



Capstone Courseware, LLC

33 Boylston Street
Jamaica Plain, MA 02130

877-227-2477
capstonecourseware.com

JSTL with WebLogic

Will Provost

Instructor's Guide

Revision 10.3.0



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





Workshop Overlay

Capstone Courseware provides an optional package of workspace and project files for WebLogic Workshop 10.3 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 Workshop 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.

And, the “win” in using Workshop for a bunch of JSP editing is probably mostly in the built-in JSP editor, which could be used without actually deploying the web application to a managed server within Workshop. The mode we recommend for using this material is to install and open the provided workspace, but then to use this only as a means to find, open, edit, and save JSP files, while managing a WebLogic server externally, from the command line, as described in the coursebook. This should give the best blend of editing facility and ease of testing, using an external browser.

But it is also possible, as with our other overlays, to perform nearly all the course tasks within the Workshop environment: start/stop a server, publish, test, debug, etc. The workspace has a built-in WebLogic server profile that assumes a domain exists at **c:\Capstone\JSTL_WebLogic\WLDomain**. So, the best way to get started is to set environment and create the domain as described in the coursebook; then start Workshop and load the workspace; then start the server from within Workshop, and go from there.

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_WebLogic/WLDomain**, and test them from there.

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. (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.

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 10.3.0 is the initial release of this course, based on Course 113, version 1.1.

Errata

There are no errata for the course at this time.

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:

- In the latter part of Lab 4, when adding a new column to the **Album** table, students may run afoul of a tendency for the `<sql:query>` tag (or some lower layer on which it relies in the WebLogic environment, more likely) to compile and cache queries. The effect is that, once the page has been tested with "SELECT * FROM ALBUM", the new column will not be fetched, even though it's there. There won't be an error; the LastSpins column in the JSP will just show "Never" even when there's a real value in that column. Warn students about this and suggest that, once they've gotten to the point in this lab at which they should be seeing actual last-played dates and times, but are still seeing "Never", to modify the SQL query in the JSP in some trivial way: list out all the required columns instead of using *, or add " where 1=1" to the end. Any such change will cause the query to be compiled fresh against the new schema, and the data from the new column will come through.





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

