The Eclipse Project 2005-2006

Kevin Haaland
Eclipse Project PMC,
IBM Rational Software

March 30, 2006

Project Highlights

- Eclipse 3.1 delivered June ’05
  - Million download challenge
  - Team promises to be nicer to Eclipse webmaster
- Eclipse 3.2 development starts July ’05
  - Development organized around six key themes,
  - 5000 defects resolved so far
- BEA contributes Java 5 annotation processing support to JDT
- Support for new platforms and JRE’s
  - HP provides Eclipse 3.1.1 for HP-UX, IA 64_32/Motif,
  - Intel Mac
  - Java 5 reference platforms
  - Minimum JRE requirements identified
- Equinox becomes part of The Eclipse Project

Equinox in 3.2

- Give OSGi a first class presence on eclipse.org
  - "Eclipse Foundation Announces Support for OSGi Service Platform R4 Specification" October 11, 2005"
  - Equinox as a standalone OSGi implementation
  - Make it easy for developers to reuse the Eclipse OSGi implementation
- Runtime refactored into several plug-ins
  - More flexible (e.g. extension registry is separate)
- Several add-on services from OSGi specification implemented
  - www.eclipse.org/equinox/bundles

PDE

- **OSGi Bundle Tooling**
  - A new launcher is available to run and debug bundles with the Equinox OSGi framework

- **Plug-in Manifest Tools**
  - PDE now participates in search and refactoring
  - Quick fixes and cleanup support. For example “Organize Manifest” will remove unused dependencies and property keys
  - Support for validating build.properties files added
  - NLS tool for extracting translatable strings from manifest files

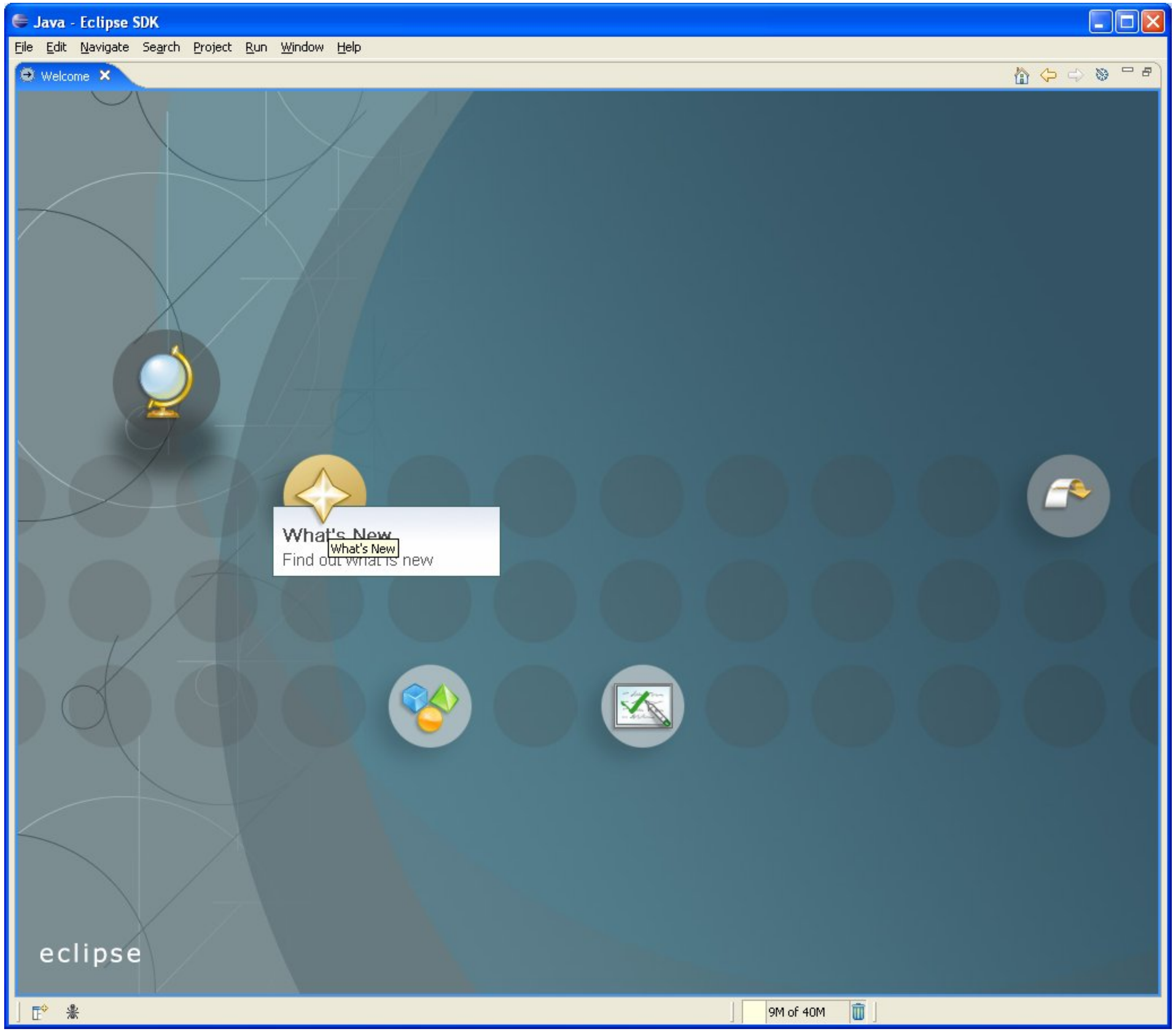
Platform - Core

- Logical Models
  - Support for interacting with logical models built above the resource layer. This allows tools that operate on files, such as repository tools, to integrate with logical model objects that don't map 1-1 with files.
  - Exploited in 3.2 by the Package Explorer, Team CVS, the Common Navigator and Team History View
- Flexible Workspaces
  - A new API for communicating with an arbitrary file system. The resources plug-in has been rewritten on top of this API to allow resources to be created that reside on other file systems.

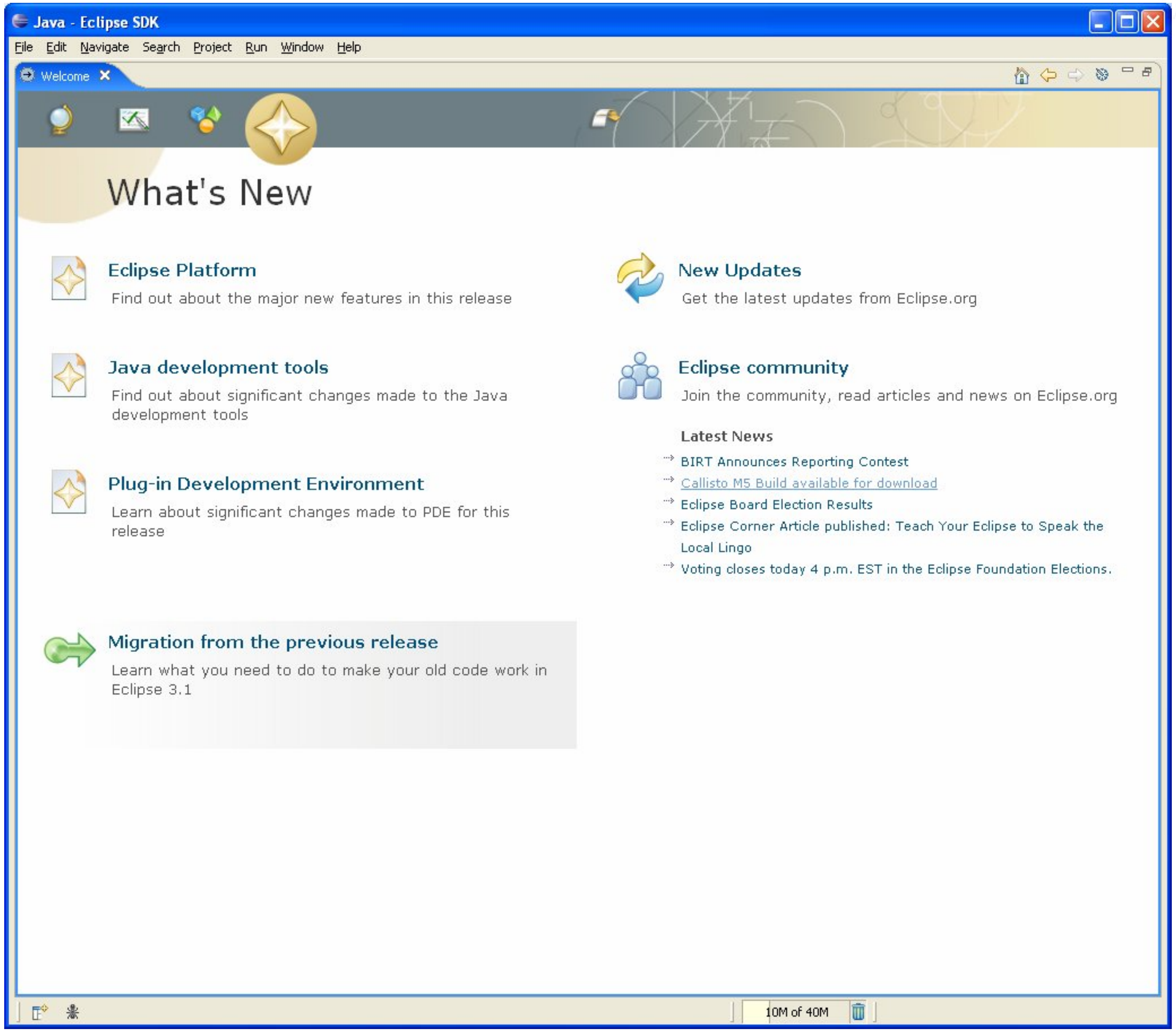
Platform – User Assistance

- Dynamic help content
  - Filtering supported based on running platform, enabled capabilities etc.
  - Plug-ins can contribute XHTML fragments that plug into existing documents
- Universal Welcome
  - Pluggable theme support
  - Configurable by products and end users

Welcome - ‘Circles’ Theme



Welcome – What's New



What's New

Eclipse Platform

Find out about the major new features in this release

Java development tools

Find out about significant changes made to the Java development tools

Plug-in Development Environment

Learn about significant changes made to PDE for this release

Migration from the previous release

Learn what you need to do to make your old code work in Eclipse 3.1

New Updates

Get the latest updates from Eclipse.org

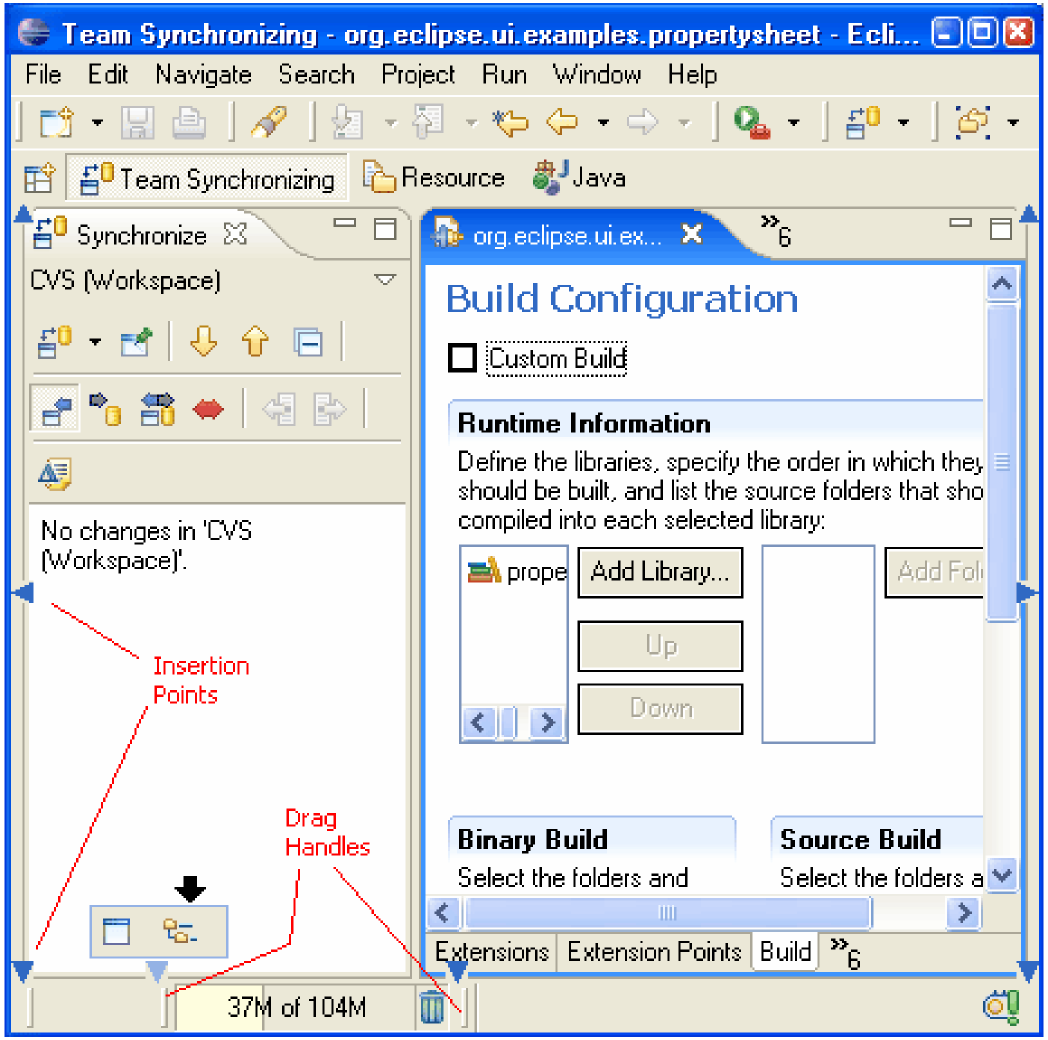
Eclipse community

Join the community, read articles and news on Eclipse.org

Latest News

- BIRT Announces Reporting Contest
- Callisto M5 Build available for download
- Eclipse Board Election Results
- Eclipse Corner Article published: Teach Your Eclipse to Speak the Local Lingo
- Voting closes today 4 p.m. EST in the Eclipse Foundation Elections.

Platform UI - More Flexible



Team Synchronizing - org.eclipse.ui.examples.propertysheet - Ecli...

File Edit Navigate Search Project Run Window Help

Team Synchronizing | Resource | Java

Synchronize

CVS (Workspace)

No changes in 'CVS (Workspace)'.

org.eclipse.ui.ex...

Build Configuration

□ Custom Build

Runtime Information

Define the libraries, specify the order in which they should be built, and list the source folders that sho compiled into each selected library:

prope | Add Library... | Up | Down | Add Fol

Binary Build

Select the folders and

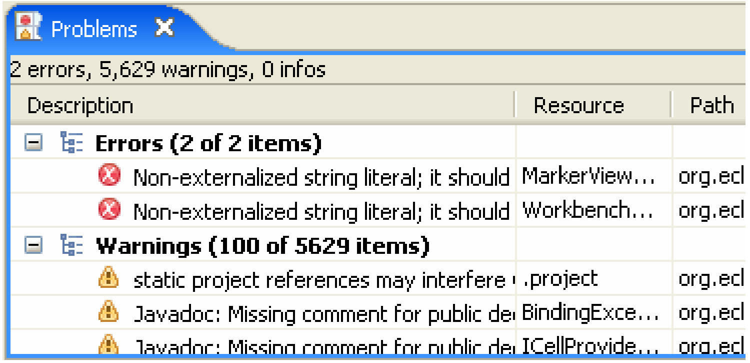
Source Build

Select the folders a

Extensions | Extension Points | Build

37M of 104M

Platform UI – More Scalable

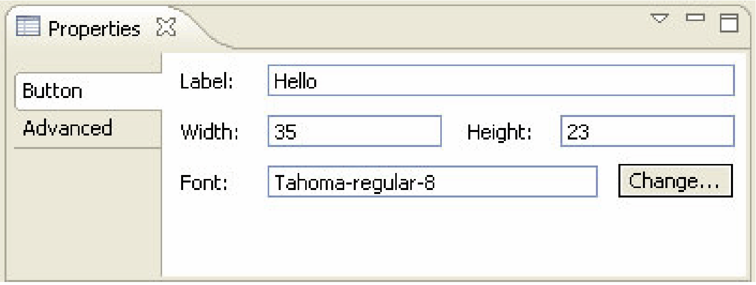


Problems

2 errors, 5,629 warnings, 0 infos

Description	Resource	Path
⊟ **Errors (2 of 2 items)**		
⊗ Non-externalized string literal; it should	MarkerView...	org.ecl
⊗ Non-externalized string literal; it should	Workbench...	org.ecl
⊟ **Warnings (100 of 5629 items)**		
⚠ static project references may interfere \	.project	org.ecl
⚠ Javadoc: Missing comment for public de	BindingExce...	org.ecl
⚠ Javadoc: Missing comment for public de	ICellProvide...	org.ecl

Problems view supports grouping

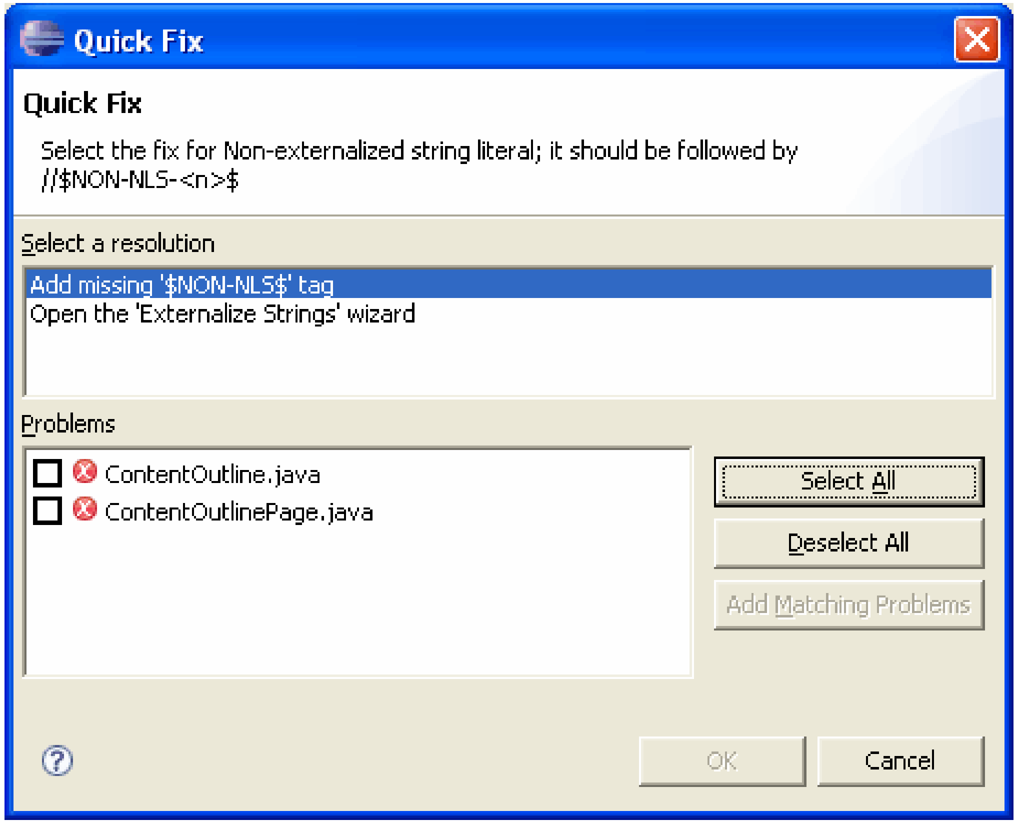


Label: Hello

Width: 35 Height: 23

Font: Tahoma-regular-8

Tabbed Properties



Quick Fix

Quick Fix

Select the fix for Non-externalized string literal; it should be followed by //$NON-NLS-<n>$

Select a resolution

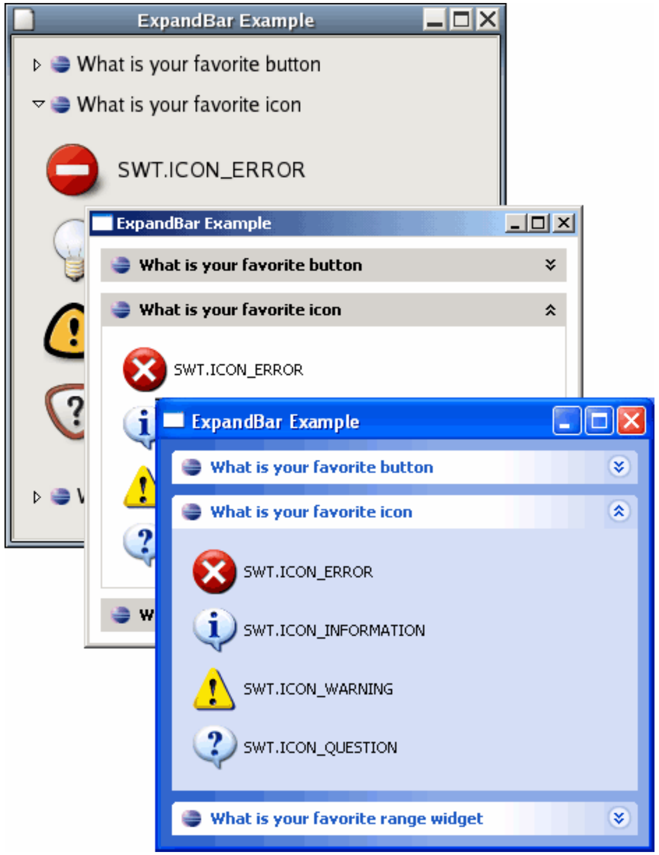
Add missing '$NON-NLS$' tag
Open the 'Externalize Strings' wizard

Problems

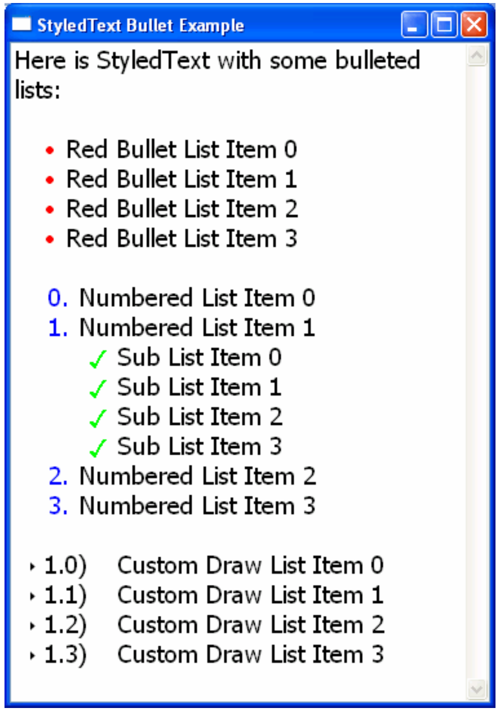
□ ⊗ ContentOutline.java
□ ⊗ ContentOutlinePage.java

Quick Fix multiple problems

SWT


ExpandBar Example
What is your favorite button
What is your favorite icon
SWT.ICON_ERROR
ExpandBar Example
What is your favorite button
What is your favorite icon
SWT.ICON_ERROR
ExpandBar Example
What is your favorite button
What is your favorite icon
SWT.ICON_ERROR
SWT.ICON_INFORMATION
SWT.ICON_WARNING
SWT.ICON_QUESTION
What is your favorite range widget

New Widgets: e.g. ExpandBar

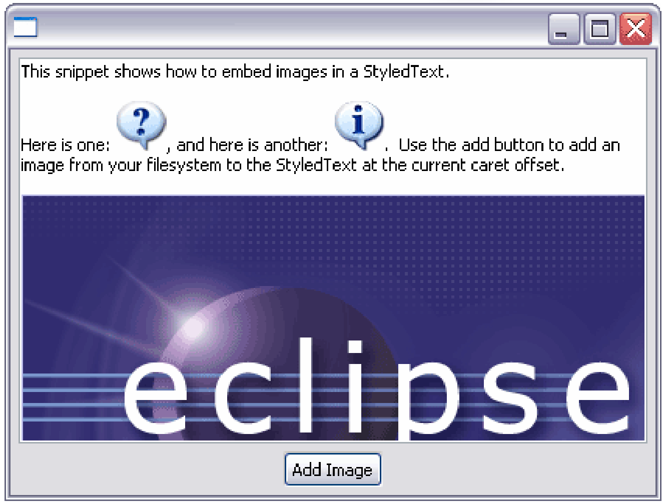


Here is StyledText with some bulleted lists:

- Red Bullet List Item 0
- Red Bullet List Item 1
- Red Bullet List Item 2
- Red Bullet List Item 3

0. Numbered List Item 0
1. Numbered List Item 1
   - ✓ Sub List Item 0
   - ✓ Sub List Item 1
   - ✓ Sub List Item 2
   - ✓ Sub List Item 3
2. Numbered List Item 2
3. Numbered List Item 3

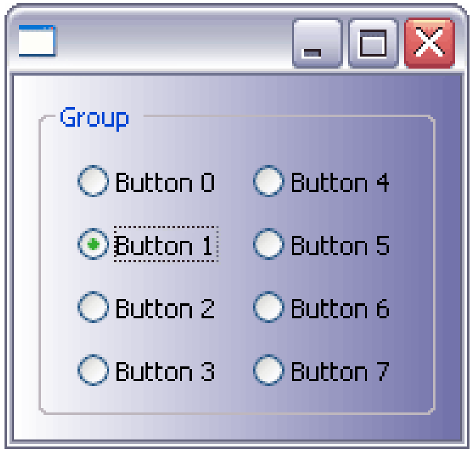
- 1.0) Custom Draw List Item 0
- 1.1) Custom Draw List Item 1
- 1.2) Custom Draw List Item 2
- 1.3) Custom Draw List Item 3



This snippet shows how to embed images in a StyledText.

Here is one: ? , and here is another: i . Use the add button to add an image from your filesystem to the StyledText at the current caret offset.

New Styled Text features. For example: embedded objects, and bullet items

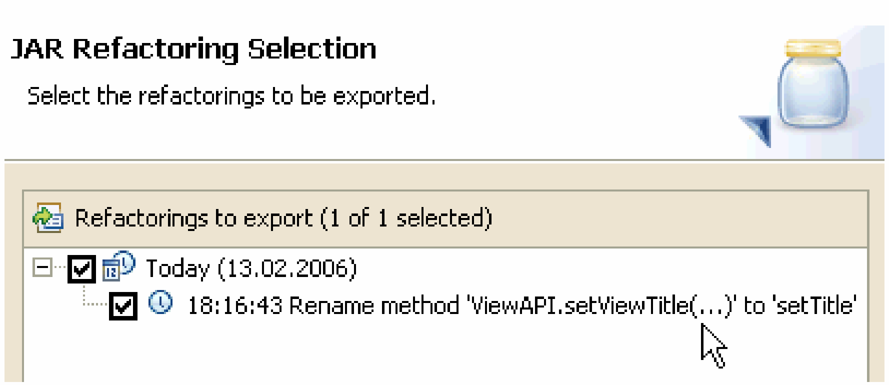


Setting background image in controls

JDT Core

- **Enable compiler participation.**
  - JDT compilation technology opened up to enable pluggable participation.
  - Compiler participants able to introspect the Java code using DOM/AST API, perform semantic analysis, create markers etc.
- **Add support for Java SE 6 features.**
  - Compiler compliance level can be set to 6.0
- **More static analysis.**
  - Code quality can be further improved by new advanced configurable compiler diagnostics, available either when building or when editing Java files.
  - The compiler will conduct some null reference analysis in order to anticipate some NullPointerException at runtime.
  - For cleaning up code, the compiler will detect obsolete $NON-NLS$ tags and unused break/continue label.
  - ++ more

JDT – Refactoring and Clean Up Wizard



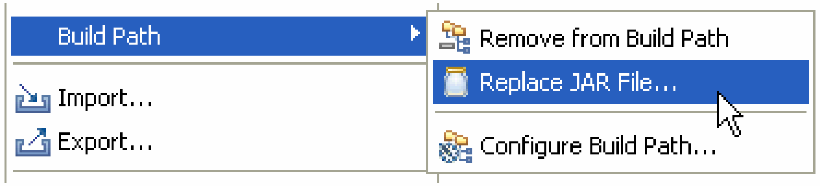
JAR Refactoring Selection

Select the refactorings to be exported.

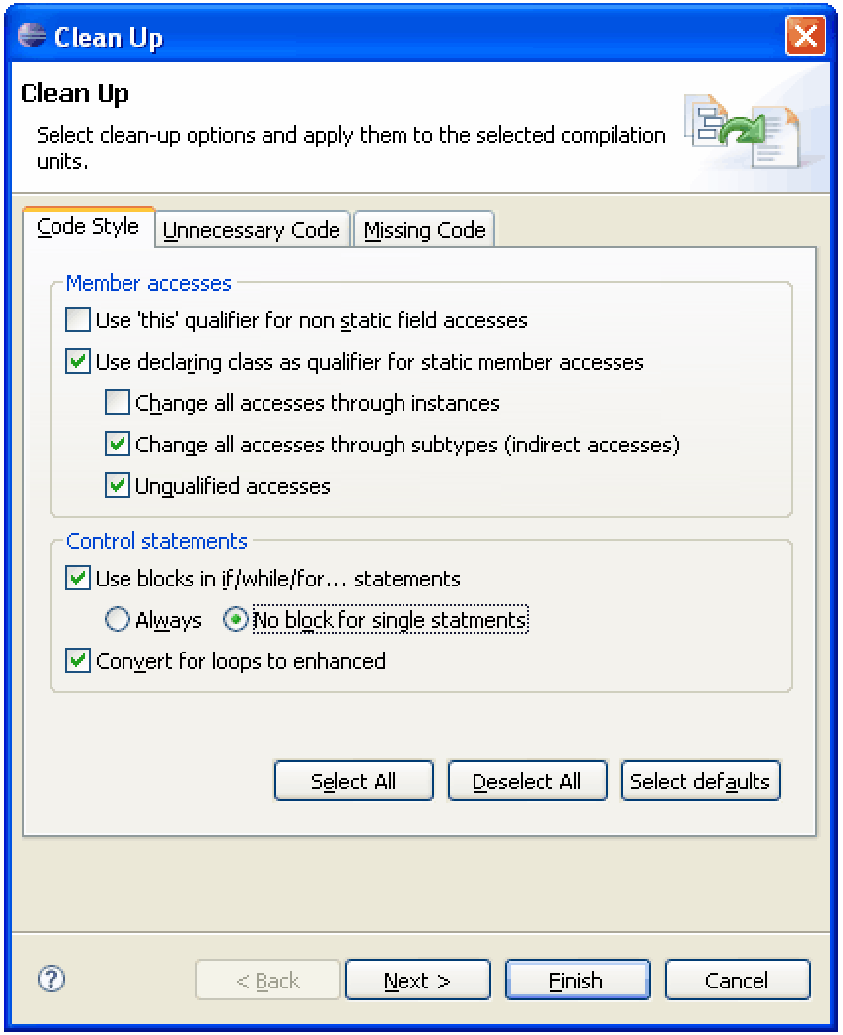
Refactorings to export (1 of 1 selected)

- ☑ Today (13.02.2006)
  - ☑ 18:16:43 Rename method 'ViewAPI.setViewTitle(...)' to 'setTitle'

When updating a project to a new and improved JAR, select a JAR in the Package Explorer and use **Build Path > Replace JAR File...**



This will automatically offer to replay the stored refactoring on the source files in the project.



Clean Up

Select clean-up options and apply them to the selected compilation units.

Code Style | Unnecessary Code | Missing Code

Member accesses

- □ Use 'this' qualifier for non static field accesses
- ☑ Use declaring class as qualifier for static member accesses
  - □ Change all accesses through instances
  - ☑ Change all accesses through subtypes (indirect accesses)
  - ☑ Unqualified accesses

Control statements

- ☑ Use blocks in if/while/for... statements
  - ○ Always ● No block for single statments
- ☑ Convert for loops to enhanced

THANK YOU!!!