Coding Standards

by Jon S. Stevens, Jason van Zyl

Table of contents

1. Coding Standards

This document describes a list of coding conventions that are required for code submissions to the project. By default, the coding conventions for most Open Source Projects should follow the existing coding conventions in the code that you are working on. For example, if the bracket is on the same line as the if statement, then you should write all your code to have that convention.

If you commit code that does not follow these conventions, you are responsible for also fixing your own code.

Below is a list of coding conventions that are specific to Turbine, everything else not specificially mentioned here should follow the official Sun Java Coding Conventions (http://java.sun.com/docs/codeconv/html/CodeConvTOC.doc.html) .

1. Brackets should begin and end on a new line and should exist even for one line statements. Examples:

```
if ( foo )
{
    // code here
}

try
{
    // code here
}
catch (Exception bar)
{
    // code here
}
finally
{
    // code here
}

while ( true )
{
    // code here
}
```

2. Though it's considered okay to include spaces inside parens, the preference is to not include them. Both of the following are okay:

```
if (foo)

or
```

```
        if ( foo )
```

3. 4 space indent. **NO tabs**. Period. We understand that many developers like to use tabs, but the fact of the matter is that in a distributed development environment where diffs are sent to the mailing lists by both developers and the version control system (which sends commit log messages), the use tabs makes it impossible to preserve legibility.

In Emacs-speak, this translates to the following command:

```
        (setq-default tab-width 4 indent-tabs-mode nil)
```

4. Unix linefeeds for all .java source code files. Other platform specific files should have the platform specific linefeeds.

5. JavaDoc **MUST** exist on all methods. If your code modifications use an existing class/method/variable which lacks JavaDoc, it is required that you add it. This will improve the project as a whole.

6. The Jakarta/Turbine License **MUST** be placed at the top of each and every file.

7. If you contribute to a file (code or documentation), add yourself to the authors list at the top of the file. For java files the preferred Javadoc format is:

```
        @author <a href="mailto:user@domain.com">John Doe</a>
```

8. All .java files should have a @version tag like the one below.

```
        @version $Id: code-standards.xml,v 1.1 2004/06/20 09:12:35 tomdz
Exp $
```

9. Import statements must be fully qualified for clarity.

```
        import java.util.ArrayList;
        import java.util.Hashtable;

        import org.apache.foo.Bar;
        import org.apache.bar.Foo;
```

And not

```
        import java.util.*;
        import org.apache.foo.*;
        import org.apache.bar.*;
```

X/Emacs users might appreciate this in their .emacs file.

```
        (defun apache-jakarta-mode ()
          "The Java mode specialization for Apache Jakarta projects."
          (if (not (assoc "apache-jakarta" c-style-alist))
              ;; Define the Apache Jakarta cc-mode style.
              (c-add-style "apache-jakarta" '("java" (indent-tabs-mode .
nil))))

          (c-set-style "apache-jakarta")
          (c-set-offset 'substatement-open 0 nil)
          (setq mode-name "Apache Jakarta")

          ;; Turn on syntax highlighting when X is running.
          (if (boundp 'window-system)
              (progn (setq font-lock-support-mode 'lazy-lock-mode)
                     (font-lock-mode t))))

        ;; Activate Jakarta mode.
        (if (fboundp 'jde-mode)
            (add-hook 'jde-mode-hook 'apache-jakarta-mode)
          (add-hook 'java-mode-hook 'apache-jakarta-mode))
```

Thanks for your cooperation.