

Preface

XML documents contain regular but flexible structures. Developers can use those structures as a framework on which to build powerful transformative and reporting applications, as well as to establish connections between different parts of documents. XPath and XPointer are two W3C-created technologies that make these structures accessible to applications. XPath is used for locating XML content within an XML document; XPointer is the standard for addressing such content, once located. The two standards are not typically used in isolation but in support of two critical extensions to the core of XML: Extensible Stylesheet Language Transformations (XSLT) and XLink, respectively. They are also finding wide use in other applications that need to reference parts of documents. These two closely related technologies provide the underpinning of an enormous amount of XML processing.

Who Should Read This Book?

Presumably, if you're browsing a book like this, you already know the rudiments of XML itself. You may have experimented with XSLT but, if so, haven't completely mastered it. (You can't do much in XSLT without first becoming comfortable with at least the basics of XPath.) Similarly, you may have experimented with XLinks; in this case, you've probably focused on linking to entire documents other than the one containing the link. XPointer will be your tool of choice for linking to portions of documents — external to or within the document where the XLink reference is made.

As support for XPath is integrated into the Document Object Model (DOM), DOM developers may also find XPath a convenient alternative to walking through document trees. Finally, developers interested in hypertext and other applications where references may have to cross node boundaries will find a thorough explanation of XPointer, the leading technology for creating those references.

You need not be an XML document author or developer to read this book. The XPath standard is fairly mature, and therefore is already incorporated in a number of high-level tools. XPointer, by contrast, is not yet a final standard; for this reason, the use of XPointers will probably be limited to experimental purposes in the short term.

Regardless of whether you're coming at the subject as primarily a document author or designer, or as a developer, *XPath and XPointer* can be revisited as often as you need it: for reference or as a refresher.

Who Should Not Read This Book?

If you don't yet understand XML (including XML namespaces) and have never looked at XSLT, you probably need to start with an XML book. John E. Simpson's *Just XML* (Prentice-Hall PTR) and Erik Ray's *Learning XML* (O'Reilly & Associates) are both good places to start.

Organization of the Book

[Chapter 1](#) introduces you to the foundations of XPath and XPointer, and where they're used.

[Chapter 2](#) gets you started with XPath's node tree model for documents and XPath syntax, as well as the set of node types accessible in XPath.

[Chapter 3](#) moves deeper into XPath, detailing the use of XPath axes, node tests, and predicates.

[Chapter 4](#) explains the tools XPath offers for manipulating content once it has been located.

[Chapter 5](#) demonstrates XPath techniques with over 30 examples using a wide variety of XPath parts.

[Chapter 6](#) examines the upcoming 2.0 version of XPath, including new features and interoperability issues.

[Chapter 7](#) explains XPointer's perspective on XML documents and how its use in URLs requires some changes from basic XPath.

[Chapter 8](#) explains the details of using XPointer syntax, including "bare names," child sequences, and interactions with namespaces.

[Chapter 9](#) delves deeper into XPointer, exploring the techniques XPointer offers for referencing points and ranges of text, not just nodes.

Conventions Used in This Book

The following font conventions are used throughout the book:

Constant width is used for:

- Code examples and fragments
- Anything that might appear in an XML document, including element names, tags, attribute values, entity references, and processing instructions
- Anything that might appear in a program, including keywords, operators, method names, class names, and literals

Constant-width bold is used for:



- User input
- Signifying emphasis in code statements

Constant-width italic is used for:

- Replaceable elements in code statements

Italic is used for:

- New terms where they are defined
- Pathnames, filenames, and program names
- Host and domain names (*www.xml.com*)

	This icon indicates a tip, suggestion, or general note.
	This icon indicates a warning or caution.

Please note that XML (and therefore XPath and XPointer) is case sensitive. Therefore, a `BATTLEINFO` element would not be the same as a `battleinfo` or `BattleInfo` element.

Comments and Questions

Please address comments and questions concerning this book to the publisher:

O'Reilly & Associates, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
(800) 998-9938 (in the United States or Canada)
(707) 829-0515 (international/local)
(707) 829-0104 (fax)

There is a web page for this book, which lists errata, examples, or any additional information. You can access this page at:

<http://www.oreilly.com/catalog/xpathpointer>

To comment or ask technical questions about this book, send email to:

bookquestions@oreilly.com

For more information about books, conferences, Resource Centers, and the O'Reilly Network, see the O'Reilly web site at:

<http://www.oreilly.com>

Acknowledgments

It's almost laughable that any technical book has just a few names on the cover, if that many. Such books are always the product of many minds and talents being brought to bear on the problem at hand.

For their help with *XPath and XPointer*, I am especially indebted to a number of individuals. Simon St.Laurent, my editor, has for years been a personal hero; I was flattered that he asked me to write the book in the first place and am grateful for his patience and support during its development. I came to XPath in particular by way of XSLT, and for this reason I happily acknowledge the implicit contributions to this book from that standard's user community, especially (in alphabetical order): Oliver Becker, David Carlisle, James Clark, Bob DuCharme, Tony Graham, G. Ken Holman, Michael Kay, Evan Lenz, Steve Muench, Dave Pawson, Wendell Piez, Sebastian Rahtz, and Jeni Tennison. J. David Eisenberg, Evan Lenz, and Jeni Tennison served as technical reviewers during the book's final preproduction stage; words cannot express how grateful I am for their patience, thoroughness, and good humor. Acknowledging the (unwitting or explicit) help of all those people does not, of course, imply that they're in any way responsible for the content of this book; errors and omissions are mine and mine alone.

I am also grateful to my colleagues and superiors in the City of Tallahassee's Public Works and Information Systems Services departments for their support during the writing of *XPath and XPointer*. They have endured far more than their deserved share of blank, preoccupied stares from me over the last few months.

Finally, to my wife Toni: to paraphrase Don Marquis's dedication to his *Archie and Mehitabel*, thanks "for Toni knows what/and Toni knows why."