**Configuration Management, Tools and Standards**

**for**

**ZetaFish**

Version 1.4

December 5, 2010

Prepared by:

Patrick Michalina

Malick Noor Chad Albrecht



**Team Zeta Dragons**

CSC 478B - Software Engineering Capstone

Fall 2010

University of Illinois at Springfield



Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Chad Albrecht | 9/20/2010 | Initial Version | 1.0 |
| Chad Albrecht | 11/22/2010 | Added more specifics on CM | 1.1 |
| Chad Albrecht | 11/27/2010 | Added Version Format detail | 1.2 |
| Chad Albrecht | 12/4/2010 | Added links and more detail | 1.3 |
| Patrick Michalina | 12/5/2010 | Added Mac JAR Wrapper content and additional detail. | 1.4 |

Table of Contents

[1 Introduction 1](#_Toc279312253)

[2 Getting Started 1](#_Toc279312254)

[3 Versioning Format Details 1](#_Toc279312255)

[4 Development Tool Details 2](#_Toc279312256)

[4.1 Development Language 2](#_Toc279312257)

[4.2 Integrated Development Environments (IDEs) 2](#_Toc279312258)

[4.2.1 IDE Choice #1 - Eclipse 3](#_Toc279312259)

[4.2.2 IDE Choice #2 - NetBeans 4](#_Toc279312260)

[4.3 Additional Tools 5](#_Toc279312261)

[4.3.1 PC JAR Wrapper - JSmooth 5](#_Toc279312262)

[4.3.2 Mac JAR Wrapper – Apple’s JAR Bundler 5](#_Toc279312263)

[4.3.3 Code Documentation - JavaDoc 5](#_Toc279312264)

[5 Configuration Management Details 6](#_Toc279312265)

[5.1 Configuration Management Platform 6](#_Toc279312266)

[5.2 Revision Control System (RCS) - SVN 6](#_Toc279312267)

[5.3 RCS Project Host / Repository – Google Code 6](#_Toc279312268)

[5.4 SVN Client Installation Instructions 6](#_Toc279312269)

[5.4.1 Installing Subclipse 6](#_Toc279312270)

[5.4.2 NetBeans SubVersion Client 7](#_Toc279312271)

[6 Design and Modeling Tools 8](#_Toc279312272)

[6.1 General Diagrams – Microsoft Visio 8](#_Toc279312273)

[6.2 UML Models - MoDisco 8](#_Toc279312274)

[6.3 UML Diagrams – UML2 Tools 9](#_Toc279312275)

[6.4 UI Modeling - Microsoft Expression Blend/SketchFlow 9](#_Toc279312276)

[7 Coding Standards 10](#_Toc279312277)

[7.1 General 10](#_Toc279312278)

[7.2 Comments 10](#_Toc279312279)

[7.3 Method Headers 10](#_Toc279312280)

[7.4 Design Traceability 10](#_Toc279312281)

[8 Cited References 12](#_Toc279312282)

This page intentionally left blank.

# Introduction

This document describes all the tools and technologies used to enable development of the ZetaFish product.

# Getting Started

Rather than trying to understand how each of the sections contained in this document fit together, a developer should follow these steps in order to setup his/her development environment.

1. Install Eclipse or optionally NetBeans – Section 4.2.
2. Based on the IDE choice, install and configure the SVN client. – Section 5.4.
3. Get the HEAD version of the project from the repository.
4. Run ZetaFish in Debug mode.
5. Review the Versioning Format Details – Section 3.
6. Make changes as required.
7. Commit changes back to the repository.

# Versioning Format Details

For the purpose of consistency in identifying specific sets of related source files, the following versioning format should be used.

{Major}.{Minor}.{Revision}

Figure - Version Format

|  |  |
| --- | --- |
| Version Part | Description |
| *Major* | Incremented for significant addition of feature/functionality. |
| *Minor* | Incremented for slight modification to existing feature/functionality. Reset to zero on an increment of the Major number. |
| *Revision* | Incremented on bug fix or any rebuild. Reset to zero on an increment of the Major number. |

Table - Version Part Descriptions

# Development Tool Details

|  |  |
| --- | --- |
| Function | Tool/Technology |
| Development IDE | Eclipse or NetBeans |
| PC JAR Wrapper | JSmooth |
| Mac JAR Wrapper | Apple’s Jar Bundler |
| Code Documentation | JavaDoc |

Table - Development Tool Choices

## Development Language

Java. Using a compiler capable of compliance level 1.6 or better.[[1]](#endnote-1) Additionally a Java Runtime Environment(JRE) of 6 or better will be required.

## Integrated Development Environments (IDEs)

Eclipse or NetBeans. Given the varying backgrounds and skill sets on this project, we felt having a selection of IDEs would prove beneficial over the course of the semester. As such, we selected to use either Eclipse or NetBeans which are discussed in upcoming sections.

### IDE Choice #1 - Eclipse

Eclipse is a full featured Java development IDE that has a substantial open-source development community.[[2]](#endnote-2)

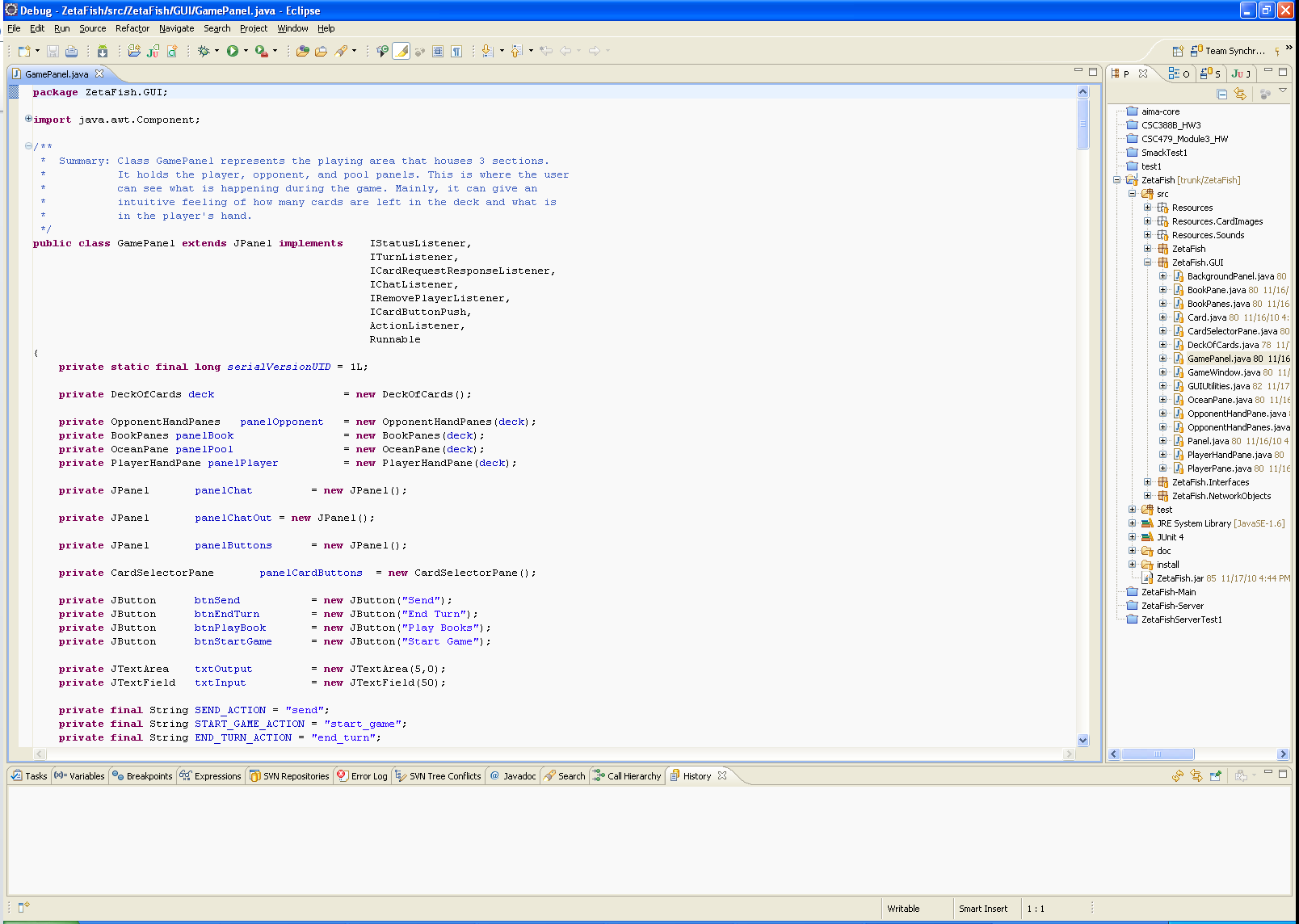


Figure – Eclipse

Eclipse Java EE IDE for Web Developers, Build id: 20100218-1602

### IDE Choice #2 - NetBeans

The NetBeans IDE is an award-winning integrated development environment available for Windows, Mac, Linux, and Solaris. The NetBeans project consists of an open-source IDE and an application platform that enable developers to rapidly create web, enterprise, desktop, and mobile applications using the Java platform, as well as JavaFX, PHP, JavaScript and Ajax, Ruby and Ruby on Rails, Groovy and Grails, and C/C++.

The NetBeans project is supported by a vibrant developer community and offers extensive documentation and training resources as well as a diverse selection of third-party plugins.[[3]](#endnote-3)

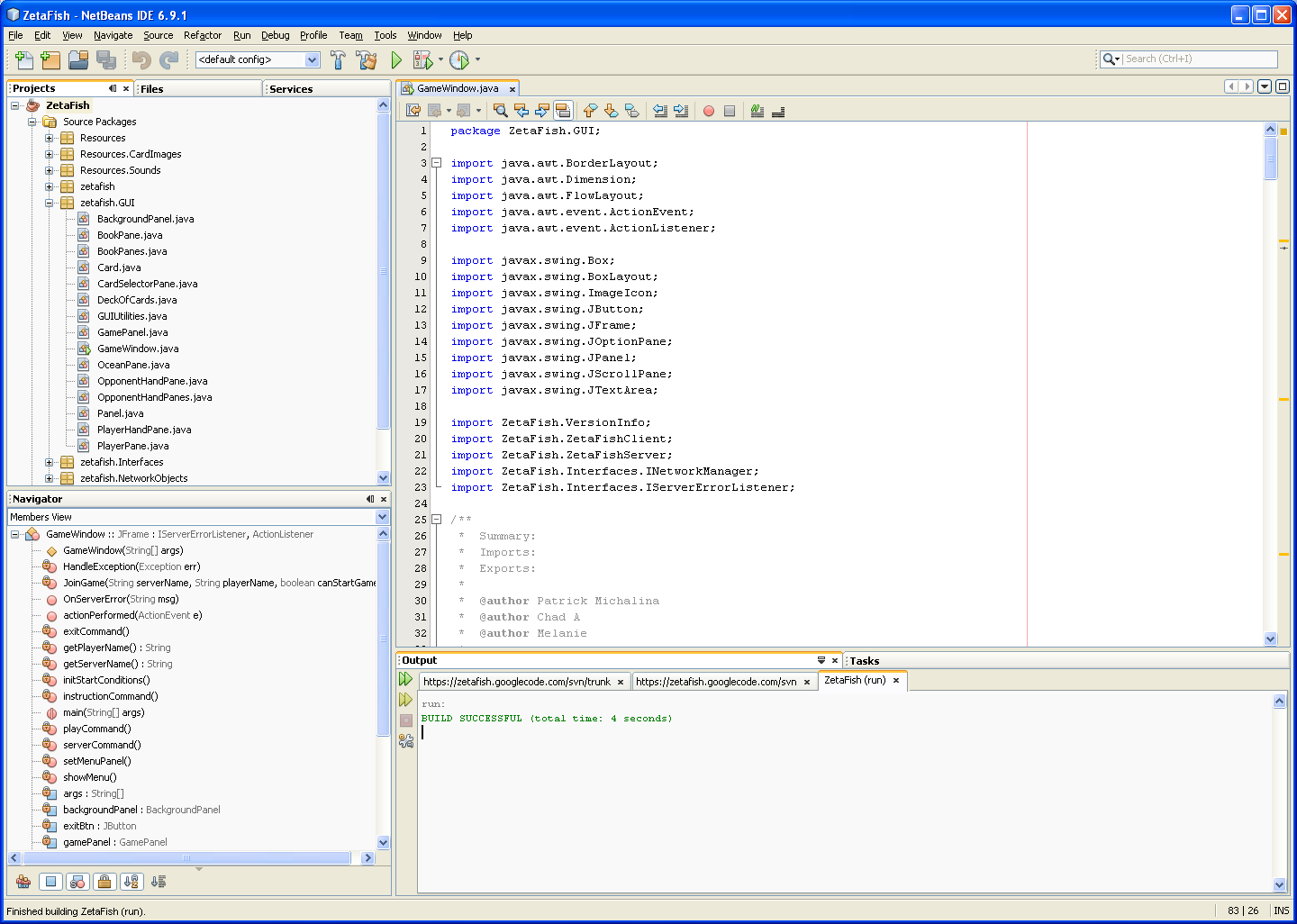


Figure - NetBeans IDE

NetBeans IDE 6.9.1 (Build 201011082200)

## Additional Tools

### PC JAR Wrapper - JSmooth

JSmooth is a Java Executable Wrapper. It creates native Windows launchers (standard .exe) for your java applications. It makes java deployment much smoother and user-friendly, as it is able to find any installed Java VM by itself.

When no VM is available, the wrapper can automatically download and install a suitable JVM, or simply display a message or redirect the user to a web site.[[4]](#endnote-4)

For more information see:

<http://jsmooth.sourceforge.net/>

### Mac JAR Wrapper – Apple’s JAR Bundler

Apple's Jar Bundler is a Java Mac OSX Wrapper. It creates native Mac OS X launchers (standard .app) from most JAR files. It makes Java deployment much smoother and user-friendly. Java updates are integrated into all system updates in OSX so most users should have a newer versions of Java installed.[[5]](#endnote-5)

For more information see:

<http://developer.apple.com/library/mac/#documentation/Java/Conceptual/Jar_Bundler/About/About.html>

### Code Documentation - JavaDoc

Javadoc is a tool for generating API documentation in HTML format from doc comments in source code.[[6]](#endnote-6)

For more information see:

<http://www.oracle.com/technetwork/java/javase/documentation/index-jsp-135444.html>

# Configuration Management Details

## Configuration Management Platform

Through various discussions and vetting of tools the following items were selected to enable the configuration management of ZetaFish.

|  |  |
| --- | --- |
| Function | Tool/Technology |
| Revision Control System (RCS) | SVN |
| RCS Project Host | http://code.google.com/hosting/ |

Table - CM Platform Choices

## Revision Control System (RCS) - SVN

Apache Subversion (often abbreviated SVN, after the command name svn) is a software versioning and a revision control system founded and sponsored in 2000 by CollabNet Inc. Developers use Subversion to maintain current and historical versions of files such as source code, web pages, and documentation. Its goal is to be a mostly-compatible successor to the widely used Concurrent Versions System (CVS).[[7]](#endnote-7),[[8]](#endnote-8)

## RCS Project Host / Repository – Google Code

Project Hosting on Google Code provides a free collaborative development environment for open source projects. Each project comes with its own member controls, Subversion/Mercurial repository, issue tracker, wiki pages, and downloads section. Our project hosting service is simple, fast, reliable, and scalable, so that you can focus on your own open source development.[[9]](#endnote-9)

The ZetaFish project will use the SVN interface on Google Code to perform version control of the source code. The location of the project is:

<http://code.google.com/p/zetafish/>

## SVN Client Installation Instructions

### Installing Subclipse

Subclipse is an Eclipse Team Provider plug-in providing support for Subversion within the Eclipse IDE.[[10]](#endnote-10)

1. Download Eclipse if you haven't already done so
2. Go to the Subclipse homepage and look up the Eclipse update site URL
3. Open Eclipse
4. Click Help -> Software Updates -> Find and Install...
5. Select Search for new features to install
6. Click Next
7. Click New Remote Site
8. Type in a name of your choice in the Name text field and copy-paste the Eclipse update site URL in the URL text field
9. Click OK
10. Select the new site you just created.
11. Click Finish
12. Expand the update site tree
13. Select the Subclipse plugin
14. Click Next and follow instructions.

Adding the ZetaFish repository in Subclipse

1. Open Eclipse
2. Click the small Open Perspective icon located in the top right hand corner of the IDE.
3. Choose Other and select the SVN Repositry Exploring perspective
4. Right click in the left SVN Repository pane and select New -> Repository Location...
5. In the URL text field type in https://zetafish.googlecode.com/svn
6. Click Finish

Creating the ZetaFish checkout Project

1. Open Eclipse
2. Click File -> New -> Project...
3. Select SVN -> Checkout projects from SVN
4. Click Next
5. Select (click) the zetafish repository
6. Click Next
7. Select the trunk folder
8. Click Next
9. Click Finish
10. Select Java Project
11. Click Next
12. Type in ZetaFish for the project name
13. Click Finish

### NetBeans SubVersion Client

NetBeans IDE provides tight integration with Subversion client versions 1.5.x and higher. The IDE's Subversion support is designed to help streamline the development process for groups working from a shared repository, enabling you to perform versioning tasks directly from your project system within the IDE.[[11]](#endnote-11)

For more information see:

<http://netbeans.org/kb/docs/ide/subversion.html>

# Design and Modeling Tools

|  |  |
| --- | --- |
| Function | Tool/Technology |
| General Diagrams | Microsoft Viso |
| UML Models | MoDisco |
| UML Diagrams | UML2 Tools |
| UI Modeling | Microsoft Expression Blend/SketchFlow |

Table - Design and Modeling Tool Choices

## General Diagrams – Microsoft Visio

Microsoft Visio is a diagramming program for Microsoft Windows that uses vector graphics to create diagrams.[[12]](#endnote-12)

For more information see:

<http://office.microsoft.com/en-us/visio/>

Visio can be downloaded as part of the Microsoft Dream Spark program available to students.

<https://www.dreamspark.com>

## UML Models - MoDisco

MoDisco provides a framework to develop model-driven tools for software modernization. It aims at supporting use-cases such as Quality Assurance, Documentation, Improvement and Migration.[[13]](#endnote-13)

For more information see:

<http://www.eclipse.org/MoDisco/>

## UML Diagrams – UML2 Tools

UML2Tools allow the creation of UML diagrams within the Eclipse IDE.

For more information see:

<http://wiki.eclipse.org/MDT-UML2Tools>

## UI Modeling - Microsoft Expression Blend/SketchFlow

SketchFlow, a feature of Expression Studio Ultimate, gives you the ability to quickly and effectively map out and iterate the flow of an application UI, the layout of individual screens and perhaps most importantly for modern applications the transition from one application state to another.[[14]](#endnote-14)

For more information see:

<http://www.microsoft.com/expression/products/Sketchflow_Overview.aspx>

Expression can be downloaded as part of the Microsoft Dream Spark program available to students.

<https://www.dreamspark.com>

# Coding Standards

## General

Due to the temporary nature of this project, the team feels that implementation of specific coding standards other than documentation through comments is not necessary. While there are no specific standards, developers are encouraged to follow coding best-practices to improve readability and maintainability. These best-practices include:

1. Separation of Concerns.[[15]](#endnote-15) (SoC)
2. Don't repeat yourself.[[16]](#endnote-16) (DRY)
3. Loose coupling.[[17]](#endnote-17)
4. High cohesion.[[18]](#endnote-18)
5. Design by contract.[[19]](#endnote-19) (DbC)

## Comments

Every method within the source code should include the method header defined by Section 7.3. Additionally any complex logic within a method should include a brief description of the intent of the code.[[20]](#endnote-20)

## Method Headers

Every method in the source code should include the comment block briefly describing its intended function and/or purpose.

/\*\*

\* {Method description}

\*/

Code Block - Function Header Template

NOTE: The extra asterisk in the first line of the comment block declares the block to be used by JavaDoc. (See Section 4.3.3)

## Design Traceability

Wherever code is written specifically to implement a portion of the software design a comment block should be added to note this.

/\* (Design A.B.C} vX.Y\*/

Code Block 2 – Design Trace Template

In Code Block 1 A.B.C refers to the section number in the Conceptual Design Document and X.Y refers to the version of the Conceptual Design Document.

To aid in the location of design elements within the source code, the use of Java tags is recommended.[[21]](#endnote-21) The use of the “Design” tag is shown in Figure 4.

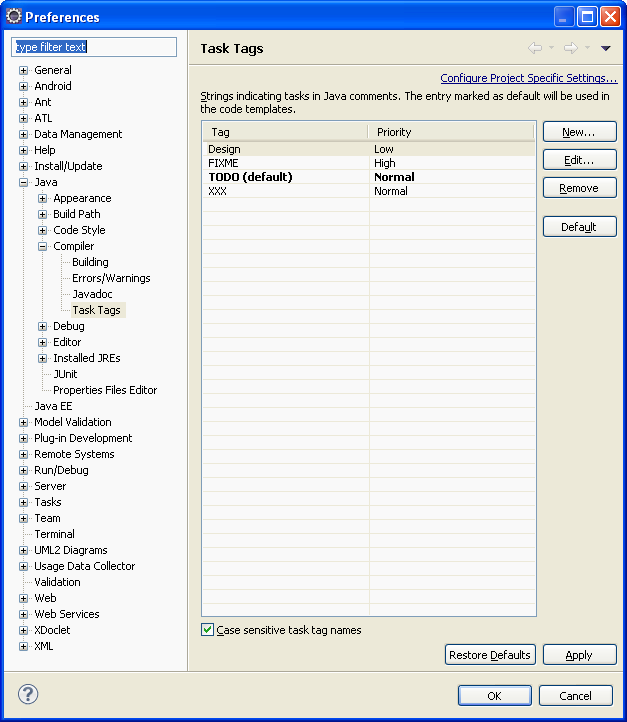


Figure 4 - Task Tag Preferences

# Cited References

1. http://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html [↑](#endnote-ref-1)
2. http://www.eclipse.org/downloads/moreinfo/java.php [↑](#endnote-ref-2)
3. http://netbeans.org [↑](#endnote-ref-3)
4. http://jsmooth.sourceforge.net/ [↑](#endnote-ref-4)
5. http://developer.apple.com/library/mac/#documentation/Java/Conceptual/Jar\_Bundler [↑](#endnote-ref-5)
6. http://www.oracle.com/technetwork/java/javase/documentation/index-jsp-135444.html [↑](#endnote-ref-6)
7. http://en.wikipedia.org/wiki/Apache\_Subversion [↑](#endnote-ref-7)
8. http://subversion.apache.org/ [↑](#endnote-ref-8)
9. http://code.google.com/p/support/wiki/GettingStarted [↑](#endnote-ref-9)
10. http://subclipse.tigris.org/ [↑](#endnote-ref-10)
11. http://netbeans.org/kb/docs/ide/subversion.html [↑](#endnote-ref-11)
12. http://en.wikipedia.org/wiki/Microsoft\_Visio [↑](#endnote-ref-12)
13. http://www.eclipse.org/modeling/mdt/?project=uml2 [↑](#endnote-ref-13)
14. http://www.microsoft.com/expression/products/Sketchflow\_Overview.aspx [↑](#endnote-ref-14)
15. http://en.wikipedia.org/wiki/Separation\_of\_concerns [↑](#endnote-ref-15)
16. http://en.wikipedia.org/wiki/Don't\_repeat\_yourself [↑](#endnote-ref-16)
17. http://en.wikipedia.org/wiki/Loose\_coupling [↑](#endnote-ref-17)
18. http://en.wikipedia.org/wiki/Cohesion\_(computer\_science) [↑](#endnote-ref-18)
19. http://en.wikipedia.org/wiki/Design\_by\_contract [↑](#endnote-ref-19)
20. http://blog.chadalbrecht.com/post/2004/10/14/Code-Commenting.aspx [↑](#endnote-ref-20)
21. http://help.eclipse.org/help33/index.jsp?topic=/org.eclipse.jdt.doc.user/reference/ref-properties-task-tags.htm [↑](#endnote-ref-21)