



Capstone Courseware, LLC

33 Boylston Street
Jamaica Plain, MA 02130

877-227-2477
capstonecourseware.com

XSLT

Will Provost

Instructor's Guide

Revision 1.2



Revision Notes

Revision 1.2 significantly expands and reorganizes the course content, taking the course from an overly challenging 2-day timeline to a 4-day one that should be comfortable, but not excessive, for most audiences.

- There is a new introductory chapter to set the stage for Chapters 2 and 3 on XPath and XSLT template-matching rules.
- Though Chapter 3 still focuses on a rigorous approach to the built-in template rules, a new primary lab is simpler and more concrete than The Matrix, which is retained as an optional second lab.
- There is more hands-on work in callable templates.
- Chapter 7 is now “Advanced XSLT” and includes new material on performance and debugging while shedding some extra weight.
- There is a new final chapter on XSLT extensions.
- Much more hands-on exercise overall. This topic has proven to be unusual in the balance of lecture to lab that seems to be optimal: even fairly experienced students seem to benefit from a lot of time for labs, and from a lot of labs, just practice, practice, practice.
- The course now has a dedicated GUI “console,” a sort of mini-IDE for XPath and XSLT, that makes lab work much simpler to carry out.





Revision 1.1 is a restructuring of the modules on XPath and XSLT. Topics have been reshuffled a bit, with the aim of improving flow and of giving more emphasis to introductory topics and more hands-on exercise in basic XSLT transformation. Intro XML module has been deleted.

- XPath introduction has been expanded with more description of its use and context, more motivation.
- Added a few diagrams in the very early part of the module to clarify discussions.
- Module 2, Chapter 2 has been rearranged to simplify the flow of topics: where output method and whitespace-handling techniques were interspersed with discussion of templates and rules, they now follow the main “ABCs of XML” demo.
- Module 2, Chapter 3 now has fewer topics and more labs on those topics. Variables, parameters, and callable templates have been removed to module 503, letting this chapter focus on basic output control: values, copies, loops, and conditionals.
- A new first chapter for Module 3 covers variables, parameters, callable templates, and template management using priority and modes. This is mostly extracted from the previous version of module 502, chapter 3, which seemed to cover too much ground too quickly.
- What is now Module 3, Chapter 3 has become a broader exploration of various techniques for polishing the XSLT output: numbering, formatting, string parsing, calculating new values, etc.
- Many of the more esoteric topics have been removed from the later chapters. Tail recursion is still covered in Chapter 3, with an optional lab, but most of the other more adventurous topics are gone, being more properly the domain of application code (which may or may not leverage XSLT). Chapter 4 is slimmed down considerably as a result, focusing on multiple source documents and import/include of transforms.
- A new case study on transforming a rental-housing database from XML document to HTML presentation now provides the four labs for Chapter 3.
- Use of XML with CSS stylesheets has been de-emphasized, with more focus on XML-to-HTML transformation.

Revision 1.0.4 adds some introductory discussion and another example to chapter 2.

Revision 1.0 is the initial public release.





Course Overview and Philosophy

Over several iterations this course has evolved into quite a deep treatment of XPath and XSLT. It has been my experience that these topics – XSLT especially – demand a fairly rigorous treatment, certainly much more so than conventional wisdom would suggest; that the only two modes that really work are a very brief survey/cookbook treatment or a multi-day course that gets into the gory details of axis/node-test/predicate and built-in template rules and the whole odd set of inventions that is XSLT. It's a language that's terribly easy to get lost in, to think one has understood it by example, and then find that nothing is as it had seemed. Even experienced programmers – or maybe especially them – find it confusing, maybe because it invites so many comparisons to familiar languages and then rejects them.

Hence this newest version of the course, which is designed to (a) hammer home the above points and (b) insist that students take it slowly and get familiar with each concept and technique before building up. The challenge in this sort of presentation, obviously, is that students may be impatient and want more of a quick-start approach where they're building complex examples in the first day. The early chapters are a bit dry, especially Chapter 3 on the built-in rules. I have found that there is a great payoff in the latter half of the class, where in earlier versions of the course students would hit variables, callable templates, and especially grouping, and would really hit the wall, and now they can attack the more advanced material with confidence. But it does take a bit of motivation to get them there!

More specific discussions of module and chapter contents, and suggestions for effective teaching, appear on the pages following the timeline below.





Timeline

Day 1

Chapter 1	Getting Started with XSLT
Chapter 2	XPath
Chapter 3	Templates and Production (start)

Day 2

Chapter 3	Templates and Production (finish)
Chapter 4	Dynamic Content and Flow Control
Chapter 5	Variables and Template Management (start)

Day 3

Chapter 5	Variables and Template Management (finish)
Chapter 6	Sorting and Grouping
Chapter 7	Advanced XSLT

Day 4

Chapter 8	Multiple Documents and Transforms
Chapter 9	Extensions

Very well-qualified students may consider a 3-day timeline; skills in hand-authoring XML must be excellent, and usually for this to succeed students must be experienced software developers.





Chapter 1 – Getting Started with XSLT

This is a quick introduction to XSLT, showing how it's used and giving examples of the three output methods. It helps to set the stage for Chapter 2, which goes pretty deep on XPath before the students see any more XSLT. It's also an opportunity for students to get familiar with the primary tool for the course, the XPath/XSLT Console application. But mostly it's about giving everyone a sense for the role of XSLT: source + transform = result and how source vs. transform information is managed in separate documents.

Chapter 2 – XPath

This chapter focuses solely on XPath. This is essential for XSLT, of course, and fluency in XPath is helpful in several other key XML technologies. We've tried to develop the expression grammar incrementally, although this is not entirely possible – even the XPath recommendation contains a great deal of forward-reference, and by one author's estimation the 1.0 grammar is fundamentally broken. We try to make a virtue of the, um, organic nature of the language, and emphasize the simplicity and convenience that are at least possible for most purposes, while acknowledging that some XPath constructions are downright ugly but may be necessary for certain conditions. Try to keep to a theme of XPath as a way of expressing a question, the way the first few pages do and the first lab does. This is the best and simplest motivation for the language until more practical uses are seen in later chapters on XSLT.

Chapter 3 – Templates and Production

This chapter digs into the basic mechanisms of XSLT: built-in template rules, template matching, whitespace control, etc. All of these might be ignored in simple examples – and as discussed earlier this is the chapter that will most try students' patience – but without understanding these things, the transform author cannot keep control of the output for very long, and this in turn can lead to a sense that XSLT processing, or the language itself, are somehow imprecise. Try to clarify for students that once they gain control of production, the next chapter will introduce plenty of more interesting source data and transformation problems.

Chapter 4 – Dynamic Content and Flow Control

Now we look at source-data processing: how to get data for production, and how to control the order of output, which for programmers boils down to flow control. Now the exercises get a good deal more interesting, because we're dealing more naturally with source data. Once the ability to find and reproduce values and element trees is understood, the chapter turns to "flow control" constructs, including conditionals and loops. The more experienced programmers in a classroom may be tempted to see this as the real language, so to speak, and to assume that general-purpose algorithmic programming can be effected in XSLT. This is of course not the case, and a discussion to this effect might be helpful, before anyone gets their hopes up!



Chapter 5 – Variables and Template Management

This chapter introduces XSLT variables and parameters, and focuses on a few common uses thereof: capturing context nodes with variables for use in XPath expressions; applying templates with parameters; and calling templates with parameters. Students will also get more exercise in use of variables and callable templates in later chapters. Certainly the trickiest concept here is that “variables aren’t.” But students shouldn’t find the exercises in this chapter too daunting; the previous chapter tends to be the “hump” and typically students are in the groove at this stage.

Chapter 6 – Sorting and Grouping

Here we take a look at transformation through a SQL-tinted glasses, if you will. Many transformation designs face the same problems as RDB clients do: how to filter, sort, and group information. Sorting is simple, of course, but grouping in XSLT is more work and also conceptually more challenging. It usually takes a few passes for the idea to sink in that selecting one distinct value for each desired group is as difficult as it is, and then another pass or two to understand how to do it. Both preceding-sibling and key approaches are covered, more or less equally, but keys should certainly be promoted as best practice for the general case.

Chapter 7 – Advanced XSLT

This chapter is a bit of a hodgepodge, but certainly there’s some good, useful stuff here. Students will probably find the performance and debugging information especially interesting – especially by comparison to tail recursion! which only the real die-hards will likely enjoy. Though tail recursion is a powerful, if labor-intensive, technique, the lab in this topic should only be undertaken if there is ample time and explicit student interest.

Chapter 8 – Multiple Documents and Transforms

This short chapter covers the XSLT document function and the ability to import and include external transforms.

These techniques will probably lead students to broader questions about the role of XSLT in applications – when are transformations invoked, by what components, how are results transmitted and received, etc. This leads the question of how to integrate XSLT with programming languages and platforms, which might be discussed briefly as a segue to the next and final chapter on extensions.





Chapter 9 – Extensions

This chapter is both a natural conclusion and a jumping-off point. On one hand, many “missing” features of the language can now be addressed, such as converting RTFs to node sets for chaining transformations, and redirecting output to multiple streams. On the other, the use of extensions carries us out of the space of standard XPath and XSLT, to EXSLT, specific processors, specific languages, and ultimately toward the idea of custom extensions. The chapter’s exercises can only go so far without getting much more specific in tool and platform requirements, so this chapter sticks with some fairly simple uses of very common extensions. Depending on student interest, the discussion could go quite a ways beyond the end of the coursebook, and certainly there are good opportunities here for customized demonstrations and exercises based in a specific processor and with the appropriate development tools on hand.





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 course 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:

- The XPath/XSLT Console must be run from the appropriate working directory for a given exercise. It is not workable, for instance, to set up a desktop shortcut and then open from there every time one wants to use it. The recommended use is to have a DOS box open just for this purpose; change directories as needed for each new exercise, and run **console** from there.
- In Windows, an alternate approach which can make things much easier is to configure a new action on the Folder type. In Windows Explorer, choose Tools | Folder Options ... and the File Types tab. Right near the top you'll see the Folder type; select this and click Advanced. Use the dialog that appears to create a new action for this type, called "XPath/XSLT Console," that runs the **console** tool with the folder as it's working directory. You can then right-click any folder in Windows Explorer, choose this action from the popup menu, and work easily from there.
- Note carefully the instructions in some labs that point out that the console will not properly render certain results. The major limitations are HTML pages with external links (including images) and XML pages with associated CSS. In both cases the correct procedure is to save the result file (using **console** or **xform**) and open the file in a browser.





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

