Guide for Developers

Unit Testing Framework - XSLT

Jacek Radajewski
$Revision: 210 $
$Date: 2006-12-11 19:25:54 +1000 (Mon, 11 Dec 2006) $

Table of Contents

1 Development Environment

1.1 Recommended Tools

This section contains a list of recommended development tools. These tools have been chosen because they are free (or very cheap) and the project has been already correctly configured to work with these tools.

Most tools listed here are optional, but some, such as a Java compiler are obviously required.

1.1.1 JDK 5.0

UTF-X uses Java 5 language features so you'll need Java 5 or later. UTF-X will not compile or run on earlier versions of Java.

1.1.2 Eclipse 3.2

Eclipse is the preferred IDE. UTF-X has already been configured to work correctly with Eclipse and the configuration files are stored in the subversion repository together with the source code. When you checkout the project from within Eclipse, Eclipse will read the configuration files and you will be ready to cut code.

1.1.3 Subversion client

In order to checkout UTF-X from within Eclipse you will need a subversion plugin called Subclipse. Detailed instruction on how to install subclipse are available here.

In addition to subclipse you may also want to install a stand alone subverion client. TortoiseSVN 1.4.1 is a Microsoft Windows shell extension which allows you perform subversion operation directly from Windows Explorer. You can download subversion source or binaries for most platforms.

NOTE

Most Linux distributions already ship with subversion.

1.1.4 Ant 1.6.5

Ant is used for various tasks including building, documentation generation and testing.

1.1.5 <oXygen/> XML Editor 7.2 plugin for Eclipse

<oXygen/> XML Editor 7.2 plugin for the Eclipse platform provides a vast range of functionality for working with XML, XSLT, XSL:FO, XML Schemas and XPath. If you are mainly a Java developer and are not planning on doing a lot of work with XML and XSLT then functionalities provided by may be sufficient to you. Also, you may already have a different XML/XSL authoring tool such as XML Spy.

1.2 Configuring Your Environment

This section explains how to configure your development environment for UTF-X development. This section only covers general configuration and tools listed in the 1.1 Recommended Tools section.

1.2.1 Checkout project from subversion

This section shows how to checkout a UTF-X module from the subversion repository using the Eclipse 3.2 IDE. The subversion modules contain Eclipse configuration files and if you follow the process described below your project environment will be configured correctly after the initial checkout.

When checking the `HEAD` of the tree (or trunk) please make sure you checkout the `trunk` directory as shown in a screenshot below.

Our subversion repository is at https://utf-x.svn.sourceforge.net/svnroot/utf-x

Subversion repository contains the following modules:

1. `utf-x-framework` the framework
2. `utf-x-doc` the UTF-X Documentation package
3. `utf-x-website` UTF-X web site at sourceforge.net

The following is a step-by-step guide to checking out project from within Eclipse:

From the File menu select 'New Project'. A windows wil pop up in which you need to expand the SVN item and select 'Checkout Projects from SVN'.

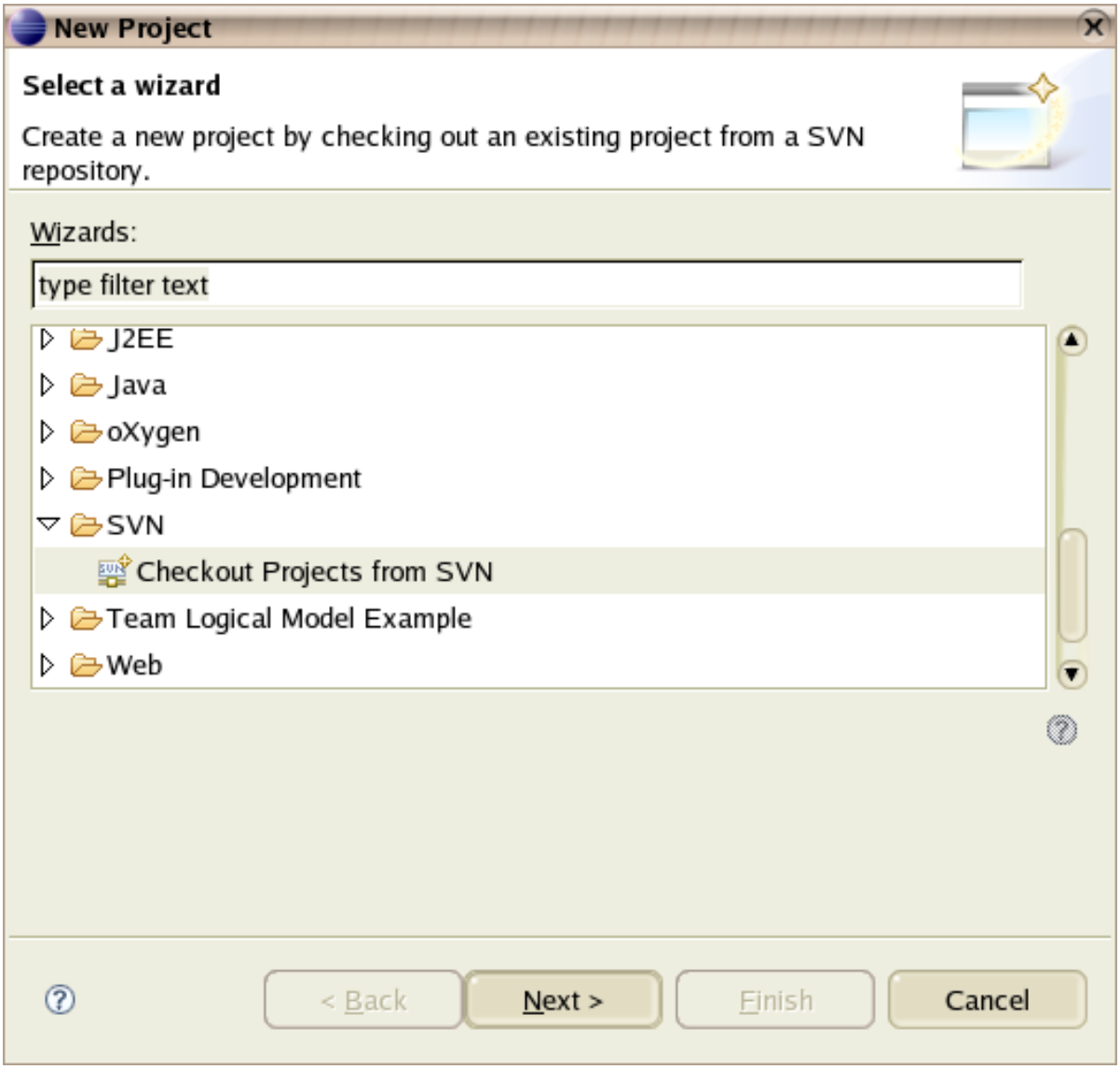

Figure 1.1: 'Checkout Projects from SVN'

Click 'Next' and select 'Create a new repository location'.

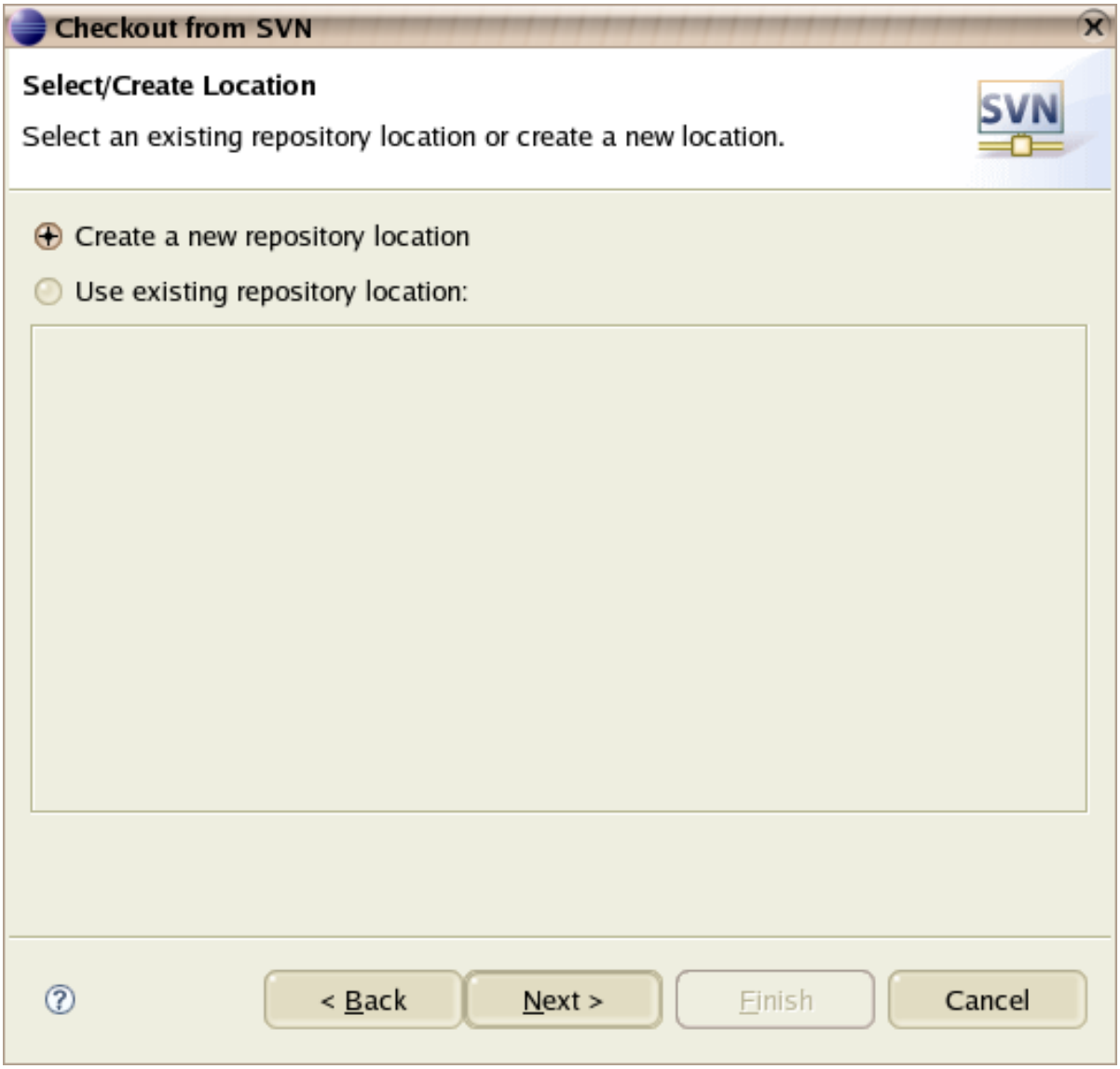

Figure 1.2: Create a repository location

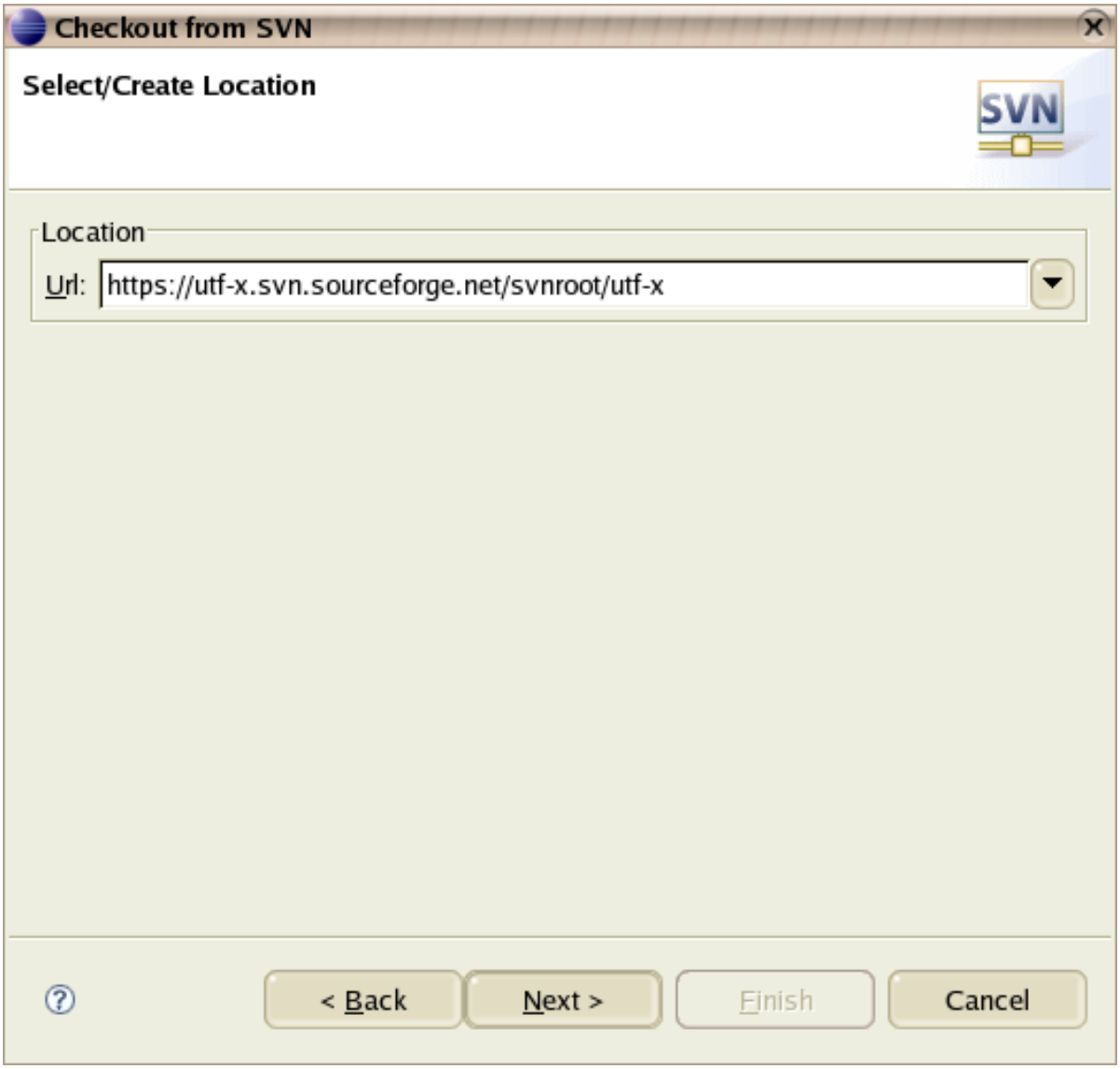

Figure 1.3: Create location

???

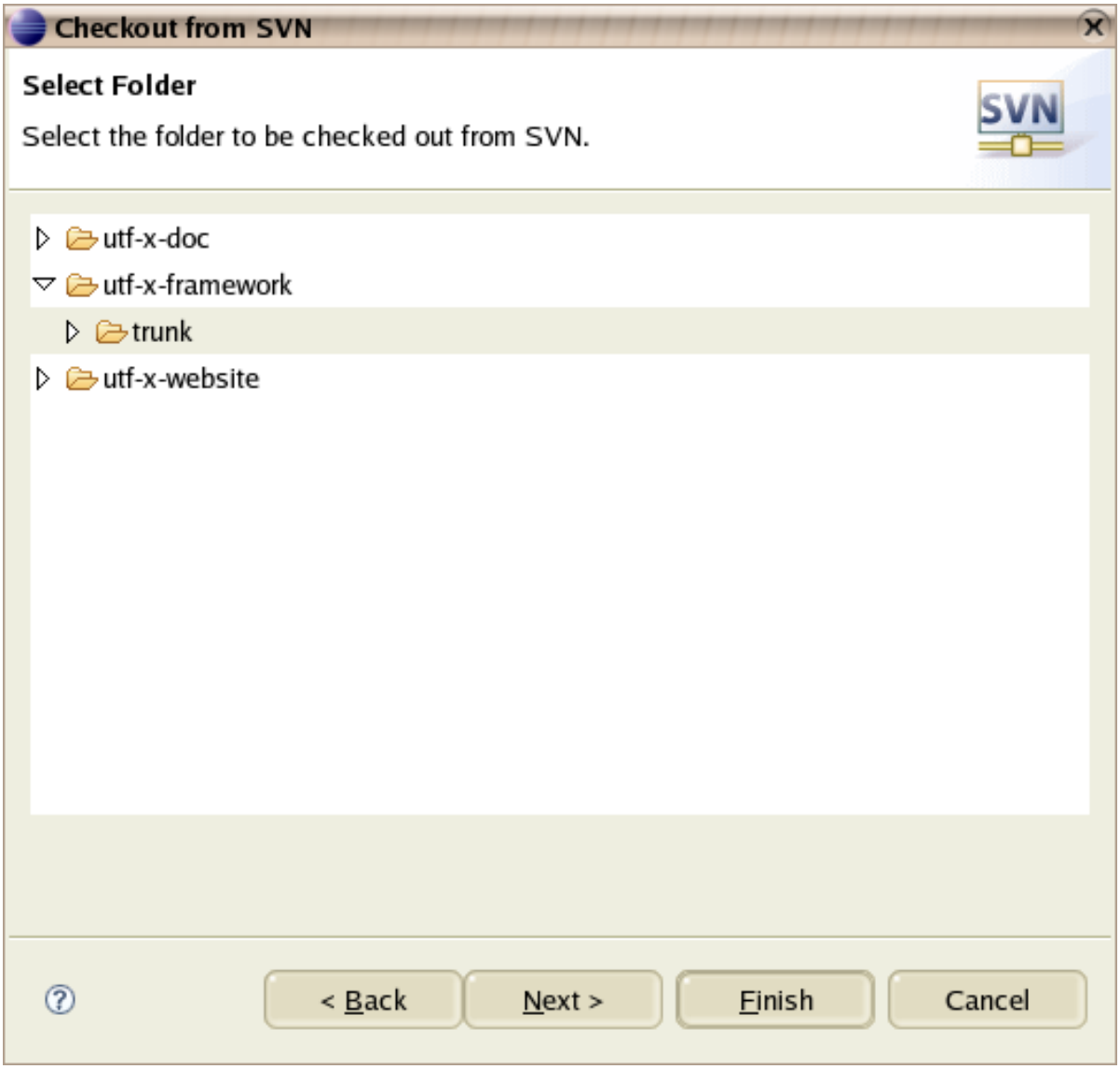

Figure 1.4: Select what you want to checkout

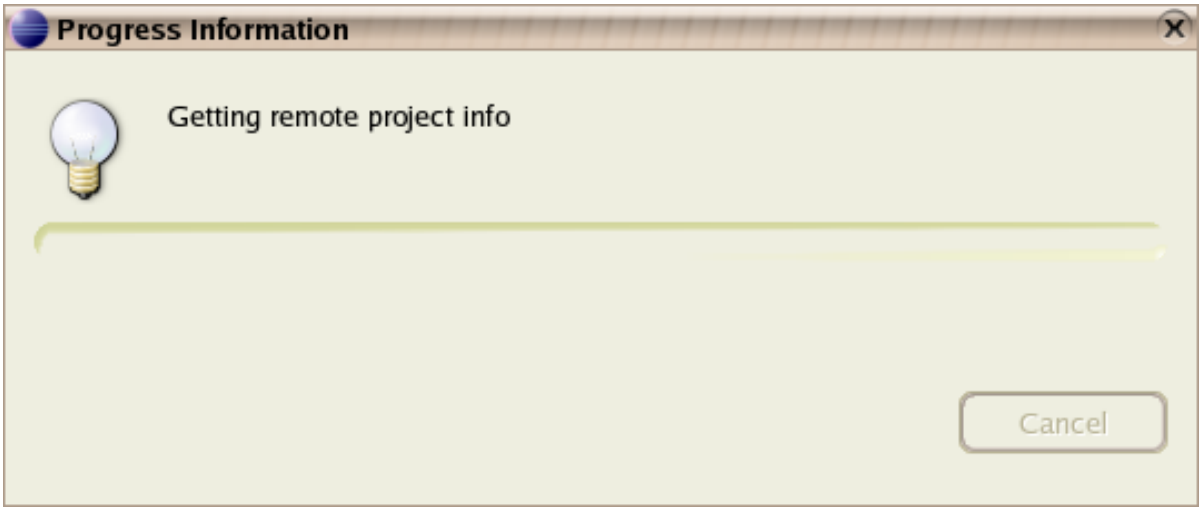

Figure 1.5:

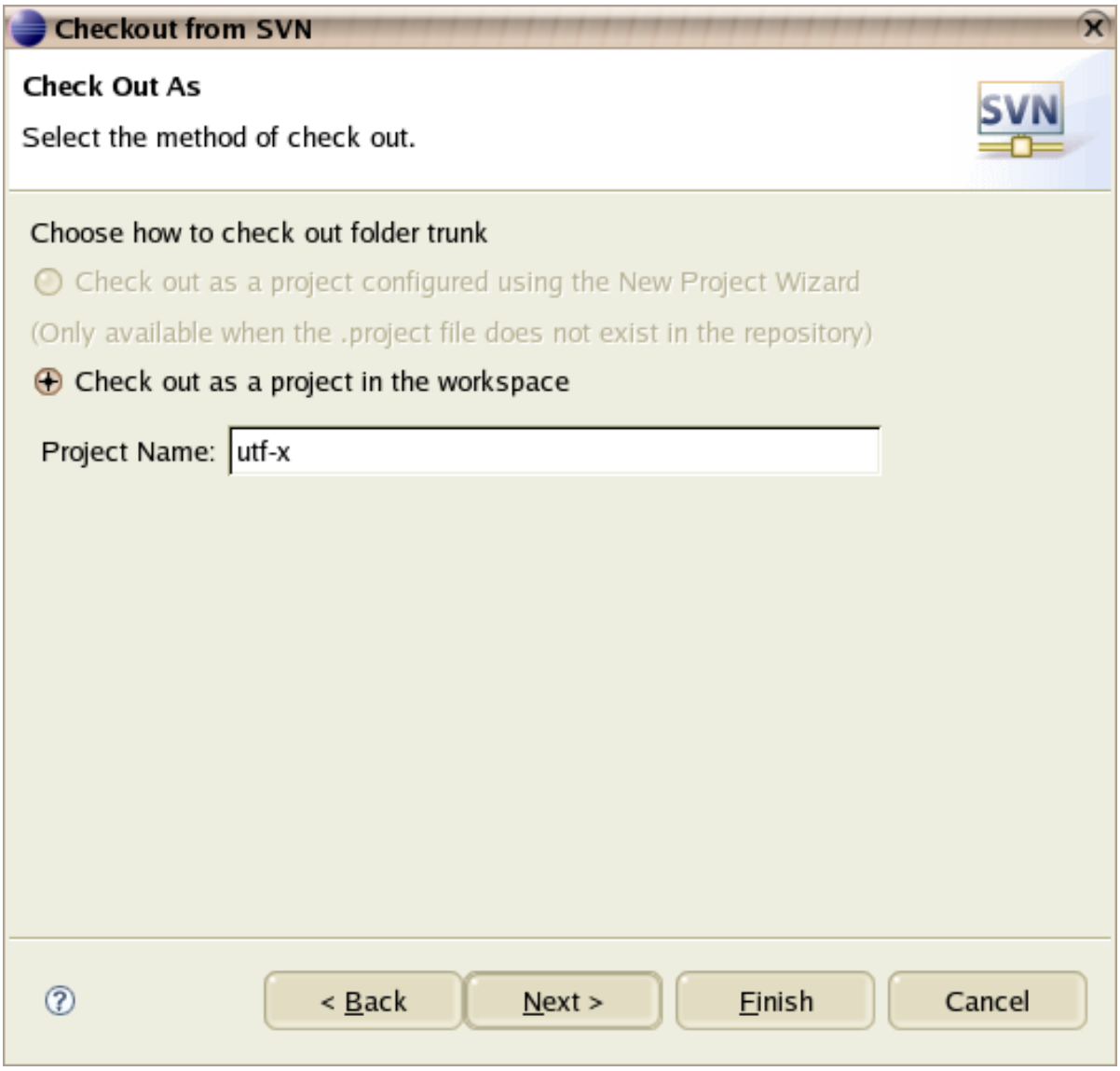

Figure 1.6:

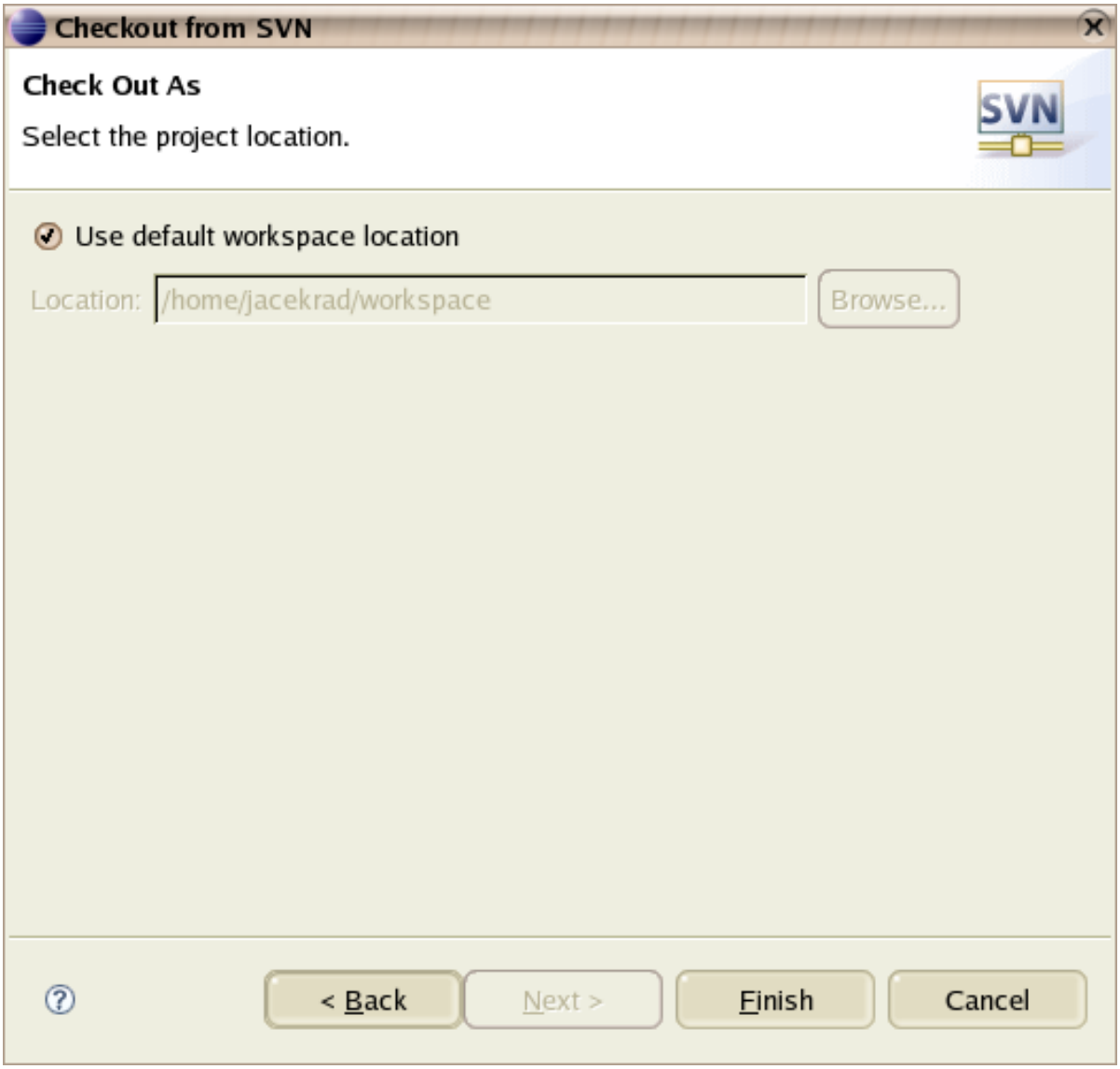
Checkout from SVN
Check Out As
Select the project location.
SVN
Use default workspace location
Location:
/home/jacekrad/workspace
Browse...
< Back
Next >
Finish
Cancel

Figure 1.7:

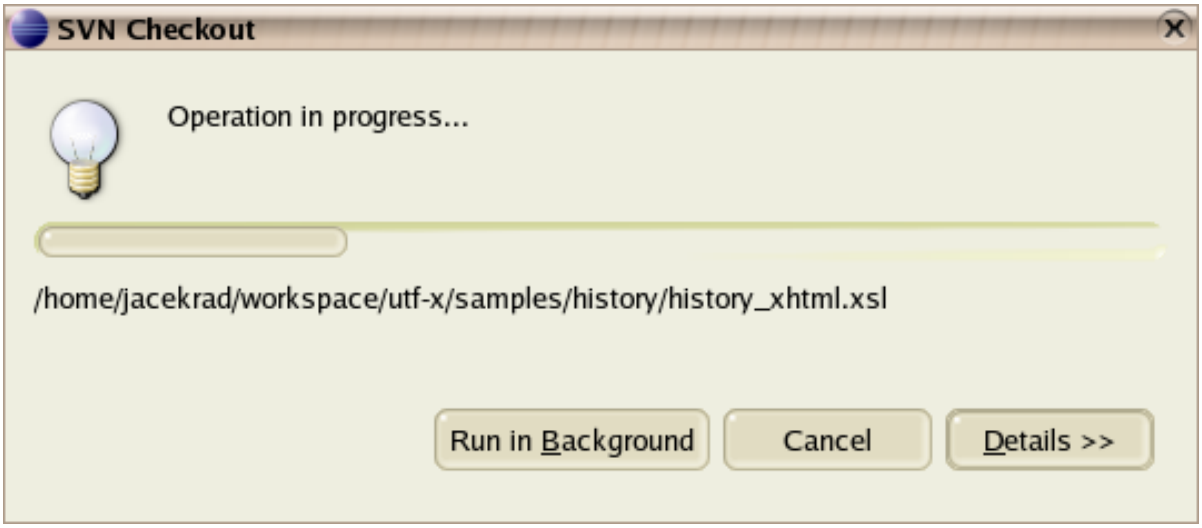
SVN Checkout
Operation in progress...
/home/jacekrad/workspace/utf-x/samples/history/history_xhtml.xsl
Run in Background
Cancel
Details >>

Figure 1.8:

1.2.2 Configuring XML Catalog

Project root directory contains an XML Catalog file `catalog.xml` file which is used by various UTF-X processes to resolve PUBLIC identifiers. You should tell the <oXygen/> XML editor about this catalog file so it knows where to find relevant DTDs.

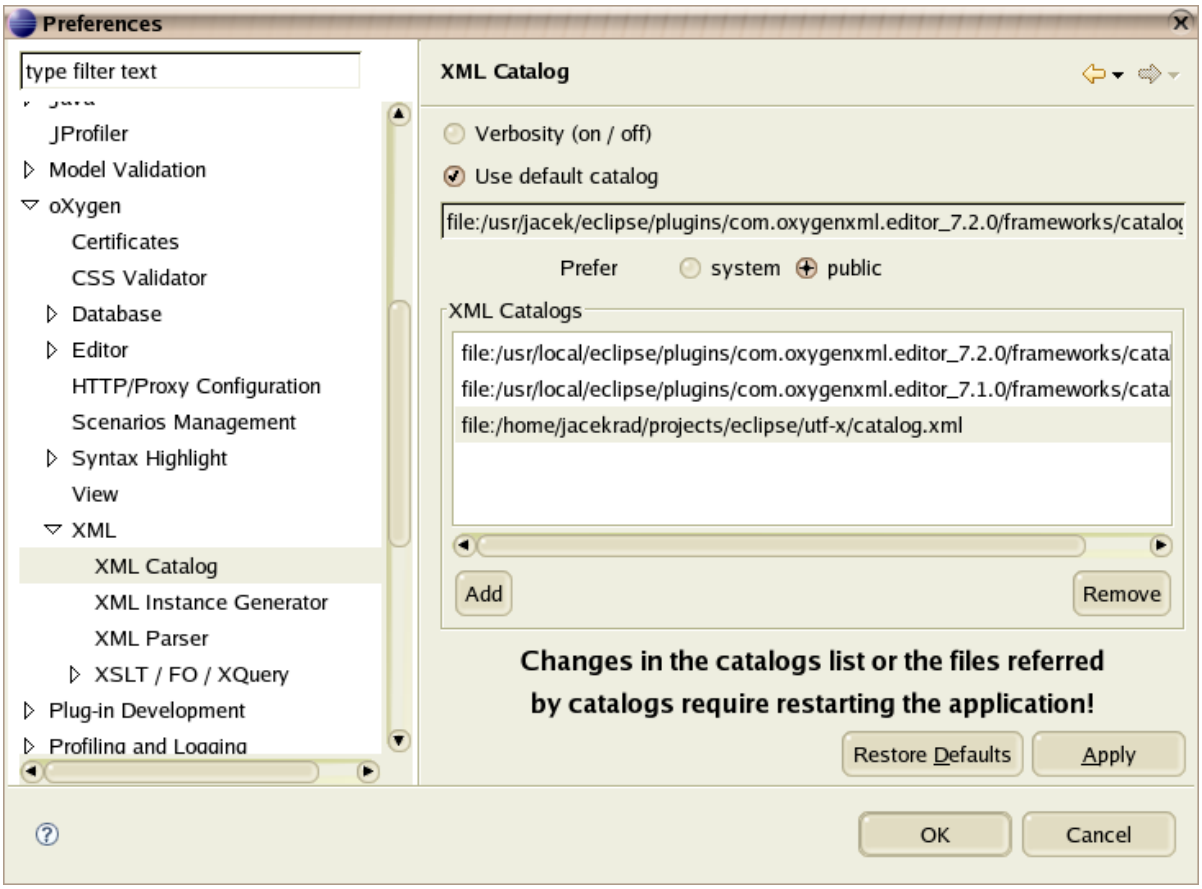

Figure 1.9: Adding UTF-X XML catalog to <oXygen/>'s catalogs.

2 Methodology

In principle the UTF-X project follows the eXtreme Programming (XP) development methodology. Even though some aspects of XP like pair programming and standup meetings are difficult (not impossible) to achieve when the team is spread across different continents, the aim is to adopt as many of the XP rules as possible.

Below is a list of currently adopted XP rules:

2.1 Release Planning

Since the project is again gaining momentum and more developers are actively involved, release planning will soon become a more formal approach.

2.2 Small Releases

Small, frequent releases ... define, how often, every 4 or 6 weeks?

2.3 Coding Standards

Coding standards ensure consistent quality results. This guide is all about standards.

2.4 Test-First-Design

Test-first-design particularly important in this project as the product we are developing is a unit testing framework which promotes test-first-design. Basic process for writing a new Java method and tests would be:

1. Write method signature.
2. Document what the method does including any exceptions thrown, pre and post conditions, all parameters and return value
3. Write one or more unit tests for the method making sure all functionality is tested.
4. Run the test(s). Should fail.
5. Implement the method until all tests pass.

Depending on the complexity of the method the above may be an iterative process. If the method is non-trivial you should create a test for one particular bit of functionality and then implement the functionality. Do the same for other bits of functionality until all bits of functionality are implemented and all tests pass.

If a bug is found in existing code then a unit test that fails due to this bug should be written first. Once the test is written you should fix the bug until the test passes. If you do this then this bug will never cause problem again.

2.5 Collective Ownership

Collective ownership means that we all own all parts of UTF-X. You may design and develop a module today but tomorrow someone else will be enhancing what you have done and vice versa. There is no "my code" or "your code", its the team's code!

2.6 Optimise Last

Design well and optimise last means that you should not design for performance, but concentrate on good design techniques and utilise well know OO design patterns. If performance is poor and needs to be improved then it can be done later, but in most cases it will be just fine.

2.7 Never add Functionality Early

Never add functionality early, just in case it may be needed at some point in the future.

2.8 KISS

The old "Keep It Simple Stupid" principle is very important. Design for what is needed now in a well, but as simple as possible manner. You will not impress anyone by demonstrating how complex code you can write. Note that shorter code does not always imply simpler code. Martin Fowler has a good refactoring example where the resulting code is longer and slower, but is much clearer and simpler.

2.9 Refactoring

Refactoring is a process whereby you improve the quality of the code without making changes to the functionality. See refactoring.com for the full refactoring catalog.

3 Code Documentation, Style and Formatting

3.1 Java

All Java source code MUST be written according to the Code Conventions for the Java Programming Language style. The easiest way to ensure that your code complies with these conventions is to use the Eclipse 3.1 built-in Java formatter.

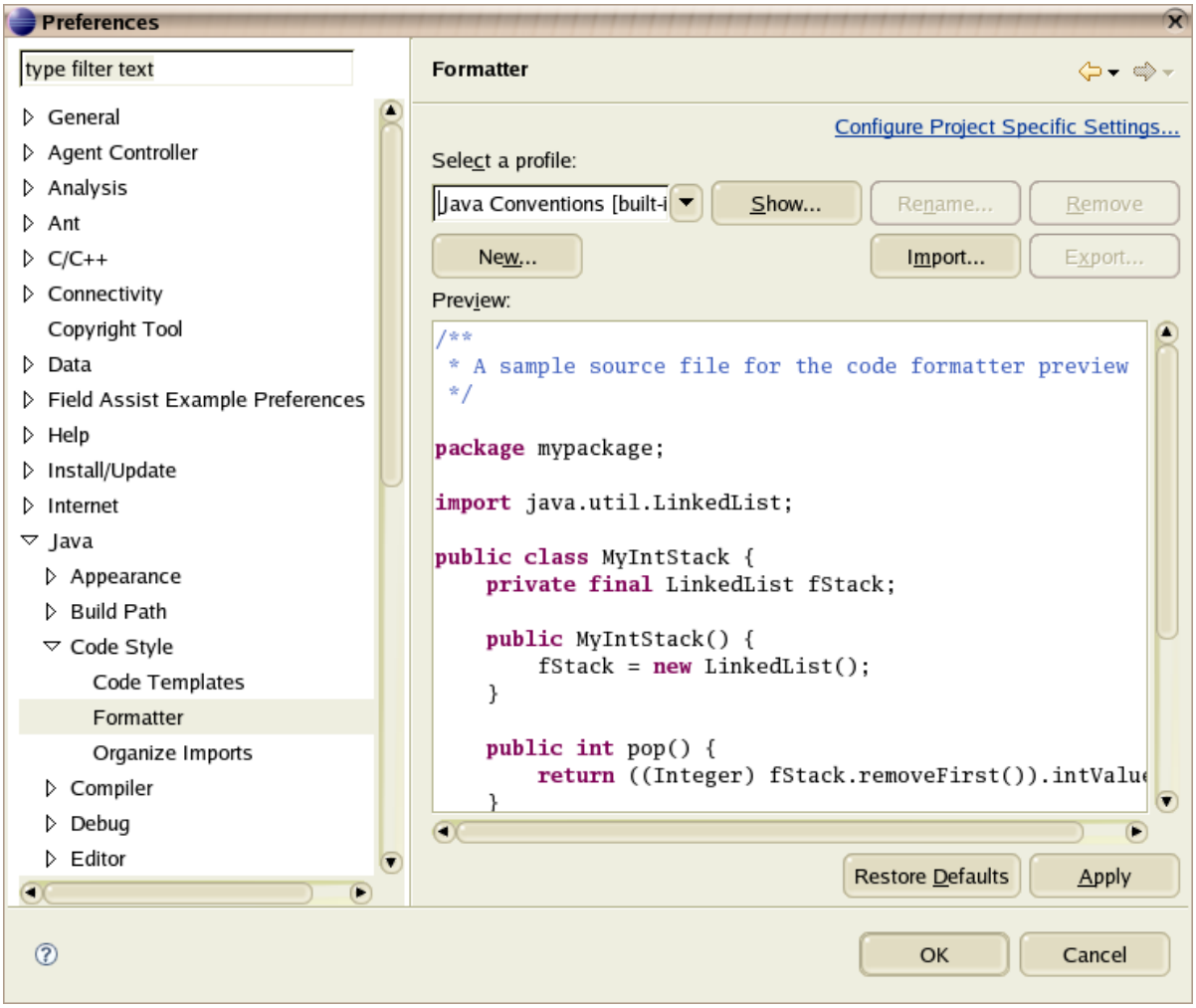

Figure 3.1: Setting up Java code formatter in Eclipse

4 Headers

4.1 Java Class

```
/**
 * One line description of the class.
 * More detailed (complete) description of the class.
 *
 * <p>
 * Copyright &copy; 2006 UTF-X developers.
 * </p>
 *
 * <p>
 * This program is free software; you can redistribute it and/or modify it under
 * the terms of the <a href="http://www.gnu.org/licenses/gpl.txt">GNU General
 * Public License v2 </a> as published by the Free Software Foundation.
 * </p>
 *
 * <p>
 * This program is distributed in the hope that it will be useful, but WITHOUT
 * ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS
 * FOR A PARTICULAR PURPOSE. See the GNU General Public License for more
 * details.
 * </p>
```

```
 *
 * <code>
 * $Source: /cvs/utf-x/framework/doc/devguide/devguide.xml,v $
 * </code>
 *
 * @author Jacek Radajewski
 * @author Other Author
 * @author Yet Another Author
 * @version $Revision: 210 $ $Date: 2006-12-11 19:25:54 +1000 (Mon, 11 Dec 2006) $ $Name:  $
 */
```

4.2 Java Method

Example of a typical Java method header with signature.

```
/**
 * Parses an XML document file and returns the total number of utfx:test elements.
 *
 * @param xmlFile the file that represents the XML document to be parsed and process
 *
 * @throws IllegalArgumentException if the file does not exist or any other IO error
 *         occurs
 * @throws SAXException if any error occurs during parsing of the file.
 *
 * @return the number of utfx:test elements found.
 */
 public int countUtfxElements(File xmlFile) throws IllegalArgumentException, SAXException {
     .
     .
     .
     return elementCount;
}
```

4.2.1 JavaDoc

JavaDoc documentation ....

4.3 XML and XSLT

```
<?xml version="1.0"?>
<!--
        ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        $Id: devguide.xml 210 2006-12-11 09:25:54Z jacekrad $
        ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

        Purpose: Detailed description ...

        ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

        Copyright(C) 2006 UTF-X Developers.

        This program is free software; you can redistribute it and/or modify
        it under the terms of the GNU General Public License v2 as published
        by the Free Software Foundation (http://www.gnu.org/licenses/gpl.txt)

        This program is distributed in the hope that it will be useful, but
        WITHOUT ANY WARRANTY; without even the implied warranty of
        MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
        General Public License for more details.

        ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
-->
```

5 UTF-X Roadmap

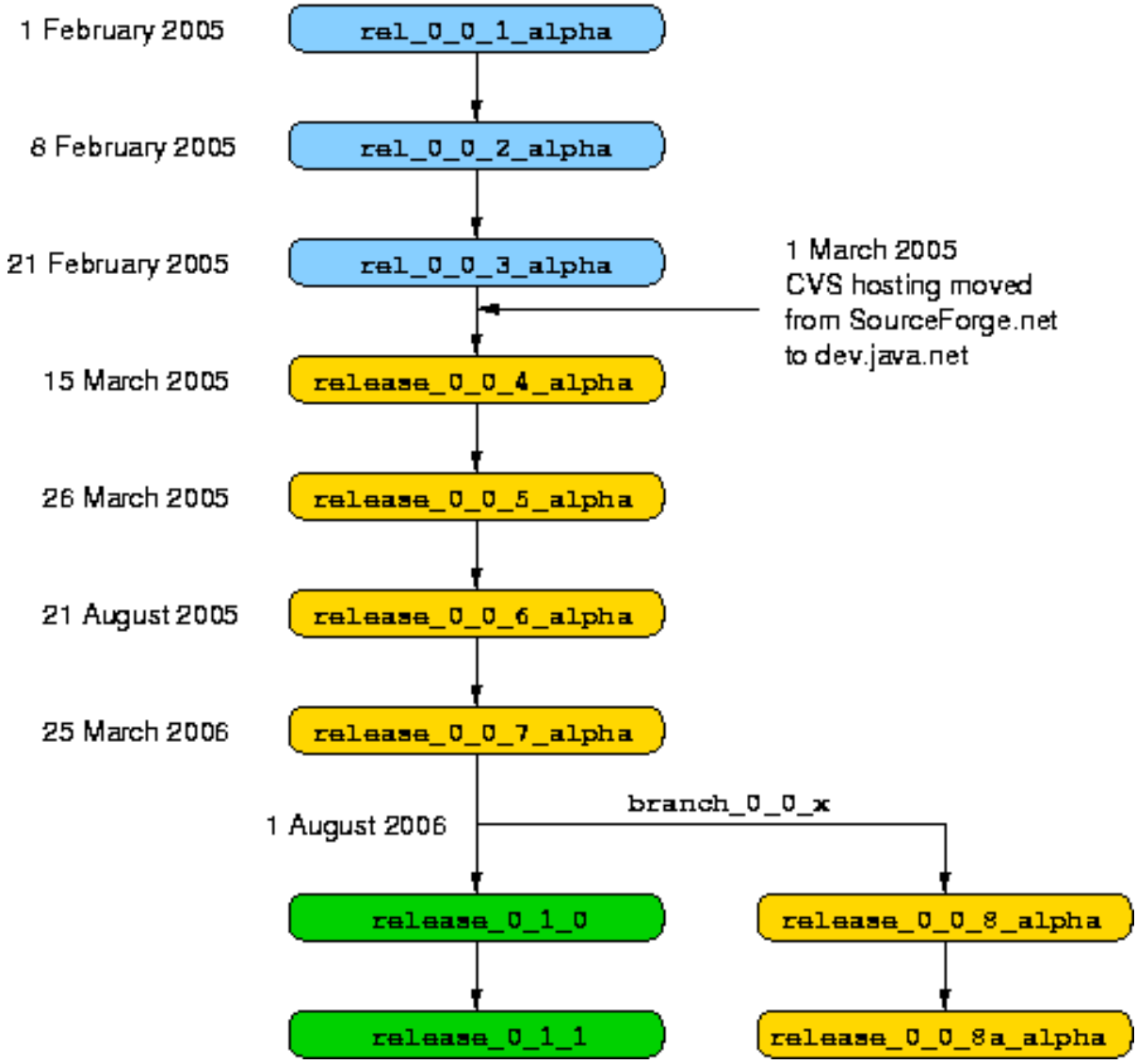

Figure 5.1: Releases and branches

6 Release Instructions

Ac copied from RELEASE_INSTRUCTIONS.txt

```
        How-to release utf-x
====================

* Decide when it is time to release.
* Tag the release with release_{version_number}.
  In Eclipse:
  - Select 'framework' directory.
  - Right-click -> Team -> Tag as Version
  - Please enter a version tag: release_0_0_6
* Checkout the entire codebase using the tag,
```

```
   and then proceed to go through with the following
   process before making the actual release:

   (need suggestions here).

 * Inform on the website and use other channels (www.cafeaulait.org).
```