



Capstone Courseware, LLC

33 Boylston Street
Jamaica Plain, MA 02130

877-227-2477
capstonecourseware.com

The JSP Standard Tag Library

Will Provost

Instructor's Guide

Revision 1.1



Module Overview and Philosophy

This module is designed for two audiences:

- Experienced JSP 1.2 authors who want to learn JSP 2.0 features and JSTL
- New JSP authors who have recently learned JSP basics.

Clearly, the first audience will move more quickly through the material, but either should do well in the standard two-day timeline. We've developed the lab structure with a few options and it should be flexible enough to keep things moving well in either case.

The module is a fairly straightforward walk through the JSTL libraries. We start with an overview chapter to introduce (or review) the JSP 2.0 expression language and to introduce the JSTL minimally. This chapter also discusses the evolving JSP 2.0 authoring style, beginning with the page "Going Scriptless." Much of this section is best-practice discussion, somewhat in advance of learning the specific techniques, but it seemed the best place for it, to put the remaining chapters in some architectural context.

The remaining chapters each introduce one of the four JST libraries. Every action in every library is documented and discussed, using a standard notation at the top of each such page to simplify the learning process and to make the book easier to use as a reference. The chapters are more than just expanded documentation, though: at various points in each chapter, common techniques and best practice relevant to some group of actions is discussed, and these techniques and practices are amplified in the labs.

The module wraps up with a significant workshop exercise, which departs from the other labs in the module in that it does not walk students through the solution step by step. Students are expected at this point to have developed the skills to attack the problem on their own, perhaps with some instructor assistance. If a client prefers it, this workshop is also designed to work well as a "final exam" for the module to show skills development.





Timeline

Day 1

Chapter 1	JSTL Overview
Chapter 2	The Core Actions
Chapter 3 (spans days)	The Formatting and i18n Actions

Day 2

Chapter 4	The SQL Actions
Chapter 5	The XML Actions





Eclipse Overlays

Capstone Courseware provides an optional package of workspace and project files for Eclipse WTP 1.5 for this course. (See the course Setup Guide for download URLs.) Instructors, use this package on your own initiative and at your own risk. You should have experience yourself with Eclipse before using the overlay package in the classroom. The workspace and projects have been tested lightly with the course but are not part of the standard product.

That said, this overlay should save a good deal of work for those who wish to use Eclipse instead of the text editor and command-line tools that are standard for the course.

Since the lab software is configured as a single web application that encompasses all examples, demos, and labs, to get this one big webapp into Eclipse required that we reconfigure the lab image itself. Specifically, when the installer for the Eclipse overlay is run, it:

- Copies the **c:\Capstone\JSTL** tree to **c:\Capstone\JSPwithEclipse\Application\JSTL**.
- Unzips the Eclipse workspace itself to **c:\Capstone\JSTLwithEclipse\Workspace**.
- Updates **c:\Capstone\JSTLwithEclipse\Application\JSTLWEB-INF\JSTL.xml** to reflect the new location – just in case you want to run a standalone Tomcat server and install the labs as described in the coursebook, alongside the managed Tomcat server that will run from within Eclipse.

Open Eclipse WTP 1.5 and open **c:\Capstone\JSTLwithEclipse\Workspace**. The workspace is pretty simple: it has a single “Dynamic Web” project for that one big web application, which is pre-deployed to a managed Tomcat 5.0 server. See the read-me file that shows when you first open the workspace for startup and testing instructions.

For the **update** scripts in the LovelsBlind and Replication case study to work, it's essential that **CC_MODULE** be set to **c:\Capstone\JSTLwithEclipse\Application\JSTL** prior to starting Eclipse. Also, because Eclipse WTP 1.5 deploys to Tomcat in a different way than we do in our non-Eclipse approach, it will be necessary to rebuild the **JSTL** application after running **update**, so that Tomcat notices the new **.tag** files.

There is no decent way to configure data sources for Tomcat 5 contexts from within Eclipse; so the examples that use data sources – **Store** and **Replication** – will have to be deployed to a standalone Tomcat instance outside Eclipse – simply follow the instructions in the coursebook as if Eclipse were not installed.

You should advise students to use **c:\Capstone\JSTLwithEclipse\Application\JSTL** as the root whenever they edit files outside of Eclipse. Changes made under **c:\Capstone\JSTL** will not be picked up by Eclipse.



Also, note that the server configurations are the same as in the Eclipse workspaces for our servlets and JSP courses. If you're teaching more than one of these modules, and don't want to have to swap workspaces, you can import the JSP and/or JSTL projects into the servlets workspace, and test them from there. Be sure to build the project before trying to publish and test it on the server! This is done in advance in the JSP and JSTL workspaces, but if you forget this step (a) it won't work, and (b) it really will be a huge pain to clean up.

Again, Capstone Courseware can only offer complete technical support on the standard course, and while we hope this overlay is convenient, it is not as thoroughly tested as the core lab image at this time. If a given exercise is giving trouble, please be certain to build and run it from the command line, using the SDK tools as prescribed in the student guide, before contacting Capstone.





Chapter 1. JSTL Overview

This chapter introduces the JSTL and then immediately steps back (or to the side) to introduce and to explain the EL, since so much of JSTL use relies on EL. EL is demonstrated in direct use in template content, which works thanks to Tomcat 5's support for JSP 2.0. (Most EL in the remainder of the module will be in JSTL actions.) Then the JSTL is considered a bit further, but detailed study of particular actions is deferred for the remaining chapters. Instead, focus on the architectural level and the concepts of decomposition in page authoring and organization. Many JSP 1.2 developers will be familiar with this approach, and will have some experience with custom tags as well; in this case it's simple enough to gloss over this section or to skip it entirely.

Chapter 2. The Core Actions

Now students get down to brass tacks, learning the core library in a somewhat lengthy chapter that gives plenty of exercise in procedural coding. Do make a point of hammering on the difference between static and dynamic attributes: the basic distinction is obvious enough, but, as the book points out over a couple specific cases, actual usage can be far from intuitive. Explain exactly why the **target** attribute of `<c:set>` can't be primed with a literal string, so students have an idea of where EL evaluation actually occurs and how it can and can't be leveraged.

Note that it is not until partway through this chapter that we do the usual environment setup pages. These are placed here because they are not strictly needed earlier: none of the code examples in Chapter 1 calls for testing. If you think students would be interested in seeing example code running right off the bat, you may want to jump ahead to these pages and to get students' environments set up early.

Chapter 3. The Formatting and i18n Actions

This chapter breaks into two sections: one on formatting dates, times, and numbers, and one on multiple language support. Students may or may not have experience with i18n problems, so it's possible that some additional discussion of locales, time zones and resources will be appropriate. Having the J2SE SDK documentation on hand in a browser may be useful for reference throughout the chapter.

Note that the book calls for browsers to be configured to establish preferences for one language over another to support the labs. Rather than include explicit instructions for a given browser in the student guide, we ask the instructor to be familiar with this process for the browser of choice in a given classroom, and to walk students through it.





Chapter 4. The SQL Actions

This chapter addresses use of relational data using SQL and JDBC data sources. This chapter is the shortest in the module, partly because (as the book points out) this practice is generally discouraged in JSPs for distributed systems, and partly because there just aren't that many actions to discuss. The SQL library is a very simple wrapper around SQL connection and syntax; between container management of connections and resource references and the literal encoding of SQL text in query and update actions, there's not much to discuss that doesn't wander too far down the road into teaching SQL or JDBC itself. If students do have a great interest in this library, you might want to discuss the use of result collections a bit further, and possibly to plan on an expansion of Lab 5C, which combines SQL and XML skills in a replication exercise. (Perhaps encourage students to write the replication code in both directions.)

Chapter 5. The XML Actions

This final chapter covers XML actions, which brings both XPath and XSLT into play. Students may or may not know XPath or XSLT. The use of each technology is intentionally limited so as not to assume too much about student background in these areas. Even so, there is likely to be a need for expanded discussion of XML, XPath and XSLT to audiences without good experience in them.

This chapter is the longest of the bunch, not in page count but in suggested time. XML is of course an increasingly important language and technology area for JSP authors, and it makes sense to dwell on these actions and techniques, more so than for the SQL actions, for instance. Also, the switch to XPath from EL in the various attribute values will take some getting used to, as will working with XSLT.

The final lab is optional, and can serve as a safety valve of sorts if the class is running ahead or behind schedule: it can be skipped if you're out of time, and it can also be allowed to run long, perhaps even to 2 hours, if there is extra time and students are interested in delving further into techniques. Possible expansions of the lab (one of which is mentioned under Chapter 4 notes) are to add transaction control to the replication process at various levels and to implement bi-directional replication.





Revision Notes

Revision 1.1 updates the course to use the JSTL 1.1 reference implementation, and also upgrades to current standards for our courseware. Major changes:

- With the change to JSTL we can use the current and final taglib URIs, ending with `/jsp/jstl/???`. No more dancing around with “normal” and “-rt” versions that were common with JSTL 1.0, and we assume a JSP 2 container for everything.
- Supporting Java classes have been pre-compiled to the `WEB-INF/classes` directory, so there's no need anymore to run **compile** scripts per example. (The **update** script for the `LoveIsBlind/Replication` case study is still necessary, to update custom **.tag** files to their latest versions before testing a given step.)
- Tomcat 5.0 is now bundled with the lab software, simplifying the setup process.
- Database setup is much simpler now thanks to a bundled MySQL 5.0 installation.
- There is now an Eclipse WTP 1.5 workspace for the course. See the section on this earlier in the document for details on what this workspace will (and won't) do.

Revision 1.0.1 updates the course to the stable release of Tomcat 5, and also includes a number of minor changes:

- The logistics of using custom tag files in the `Love is Blind` example track have been simplified quite a bit. Instead of compiling, running a separate script to update the tag files (which didn't always work), and then restarting Tomcat, in most cases it is enough to run a single update script and to let the server pick up changes and refresh.
- A bug in the internationalization of `Love is Blind` has been fixed: the `Results.jsp` was not publishing the resource bundle to request scope, and so the `members.tag` wasn't loading resources correctly.
- The `Calculator JSP` has been rebuilt with more correct HTML code for its layout.
- A new example `ElectronicDJ\Scriptlet` has been added, to remove a dependency on a code path that was actually in the JSP module. This example now serves the description in Chapter 1 about sharing objects between JSP, JSTL and other code.
- Added a page on JSP2 EL functions to Chapter 1.
- Fixed the screenshot for the `DateTime` example in Chapter 3, which was mistakenly caught while browsing with the language set to French.
- All examples have been tested on Netscape 7.1 and IE 6.

Revision 1.0 is the initial release.





Errata

Following are issues that have come to light since the most recent release of the course. These will be fixed in the next revision:

- The **Occurrence** example attempts to remove short words from the map of all words found in the document, by using `<c:set>` with no **value** attribute. This is not exactly correct, as this will just set the frequency of that short word to `""`. To really remove the map entry, we should say `value="{null}"`. (The error is benign; due to the nature of the processing we then apply to the map, a word with a zero frequency is just about the same as the word not being in the map at all.)

Troubleshooting and Tool Tips

First, be sure to be familiar with the lab and environment setup instructions in the Table of Contents. Assure that the instructions in the module Setup Guide have been followed, and that you know the locations of the installed tools. If, having followed those steps, a classroom machine is failing to run demos or lab answers correctly, here is a list of items to consider and to doublecheck:

- Generally, Tomcat is good at reloading and recompiling JSPs after they're updated, but if a student runs into trouble with JSP or Java class development and everything seems like it should be working but isn't, try shutting down and restarting Tomcat.
- The **update.bat** script for the Love is Blind and Replication tracks simplifies build and test cycles, and works reliably in all cases but one: when moving from LevelsBlind/Step5 to Step6, or back the other way. This has to do with the addition of a new **pictureURL** attribute to the custom tags, and Tomcat's willingness to let go of loaded tag handlers. Even in this case, we've found the update script will persuade Tomcat to reload in most cases, resulting in successful testing of updated code. When it doesn't, simply stop and restart Tomcat, and run the update script again.





Feedback

We truly do welcome feedback, both of a specific nature (pointing out mistakes) and general suggestions. For the former sending email with a numbered list of corrections would be most helpful.

Please send feedback to:

Will Provost
Capstone Courseware
<mailto:provost@capstonecourseware.com>
www.capstonecourseware.com

