Being more productive with Eclipse 3.x

Why I finally (mostly) switched from Emacs

dave@marquam.org

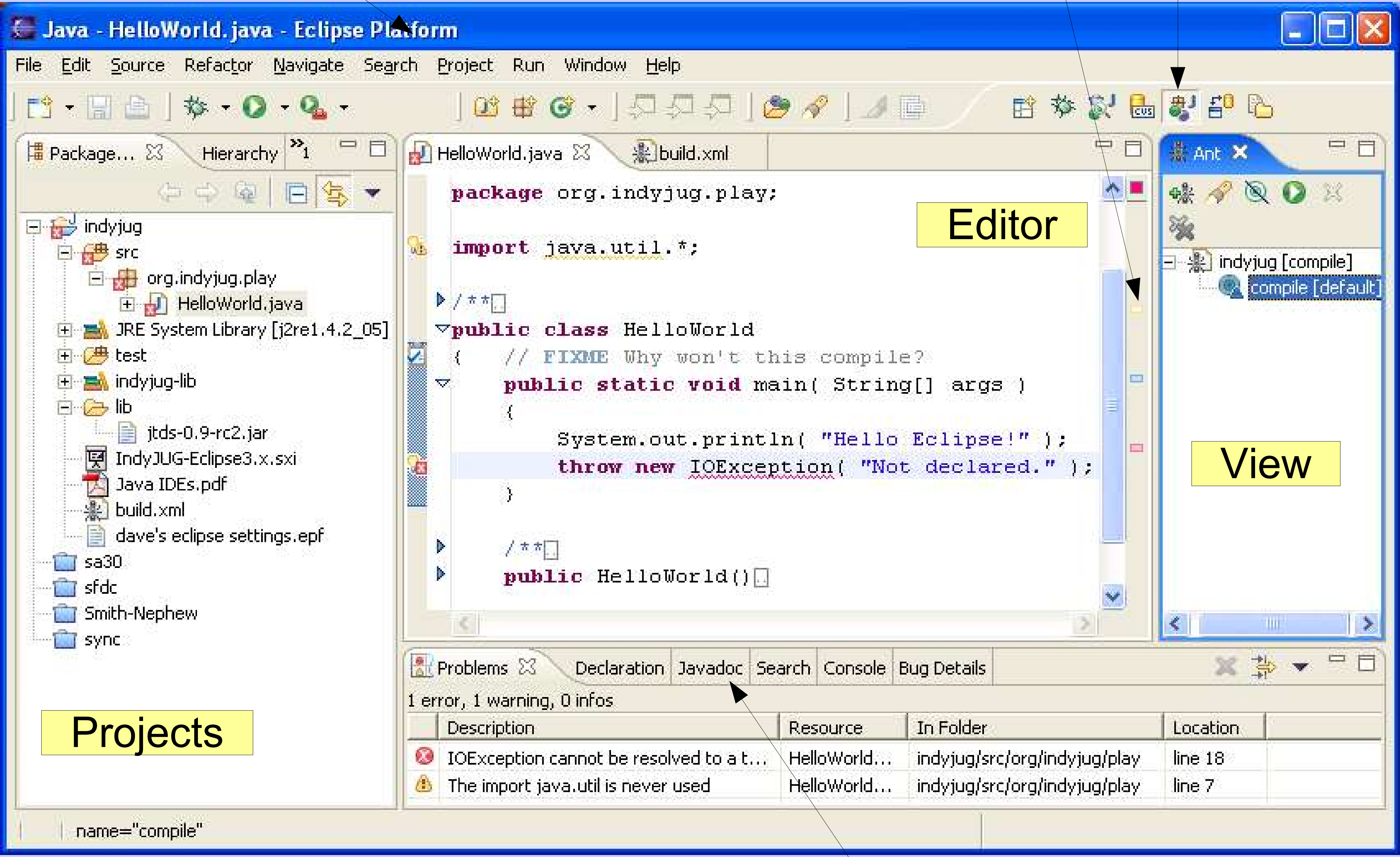
Feature Overview

- Very usable UI
- Exhaustive configurability
- Refactoring editor
- Auto-complete
- Auto-fix
- Real-time syntax checking
- Integration with CVS
- Powerful debugger
- Integration with Ant, JUnit
- Flexible project organization
- Flexible Search
- Flexible plug-in architecture
- Extensive help

UI Concepts

- Workbench
- Projects
- Perspectives
- Views
- Editors
- Annotations and Decorations
- Key Bindings

First Look



```
package org.indyjug.play;

import java.util.*;

/**□
public class HelloWorld
{    // FIXME Why won't this compile?
    public static void main( String[] args )
    {
        System.out.println( "Hello Eclipse!" );
        throw new IOException( "Not declared." );
    }

    /**□
    public HelloWorld()□
```

1 error, 1 warning, 0 infos

Description	Resource	In Folder	Location
IOException cannot be resolved to a t...	HelloWorld...	indyjug/src/org/indyjug/play	line 18
The import java.util is never used	HelloWorld...	indyjug/src/org/indyjug/play	line 7

Workbench

Annotations

Perspectives

Editor

View

Projects

More Views

Projects

- Flexible definition
  - Source directories
  - Class directory
  - Dependant Libraries
  - Per project compiler options
- Organizes Resources with projects
- Define dependencies between projects
- Define reusable groups of libraries

Perspectives

- Resource
- Java
- Java Browsing
- Debug
- CVS
- Team Synchronization (with CVS)
- Plug-in development

Views

- Package Explorer
- Hierarchy
- JUnit
- Ant
- Console
- Problems
- Java Doc
- Search
- Team Synchronization
- CVS Resource history
- Cheat sheets
- Debugging Views
  - Threads
  - Variables
  - Breakpoints
  - Expressions

Editors

- Java
- XML
- Ant
- Text
- Readme
- Plugins for JSP
- External editors

Annotations

- Errors
- Warnings
- Tasks
- Search Results
- Spelling Errors
- Bookmarks
- Others I haven't used

Decorations

- CVS Modified
- CVS New
- CVS Revision
- Method override

Editing Java

- The usual functions (cut, copy, paste, undo, redo)
- Auto complete (after . or ctrl-space)
- Auto match (, [, { etc.
- Goto declaration (class, method, field etc.)
- Auto correct (ctrl-1)
- Auto build (optionally)
- Auto format
- Organize Imports (ctrl-shift-O)
- Code wizards driven by configurable templates
- Refactoring with previews

Refactoring Java

- Rename (Alt-Shift-R)
- Move (Alt-Shift-V)
- Change Method Signature
- Push Down
- Pull Up
- Extract Interface
- Generalize Type
- Inline
- Encapsulate Field
- Extract Method
- Extract Variable
- Extract Constant
- Introduce Factory
- Convert Anonymous Class to Nested
- Move Nested Class to Own File
- Use Supertype Where Possible

Quick Fixes

- Add Javadoc
- Add import
- Create local variable
- Create missing method
- Create missing field
- Add return
- Make class abstract
- Correct non-static access for static members
- Initialize uninitialized variable
- Remove unused catch
- Surround with try/catch
- Add throws declaration to method
- Add Cast
- Externalize String
- Others...

Searching

- File
  - By name
  - By text
  - Filtered by sets
- Java
  - By name
  - For: Package, Class, Method, Constructor, Field
  - Limited to: Declarations, References, Implementors
- Help

- Creates a list of matches in a view
- Marks occurrences in editors

Code Style

- Indentation
- Braces
- White Space
- Blank Lines
- New Lines
- Control Statements
- Line Wrapping
- Comments
- Import/Export

Debugging

- Types of sessions
  - Applet
  - Application
  - JUnit
  - Remote
- Arguments
- Classpath
- Source locations
- Environment
- Multiple sessions
- Multiple threads
- Breakpoints
- Variables
- Step Filters
- Evaluate expressions
- Console
- JUnit results

Ant

- Ant View
- Ant Editor
- Defined as an external tool

JUnit

- Run As/Debug As
- JUnit View for results

Other tools

- Generate Javadoc
- Compare files
- Convert Delimiters
- Export as Jar/Zip, File System, FTP, etc.
- Import from CVS, File System, FTP, etc.
- External Tools
  - Refresh upon completion

CVS Integration

- Team Synchronization Perspective
- Resource History View
- Shows Difference between repository and workspace
- Marks conflicts and helps in resolving them
- Drives better integration behavior

- Synchronize
- Update
- Commit
- Override and Update
- Override and Commit
- Mark as Merged

Plug-ins

- Eclipse is a micro-kernel
- Everything is a plug-in (including “features”)
- Install via Feature Updates wizard or by unpacking a zip file into plugins directory
- Huge number of plugins
  - FindBugs
  - Antlr
  - Hibernate Synchronizer
  - New IBM contribution for J2EE

References

Eclipse help

Eclipse Project Presentation

- http://eclipse.org/eclipse/presentation/eclipse-slides_files/v3_document.htm

Working with Hibernate in Eclipse

- http://www.onjava.com/lpt/a/4949

http://eclipse.org/

http://www.eclipseplugincentral.com/

http://www.eclipse-plugins.info/