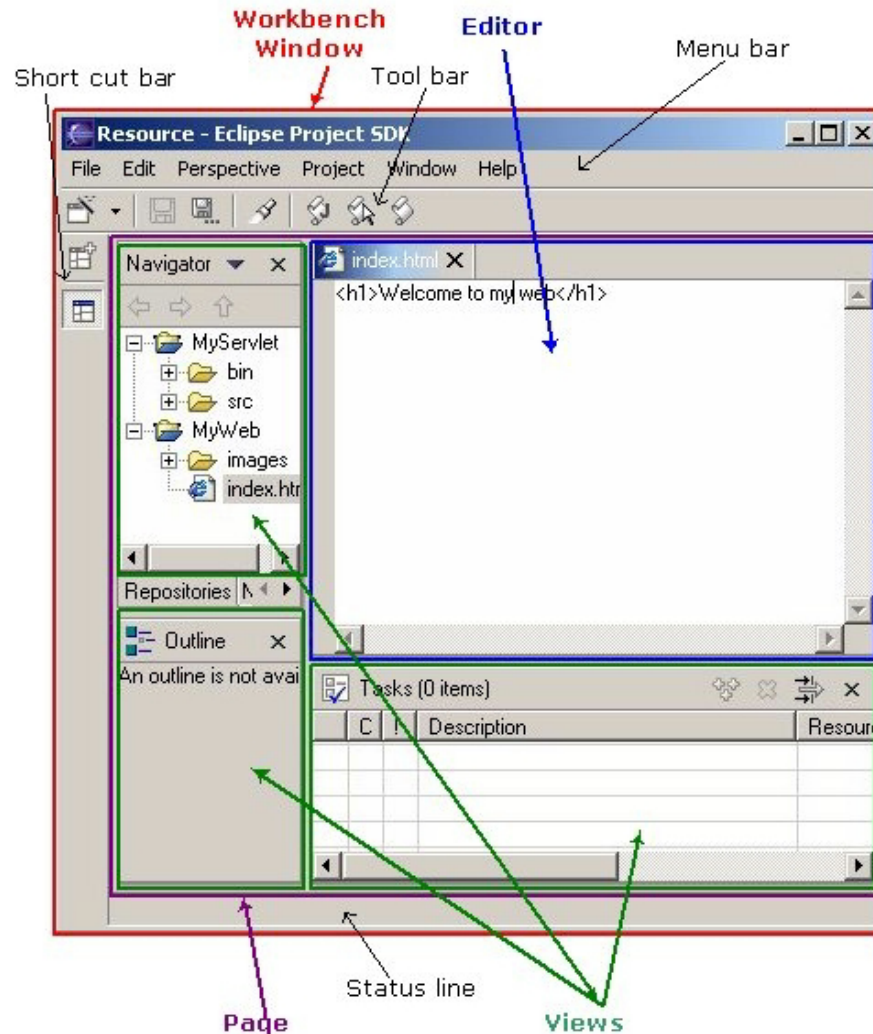
# Eclipse vs. Smalltalk

- User's Perspective
  - General Characteristics
  - Multi-Pane, MDI Interface
  - Standard IDE Features
  - Semantic Searching
  - Refactoring
- Developer's Perspective
  - General Characteristics
  - Extension API
  - Workbench Extension Points
  - Other Extension Points





# User's Perspective

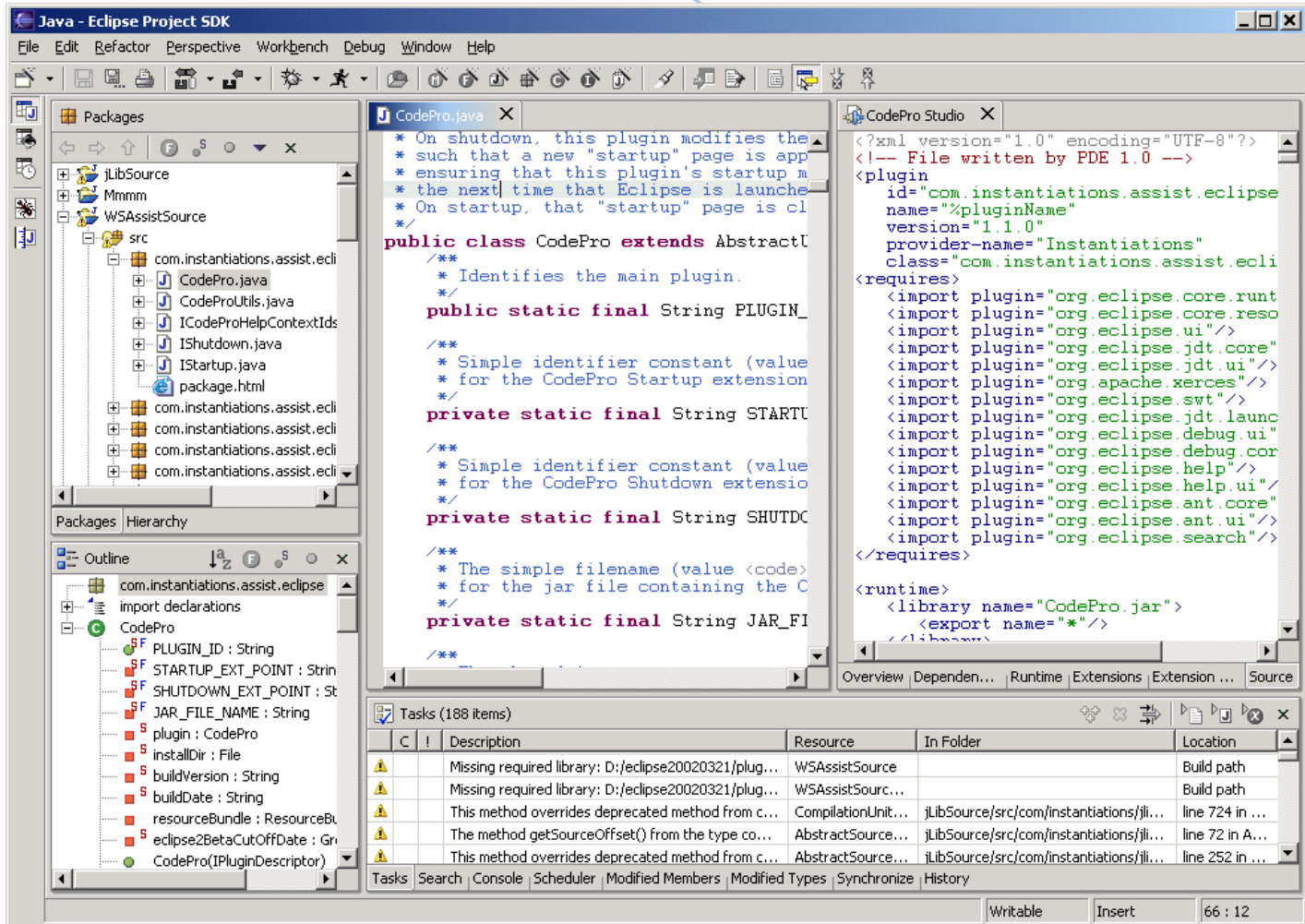


# General Characteristics

- Eclipse is a great IDE...for Java development  
(or for any file-based language)

	<b>Eclipse</b>	<b>Smalltalk IDE</b>
<b>Languages</b>	Java, HTML, XML, C++, C#, etc.	<b>Smalltalk</b>
<b>Repository</b>	CVS, PVCS, ClearCase, etc.	<b>ENVY, StORE, etc.</b>
<b>Platforms</b>	Windows, Linux, Mac, Solaris, QNX, AIX	<b>Everything</b>
<b>Navigation</b>	File & Directory based, some OO views	<b>Classes &amp; Methods, very OO</b>
<b>UI Style</b>	Single window, MDI-like	Multi-window
<b>Change Tracking</b>	Configurable “local history”	Every change saved forever (ENVY)
<b>Compilation</b>	Incremental or batch, limited hot code replacement with JDK 1.4	<b>Incremental with hot code replacement</b>
<b>Debugger</b>	OK	<b>Excellent</b>
<b>Customization</b>	Limited to exposed preferences and plugin points	<b>Virtually unlimited for a moderately skilled developer</b>
<b>Deployment</b>	<b>Easy, well defined</b>	Problematic (IDE dependent)
<b>Testing</b>	JUnit	SUnit

# Multi-Pane, MDI Interface



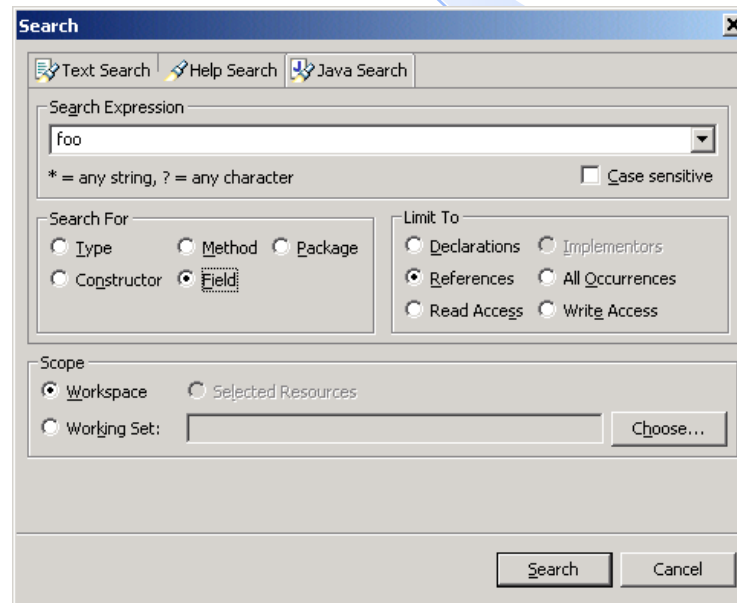
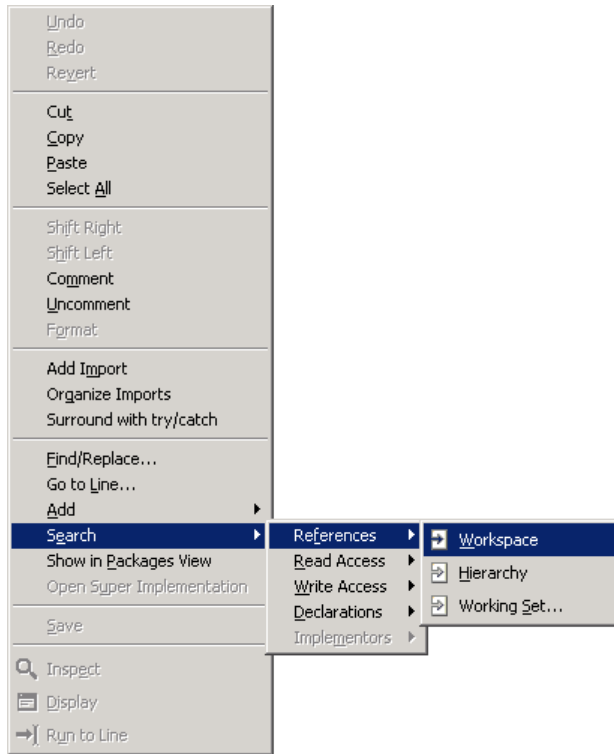
# Standard IDE Features

- Eclipse includes most “modern” IDE features like drag/drop, syntax coloring, code assist, etc. Does not include a GUI builder.
- Eclipse also includes most “standard” Smalltalk IDE features like true semantic searching (e.g., senders & implementors with various scoping options)

	<b>Eclipse</b>	<b>Smalltalk IDE</b>
<b>Drag &amp; Drop</b>	Yes	Yes (IDE dependent)
<b>Syntax Coloring</b>	Yes	Yes (IDE dependent)
<b>Formatter</b>	Yes	Yes
<b>Code Assist</b>	<b>Yes</b>	Limited
<b>Macros</b>	Yes	Yes
<b>Workspace/Scrapbook</b>	Limited, scoping problems	<b>Just “do it” anywhere</b>
<b>GUI Builder</b>	Not yet	<b>Great choices</b>
<b>Searching</b>	Very powerful, semantic-based, limited to exposed capabilities	Very powerful, semantic-based, easily scripted by user

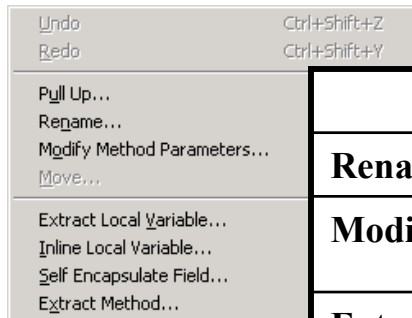
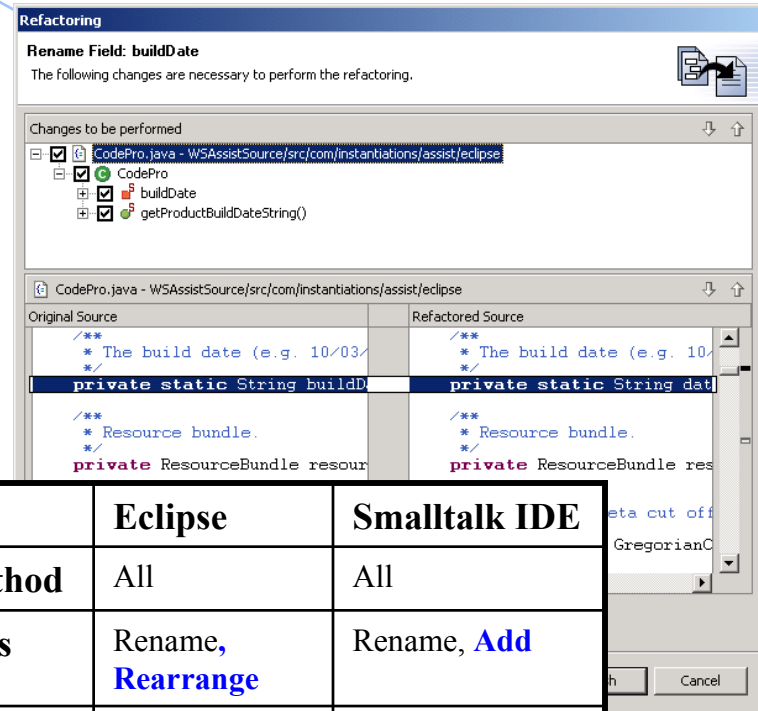
# Semantic Searching

- Declaration and Reference searches for types, methods and fields
- Additional Read and Write Access searches for fields



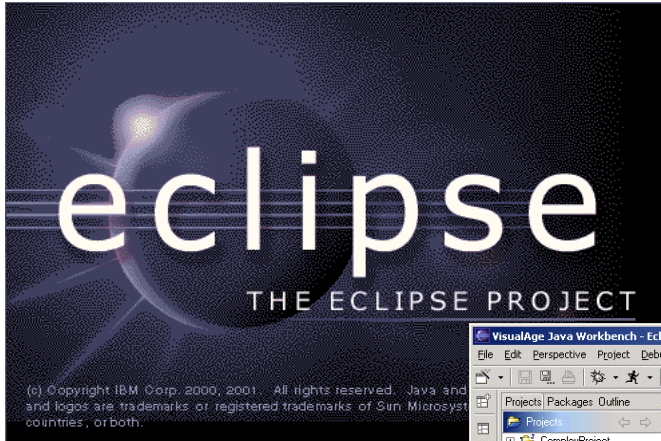
# Refactoring

- Eclipse includes extensive built-in refactoring capabilities with a comprehensive API
- The Refactoring Browser is available for most Smalltalk dialects and includes a very nice scriptable replacement engine



	Eclipse	Smalltalk IDE
<b>Rename Class/InstVar/Method</b>	All	All
<b>Modify Method Parameters</b>	Rename, <b>Rearrange</b>	Rename, <b>Add</b>
<b>Extract/Inline Method</b>	Extract Only	<b>Both</b>
<b>Inline/Extract Local Variable</b>	Both	Both
<b>Pull Up/Push Down InstVar/Method</b>	Pull Up Only	<b>Both</b>
<b>Encapsulate/Inline InstVar</b>	Encapsulate Only	<b>Both</b>

# Demo – User’s Perspective



The screenshot shows the Eclipse IDE interface. On the left, the 'Projects' and 'Packages Outline' views are visible. The 'Packages Outline' shows a project structure with a package named 'com.instantiations.guitest' containing classes 'GuiTest1', 'GuiTest2', and 'GuiTest4'. The 'GuiTest4' class is selected. The main editor window displays the source code for 'GuiTest4.java'. A comment above the class definition reads: 'This class was generated by a SmartGuide.' The code defines a class 'GuiTest4' that extends 'JFrame' and implements 'IvjEventHandler'. It includes private fields for GUI components and a constructor that calls 'super()'.

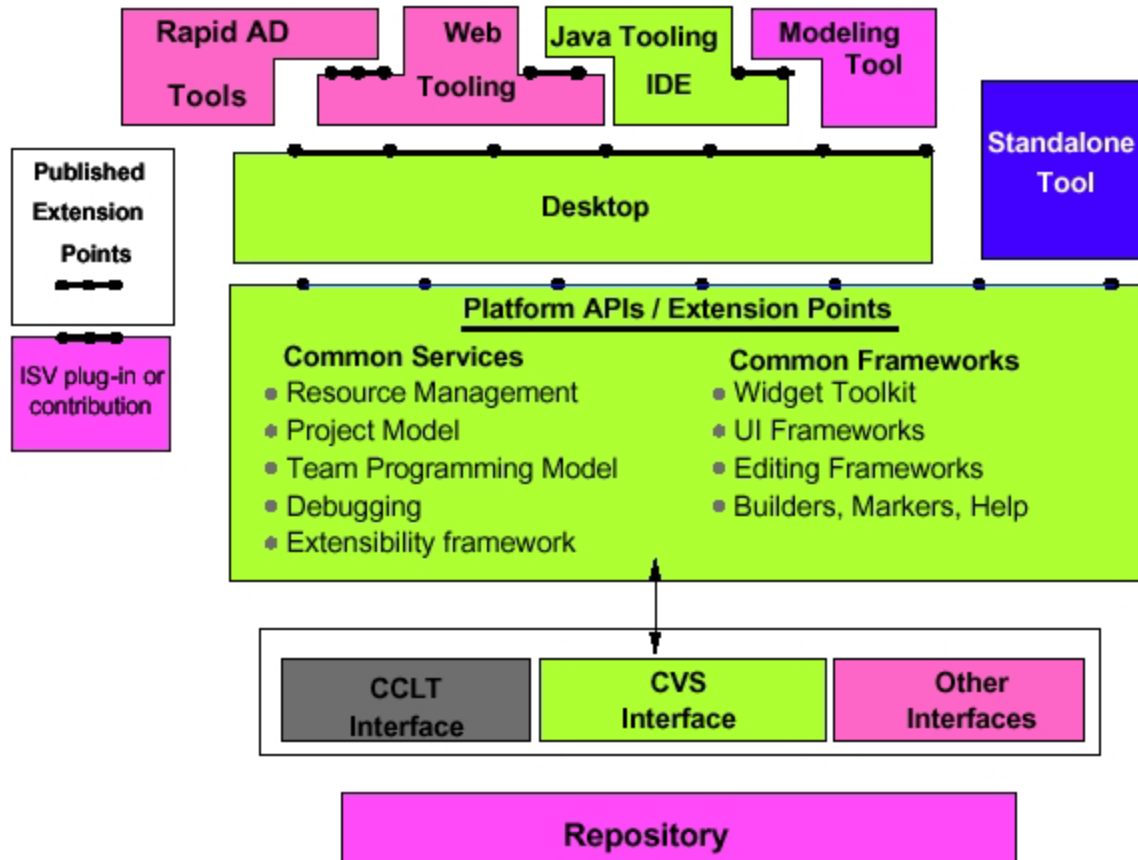
```
package com.instantiations.guitest;
import java.awt.*;
import javax.swing.*;

/**
 * This class was generated by a SmartGuide.
 */
GuiTest4 extends JFrame {
    private JPanel ivjGuiTest4Pane = null;
    private IvjEventHandler ivjEventHandler = new IvjEventHandler();
    private JButton ivjJButton1 = null;
    private JCheckBox ivjJCheckBox1 = null;
    private JList ivjJList1 = null;
    private JLabel ivjJLabel1 = null;

    class IvjEventHandler implements java.awt.event.ActionListener {
        public void actionPerformed(java.awt.event.ActionEvent e) {
            if (e.getSource() == GuiTest4.this.getJButton1()) {
                connEtoM1(e);
            }
            if (e.getSource() == GuiTest4.this.getJCheckBox1()) {
                connEtoM2(e);
            }
        }
    };

    /**
     * GuiTest4 constructor comment.
     */
    public GuiTest4() {
        super();
    }
}
```

# Developer's Perspective





# General Characteristics

- Eclipse is a well-designed, well-factored, OO environment
- Most of its weaknesses are due to Java language restrictions (had it been implemented in Smalltalk, it would have been amazing!)
- Based on the concept of layered, dependent plugins

	<b>Eclipse</b>	<b>Smalltalk IDE</b>
<b>Structure</b>	File-Based	Image-Based
<b>Source Access</b>	Full Source	Full Source
<b>Modularity</b>	<b>Highly Modular, dynamically built based on available plugins</b>	Tends to be monolithic
<b>Plugin Coupling</b>	<b>Very little, good isolation provided, no chance of collision</b>	Highly coupled, easy for one plugin to kill entire IDE, easy for tools to collide
<b>Visibility</b>	Controlled by plugins and special class loaders	<b>All classes are visible</b>
<b>Reflection</b>	Weak	<b>Highly Capable</b>
<b>Experimentation</b>	Hard, must build complete plugin	<b>Just “do it” anywhere</b>

# Extension API

- Eclipse provides a full-featured API
- Plugins are written in Java and integrated via XML
- Plugins aren't activated until actually needed

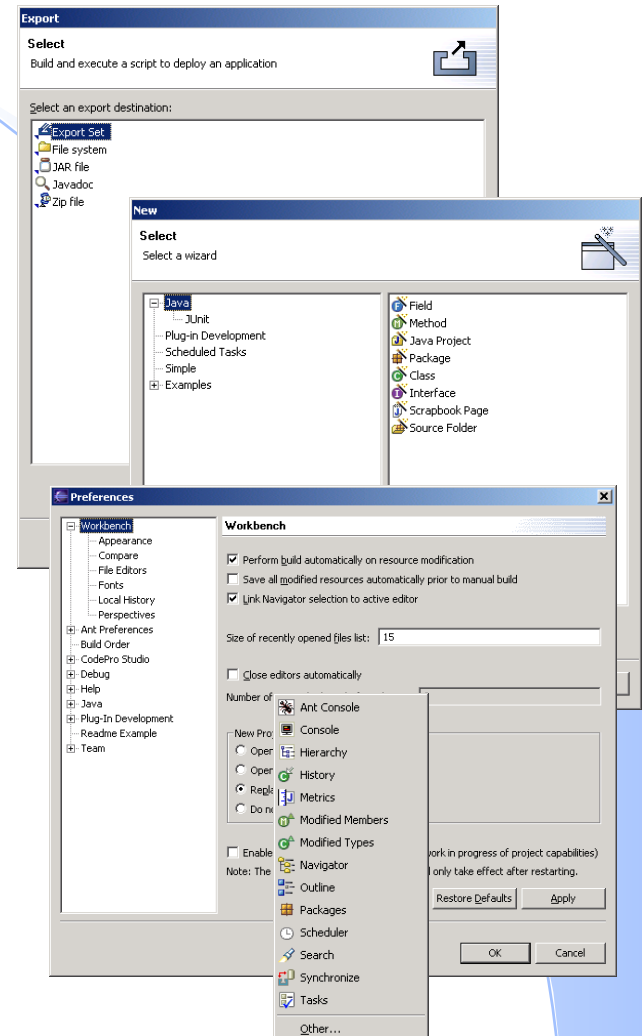
```
<objectContribution
  objectClass="org.eclipse.core.resources.IFile"
  nameFilter="*.java" id="com.foo.openMyJavaEditor">
  <action label="Edit File" icon="images/my_editor.gif"
    id="com.foo.openMyEditorAction"
    class="com.foo.OpenMyEditorActionDelegate"
    enablesFor="1" menuBarPath="additions"
    helpContextId="my_editor_action_context">
  </action>
</objectContribution>
```

	Eclipse	Smalltalk IDE
<b>Extension Language</b>	Java, XML	Smalltalk
<b>Extensibility</b>	Rigid, Formal, Well defined, Limited	Easy, Ad-hoc, Unlimited
<b>API Access</b>	Limited access (much is marked private or internal)	<b>Unlimited access, fully public</b>
<b>UI Library</b>	Native, SWT, JFace, ActiveX, no LCD	Native or Emulated (Dialect dependent), no LCD
<b>Plugin Activation</b>	Lazy (no startup hook)	<b>Startup hook available</b>

# Workbench Extension Points

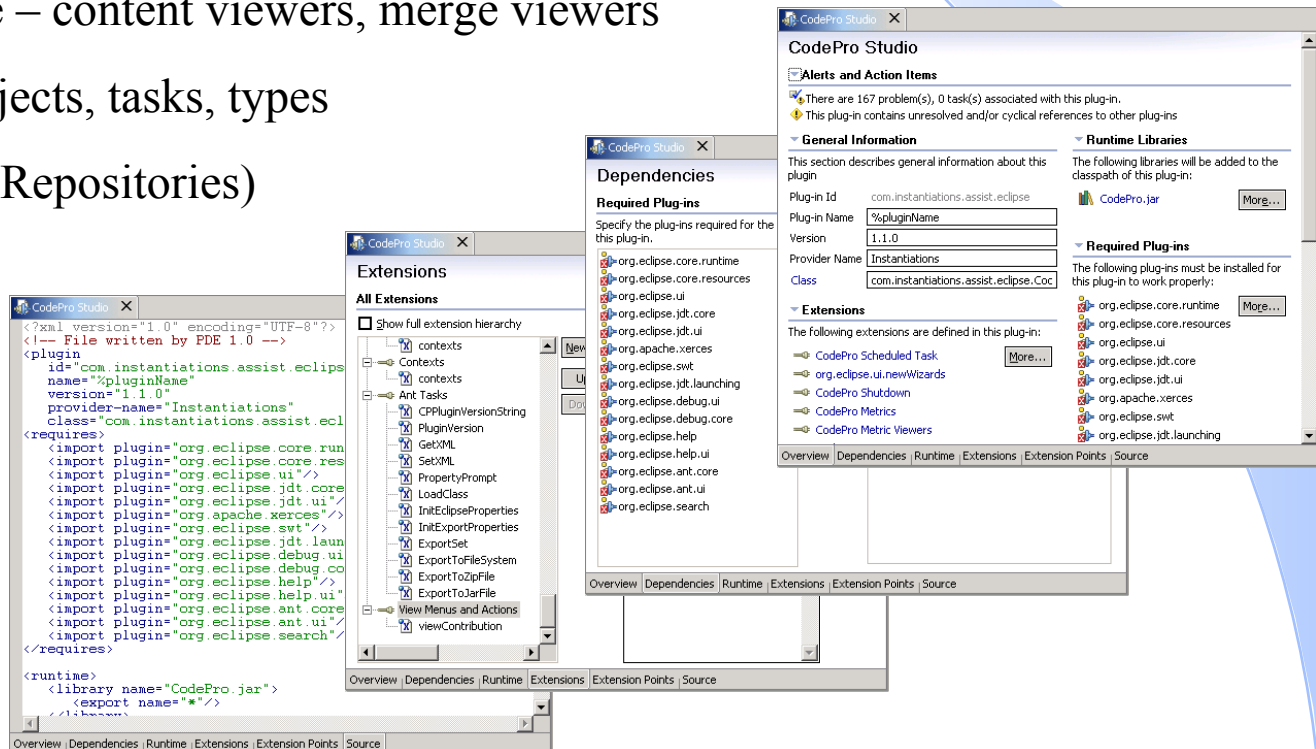
- Workbench

- **Action Sets** - used to add menus, menu items and toolbar buttons to the common areas in the workbench window
- **Editors** - used to add new editors to the workbench
- **Editor Actions** - used to add actions to the menu and toolbar for editors
- **Export Wizards** - used to register export wizard extensions
- **Import Wizards** - used to register import wizard extensions
- **New Wizards** - used to register resource creation wizard extensions
- **Perspectives** - used to add perspective factories to the workbench
- **Perspective Extensions** - used to extend perspectives
- **Popup Menus** - used to add new actions to pop-up menus
- **Preference Pages** - used to add pages to the preference dialog box
- **Property Pages** - used to add additional property page for objects of a given type
- **Views** – used to define additional views for the workbench
- **View Actions** - used to add actions to the menu and toolbar for views

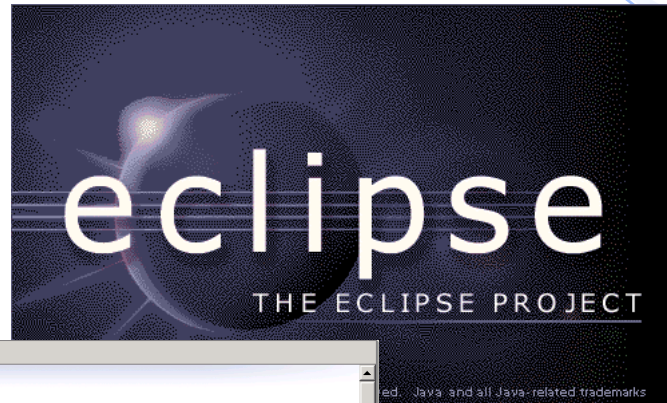


# Other Extension Points

- Workspace
  - Builders - The job of a builder is to process a set of resource changes
  - Markers - A marker is a kind of metadata (similar to properties) which can be used to tag resources with user information
  - Natures - A nature associates lifecycle behaviour with a project
- Compare – content viewers, merge viewers
- Ant – objects, tasks, types
- Targets (Repositories)
- Help



# Demo – Developer’s Perspective



The screenshot displays the Eclipse IDE interface. On the left, the "CodePro Studio" plugin configuration window is open, showing the following details:

- Alerts and Action Items:** There are 167 problem(s), 0 task(s) associated with this plug-in. This plug-in contains unresolved and/or cyclical references to other plug-ins.
- General Information:** Plug-in Id: com.instantiations.assist.eclipse; Plug-in Name: %pluginName; Version: 1.1.0; Provider Name: Instantiations; Class: com.instantiations.assist.eclipse.Coc
- Runtime Libraries:** The following libraries will be added to the classpath of this plug-in: CodePro.jar
- Required Plug-ins:** The following plug-ins must be installed for this plug-in to work properly: org.eclipse.core.runtime, org.eclipse.core.resources, org.eclipse.ui, org.eclipse.jdt.core, org.eclipse.jdt.ui, org.apache.xerces, org.eclipse.swt, org.eclipse.jdt.launching
- Extensions:** The following extensions are defined in this plug-in: CodePro Scheduled Task, org.eclipse.ui.newWizards, CodePro Shutdown, CodePro Metrics, CodePro Metric Viewers

On the right, the "Manifest File" editor shows the plugin's manifest file:

```
red. Java and all Java-related trademarks

codePro Studio X
xml version="1.0" encoding="UTF-8"?
-- File written by PDE 1.0 -->
plugin
id="com.instantiations.assist.eclipse"
name="%pluginName"
version="1.1.0"
provider-name="Instantiations"
class="com.instantiations.assist.eclipse.CodePro">
requires
<import plugin="org.eclipse.core.runtime"/>
<import plugin="org.eclipse.core.resources"/>
<import plugin="org.eclipse.ui"/>
<import plugin="org.eclipse.jdt.core"/>
<import plugin="org.eclipse.jdt.ui"/>
<import plugin="org.apache.xerces"/>
<import plugin="org.eclipse.swt"/>
<import plugin="org.eclipse.jdt.launching"/>
<import plugin="org.eclipse.debug.ui"/>
<import plugin="org.eclipse.debug.core"/>
<import plugin="org.eclipse.help"/>
<import plugin="org.eclipse.help.ui"/>
<import plugin="org.eclipse.ant.core"/>
<import plugin="org.eclipse.ant.ui"/>
<import plugin="org.eclipse.search"/>
requires>
runtime
<library name="CodePro.jar">
<export name="*/>
</library>
</plugin>
```

# Conclusion

- Smalltalk is still the **best platform** for building tools (especially IDEs)
- Smalltalk is still the **better language** for building highly reflexive and extensible tools
- **Java**, the language, is probably Eclipse's **weakest link**
- The Java IDE world is **catching up fast** and is better in many regards
- Eclipse is a solid, **well designed** system representing the **state-of-the-art** in Java IDE design
- Eclipse is “**good enough**” to support the creation of a wide variety of IDE tools and facilities
- High quality semantic searching and refactoring capabilities are **no longer unique** to the Smalltalk world
- Smalltalk vendors should not rest on their laurels and assume that Smalltalk IDEs will always be better...

# Eclipse Resources

- Eclipse & WebSphere Studio
  - Eclipse.org Home Page  
<http://www.eclipsecorner.org/>
  - Downloads  
<http://www.eclipsecorner.org/downloads/>
  - Technical Articles  
<http://www.eclipsecorner.org/articles/>
  - Mailing Lists  
<http://www.eclipsecorner.org/mail/>
  - FAQ  
<http://www.eclipse.org/eclipse/faq/eclipse-faq.html>
  - IBM WebSphere Studio Family  
<http://www-3.ibm.com/software/ad/adstudio/>
  - Eclipse Plugins  
<http://eclipse-plugins.2y.net/>
  - Newsgroups  
<news://www.eclipse.org/eclipse.tools>  
<news://news.software.ibm.com/ibm.software.websphere.studio.application-site-developer>
- Other
  - Instantiations' Web Site  
<http://www.instantiations.com/>
  - Me ;-)  
<mailto:clayberg@instantiations.com>