

# Water: Simplified Web Services and XML Programming



# Water: Simplified Web Services and XML Programming

Mike Plusch



Wiley Publishing, Inc.

**Water: Simplified Web Services and XML Programming**

Published by  
Wiley Publishing, Inc.  
10475 Crosspoint Boulevard  
Indianapolis, IN 46256  
www.wiley.com

Copyright © 2003 by Clear Methods, Inc., Cambridge, MA

Published by Wiley Publishing, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 0-7645-2536-0

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

1B/SQ/RS/QS/IN

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 750-4744. Requests to the Publisher for permission should be addressed to the Legal Department, Wiley Publishing, Inc., 10475 Crosspoint Blvd., Indianapolis, IN 46256, (317) 572-3447, fax (317) 572-4447, E-Mail: permcoordinator@wiley.com.

**LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: WHILE THE PUBLISHER AND AUTHOR HAVE USED THEIR BEST EFFORTS IN PREPARING THIS BOOK, THEY MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS BOOK AND SPECIFICALLY DISCLAIM ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES REPRESENTATIVES OR WRITTEN SALES MATERIALS. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR YOUR SITUATION. YOU SHOULD CONSULT WITH A PROFESSIONAL WHERE APPROPRIATE. NEITHER THE PUBLISHER NOR AUTHOR SHALL BE LIABLE FOR ANY LOSS OF PROFIT OR ANY OTHER COMMERCIAL DAMAGES, INCLUDING BUT NOT LIMITED TO SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR OTHER DAMAGES.**

For general information on our other products and services or to obtain technical support, please contact our Customer Care Department within the U.S. at (800) 762-2974, outside the U.S. at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

**Library of Congress Cataloging-in-Publication Data**

**Trademarks:** Wiley, the Wiley Publishing logo and related trade dress are trademarks or registered trademarks of Wiley Publishing, Inc., in the United States and other countries, and may not be used without written permission. Water and Steam are trademarks of Clear Methods, Inc. All other trademarks are the property of their respective owners. Wiley Publishing, Inc., is not associated with any product or vendor mentioned in this book.

## About the Author

Mike Plusch is the co-designer of the Water language and the founder of Clear Methods, Inc. Clear Methods is building tools to commercialize Water technologies, and Plusch is currently the CTO. Plusch was previously a senior architect at Bowstreet, a pioneer in Web services. He also worked in the language group at Harlequin that developed compilers and tools for Dylan, ML, and Lisp.

While attending MIT, Plusch worked at the Center for Coordination Science on the Object Lens project. He did his thesis on Lego design collaboration tools at the MIT Media Lab in the Lego/Logo group. Plusch received a B.A. in Computer Science from MIT and a B.A. in Management from MIT Sloan.

# Credits

**EXECUTIVE EDITOR**

Chris Webb

**PROJECT EDITOR**

Chandani Thapa

**COPY EDITOR**

Jeremy Zucker

**EDITORIAL MANAGER**

Mary Beth Wakefield

**VICE PRESIDENT & EXECUTIVE**

**GROUP PUBLISHER**

Richard Swadley

**VICE PRESIDENT AND EXECUTIVE**

**PUBLISHER**

Bob Ipsen

**EXECUTIVE EDITORIAL DIRECTOR**

Mary Bednarek

**PROJECT COORDINATOR**

Dale White

**GRAPHICS AND PRODUCTION**

**SPECIALISTS**

Sean Decker

**QUALITY CONTROL TECHNICIAN**

Carl Pierce

**BOOK DESIGNER**

[paste text here]

**ILLUSTRATOR**

[paste text here]

**PROOFREADING AND INDEXING**

TECHBOOKS Production Services

**COVER IMAGE**

[paste text here]

# Preface

Water is a general-purpose programming language that is designed to build services and applications for the World Wide Web (Web) and for stand-alone programs. Water uses an XML syntax for representing code, data, and presentation. It extends XML to include full programming logic so it is easy to build new services and applications with XML. Water uses a ConciseXML syntax, which is a superset of XML and is backward compatible with XML 1.0. In other languages, XML is manipulated through an application programming interface (API). The object system for leading programming languages is completely different than the object structure for XML. This difference adds significant complexity when building XML-based software with other programming languages, because XML is treated as a foreign language that requires a special API to access data and call functions. Water eliminates the mismatch between XML and a programming language, because Water uses an XML syntax and has a type system and object system that is pure Web services and XML.

Today, when someone says that are building a Web service, they mean that they are creating an XML interface to some service written in a programming language. A full service has two parts: implementing the business logic in a programming language, and exposing the function through an XML interface. Water lets you write the logic and create the interface – all with an XML syntax. Many tools on the market can generate code to create an XML interface to the service. But Water is a pure Web service programming language, where services are XML by their very nature – no code generation required. This significantly reduces the required effort to create Web services.

## How This Book is Organized

The topics covered in this book would typically be found in more than ten different books. The challenge is to present the material in a way that the book can be read from beginning to end as well as serve as reference for a particular topic. Each chapter covers a different Water standard and has many small examples to help teach the topic – I wouldn't be surprised if there were over 1,500 examples in this book. The chapters are grouped into the following parts:

### Part I: Water Web Services and XML

The Water language covers much more than just programming. This part describes fifteen Water standards on Web services or XML. Each chapter focuses on a particular use of Water and builds upon the previous chapter.

## Part II: Water Programming

The Water language is a general-purpose programming language, and each chapter in this part describes a Water standard for dynamic processing and business logic.

## Part III: Advanced Water Programming

Part III describes the advanced Water standards for programming. Most of the Water standards covered in this part are not even possible in most programming languages, so please don't think Water is hard just because these topics are hard to understand. If you want to explore the true power of Water in building sophisticated abstractions, then this part is for you.

## Part IV: Water User Interface

This could have been the first part in the book, but I wanted to make sure that people didn't mistake Water for some toy HTML-based language. These chapters cover the Water standards for user interface, including input and display.

## Part V: Water Tools

Most languages don't standardize on the tools, and therefore each IDE implements its own standards. These Water standards include topics that are typically not part of a language specification. Part V includes chapters on testing, documentation, and debugging.

## Part VI: Water in Practice

These chapters cover the Water development process, a visual representation language for Water, and the building of a more complex application.

# Conventions Used in This Book

Each chapter in this book begins with a heads-up of the topics covered in the chapter and ends with a summary of what you should have learned by reading the chapter.

Throughout this book, you will find icons in the margins that highlight special or important information. Keep an eye out for the following icons:



A definition defines a particular term that is being introduced for the first time.

---



A cross-reference indicates where in the book you can find more information on the topic at hand.



A note means that the text that follows is a short aside to the main point.



A tip indicates some useful information that might be useful to you.

In addition to the preceding, the following formatting and typographical conventions appear throughout the book:

- ◆ A fixed width font is used for all code, output, as well as Web addresses (URIs).
- ◆ An arrow ( $\rightarrow$ ) separates the source code from the resulting output from executing the code, for example: `1.<plus 1/>`  $\rightarrow$  2
- ◆ *Italic* text is used on the first occurrence of an important term in the chapter.

### What Is a Sidebar?

A sidebar is used to provide additional information on the topic at hand. The information usually covers an advanced topic, and in most cases, the sidebar can be skipped on your first reading.

## Online Water Resources

The Water Language site ([www.waterlang.org](http://www.waterlang.org)) is the main resource for the Water language. The following things are available on the site:

- ◆ Executable examples from the book
- ◆ Corrections to the printed book
- ◆ Water articles
- ◆ Reference specifications for Water standards
- ◆ Water community
- ◆ Free downloadable Water IDE and runtime engine
- ◆ Production information and licensing
- ◆ List of Water consultants
- ◆ Links to Water user groups

I encourage you to join the Water language community and sign up for special announcements. This is the best way to keep current on developments in the Water language. The Water community site features links for reporting bugs and making suggestions. If you want to contact me directly, I can be reached at [mplusch@clearmethods.com](mailto:mplusch@clearmethods.com).

# Contents at a Glance

	Preface .....	vii
<b>Part I</b>	<b>Water Web Services and XML</b>	
<hr/>		
Chapter 1	Introduction to Pure Web Service Programming ...	3
Chapter 2	ConciseXML Syntax and XML 1.0 .....	15
Chapter 3	Water Object: Representing Objects in XML .....	21
Chapter 4	Water Type: Creating Custom Types – XSD .....	31
Chapter 5	Water Contract: Specifying APIs – DTD and Schemas .....	41
Chapter 6	Water Path: Accessing Data – XPATH .....	49
Chapter 7	Water Transform: Transforming XML Objects – XSLT .....	59
Chapter 8	Water Import: Converting Data to XML .....	73
Chapter 9	Water Export: Converting an Object to a String ...	83
Chapter 10	Water Identifier and URI: Naming Resources .....	91
Chapter 11	Water Filesystem: Using Files and Folders .....	99
Chapter 12	Water Web: Accessing Remote Resources – WSDL .....	107
Chapter 13	Water Protocol: Creating Custom Protocols – SOAP .....	111
Chapter 14	Water Server: Creating Application Servers .....	117
Chapter 15	Water Namespace: Dividing the Web .....	129
Chapter 16	Water Registry: Listing Resources – UDDI .....	133
<b>Part II</b>	<b>Water Programming</b>	
<hr/>		
Chapter 17	Water Flow: Controlling the Flow of Execution ...	141
Chapter 18	Water Logic: Performing Boolean Comparisons ...	155
Chapter 19	Water Methods: Learn Once, Use Everywhere ...	161
Chapter 20	Water Math: Representing Calculations in XML ...	177
Chapter 21	Water Date and Time: Managing Time .....	183
Chapter 22	Water Object System: Understanding Multi-role Objects .....	193
Chapter 23	Water Database: Unifying Relational and Object Models .....	199
Chapter 24	Water Bridge to Java: Integrating Water and Java. ....	213
Chapter 25	Water Pattern: Introducing Readable Regular Expressions .....	217

---

<b>Part III</b>	<b>Advanced Water Programming</b>	
<hr/>		
Chapter 26	Water Metafield: Adding Metadata to Fields . . . .	229
Chapter 27	Water Active Value: Changing Values from Static to Dynamic . . . . .	233
Chapter 28	Water Fluid Variable: Using Dynamic-Scoped Variables . . . . .	239
Chapter 29	Water View: Using A Capability Security Model . . . . .	245
Chapter 30	Water Execute: Simplifying How Software Writes Software . . . . .	253
<b>Part IV</b>	<b>Water User Interface</b>	
<hr/>		
Chapter 31	Water Hypertext: Using XHTML for Presentation . . . . .	263
Chapter 32	Water App: Building Applications with Water . . .	267
Chapter 33	Water Bridge to JavaScript: Adding Logic to Web Pages . . . . .	277
Chapter 34	Water Style: Adding Water to CSS . . . . .	285
Chapter 35	Water Color: Making Color an XML Object . . . . .	289
<b>Part V</b>	<b>Water Tools</b>	
<hr/>		
Chapter 36	Installing Water . . . . .	295
Chapter 37	Steam IDE: Using the Editor . . . . .	299
Chapter 38	Water Debug: Integrated Debugging with Development . . . . .	303
Chapter 39	Water Test: Building Self-Testing Software . . . .	309
Chapter 40	Water Doc: Integrating Documentation and Code . . . . .	317
<b>Part VI</b>	<b>Water in Practice</b>	
<hr/>		
Chapter 41	Water Process: Using Water Throughout the Software Lifecycle . . . . .	325
Chapter 42	Water Pics and Sketch: Unifying Diagrams and Code – UML . . . . .	329
Chapter 43	Building a Water App for Frequently Asked Questions . . . . .	351
	Index . . . . .	359

# Contents

	Preface .....	vii
<b>Part I</b>	<b>Water Web Services and XML</b>	
<b>Chapter 1</b>	<b>Introduction to Pure Web Service Programming</b> . . . .	3
	Why Do We Need Another Language? .....	3
	What is Water's Paradigm? .....	4
	Water Development Tools .....	4
	Hello World .....	5
	Water Web Services .....	6
	Creating a form .....	6
	Defining a Web service interface .....	7
	Creating a new Web service server .....	8
	Creating an XML object and custom data definition .....	9
	Implementing business logic in Water .....	10
	Calling a Web service over HTTP .....	11
	Summary .....	13
<b>Chapter 2</b>	<b>ConciseXML Syntax and XML 1.0</b> .....	15
	The World Used to be Flat .....	15
	XML is a Syntax .....	16
	Document versus Data .....	16
	ConciseXML and XML 1.0 .....	18
	Summary .....	20
<b>Chapter 3</b>	<b>Water Object: Representing Objects in XML</b> .....	21
	Understanding Water Objects .....	21
	Object .....	21
	Field .....	22
	Thing Class .....	22
	Naming .....	23
	Vector .....	24
	Primitive Objects .....	26
	Nested Objects .....	26
	Defining Class and Instance .....	27
	Examples of Water Objects .....	28
	Summary .....	30
<b>Chapter 4</b>	<b>Water Type: Creating Custom Types – XSD</b> .....	31
	Introducing Water Type .....	31
	number .....	31
	string .....	32
	char .....	32
	null .....	32

	Specifying Field Types .....	32
	Creating Custom Types .....	33
	one_of .....	34
	vector_of .....	35
	range_of .....	36
	not_of .....	37
	Using a Method as a Type .....	37
	Discussing Type versus Class .....	38
	Summary .....	39
<b>Chapter 5</b>	<b>Water Contract: Specifying APIs – DTD and Schemas .....</b>	<b>41</b>
	Defining a Contract with defclass .....	41
	Constraining Field Values Using Types .....	42
	Allowing Other Arguments .....	44
	Defining Method Contracts .....	45
	Specifying Method Parameters .....	46
	Allowing Other Keyed Arguments .....	46
	Specifying a Return Type .....	47
	Summary .....	48
<b>Chapter 6</b>	<b>Water Path: Accessing Data – XPATH .....</b>	<b>49</b>
	Retrieving Values .....	49
	Retrieving the Starting Object .....	50
	To Retrieve the Value of a Field .....	50
	Quoting Fields .....	50
	Retrieving a Nested Field Value .....	51
	Using a Variable for the Key of a Field .....	51
	Retrieving an Object Having a Field with a Specific Value .....	52
	Returning the Last Matching Object .....	53
	Returning All Matching Objects .....	53
	Filtering .....	54
	Filtering by Field Keys .....	54
	Filtering by Explicitly Named Fields .....	56
	Filtering by Field Value .....	57
	Water Paths and URIs .....	58
	Summary .....	58
<b>Chapter 7</b>	<b>Water Transform: Transforming XML Objects – XSLT .....</b>	<b>59</b>
	Simple Transformations .....	59
	Trimming Whitespace .....	60
	Transforming Multiple Fields .....	60
	Replacing Values .....	60
	Renaming Fields .....	61
	Removing Fields .....	62

---

	Advanced Transforms . . . . .	64
	Combining Multiple Fields into One Field . . . . .	64
	Splitting a Field . . . . .	66
	Sorting and Reversing . . . . .	66
	Conversions . . . . .	67
	Creating a Hierarchical Object from a Flat Object . . . . .	67
	Flattening a Nested Object . . . . .	68
	Transforming the Class or Type . . . . .	69
	Creating New Objects During Transformation . . . . .	70
	Sending Back a Modified Original . . . . .	71
	Summary . . . . .	72
<b>Chapter 8</b>	<b>Water Import: Converting Data to XML . . . . .</b>	<b>73</b>
	Importing non-XML Formatted Data . . . . .	73
	Converting Delimited Strings to XML Objects . . . . .	73
	Creating Instances from a string . . . . .	74
	Importing CSV Files . . . . .	75
	Importing Packed String Format . . . . .	75
	Importing XML Formatted Data . . . . .	77
	Converting a ConciseXML String to a Water Object . . . . .	77
	Converting XHTML to XML Objects . . . . .	78
	Converting Poorly Structured HTML . . . . .	78
	Converting HTML into Data . . . . .	79
	Converting XML 1.0 to ConciseXML . . . . .	79
	Summary . . . . .	82
<b>Chapter 9</b>	<b>Water Export: Converting an Object to a String . . . . .</b>	<b>83</b>
	Converting an Object to an XML string . . . . .	83
	Converting an Object to XML 1.0 . . . . .	84
	Converting an Object to ConciseXML . . . . .	86
	Converting Hypertext Object to XHTML . . . . .	86
	Converting an Object to a non-XML string . . . . .	87
	Converting an Object to Text . . . . .	87
	Comma Separated Values . . . . .	87
	Simple Reporting . . . . .	88
	Summary . . . . .	89
<b>Chapter 10</b>	<b>Water Identifier and URI: Naming Resources . . . . .</b>	<b>91</b>
	Creating Identifiers . . . . .	91
	Using Water Identifier . . . . .	92
	More Ways to Create a URI . . . . .	93
	Creating Relative Identifiers . . . . .	95
	Creating a relative URI from a Water Expression . . . . .	96
	Local Water Identifier . . . . .	97
	Summary . . . . .	97

<b>Chapter 11</b>	<b>Water Filesystem: Using Files and Folders</b> . . . . .	99
	Reading from a File . . . . .	99
	Writing to a File . . . . .	100
	Including Files . . . . .	101
	Importing and Exporting a File . . . . .	101
	Calling a File as a Method . . . . .	102
	Creating Folders . . . . .	102
	Creating Files and Folders . . . . .	104
	Batch File Processing . . . . .	105
	Summary . . . . .	106
<b>Chapter 12</b>	<b>Water Web: Accessing Remote Resources – WSDL</b> . . . . .	107
	Creating and Calling Water Web Resources . . . . .	107
	Specifying a Water Web Contract . . . . .	109
	Specifying the Protocol . . . . .	110
	Summary . . . . .	110
<b>Chapter 13</b>	<b>Water Protocol: Creating Custom Protocols – SOAP</b> . . . . .	111
	Looking at a First Example . . . . .	111
	Overview of a Protocol . . . . .	112
	Stages of a Web Protocol . . . . .	113
	Custom Protocol Pipeline . . . . .	114
	Summary . . . . .	115
<b>Chapter 14</b>	<b>Water Server: Customizing Application Servers</b> . . . . .	117
	Creating a Water Server . . . . .	117
	Serving XML Object . . . . .	119
	Creating a New Instance . . . . .	120
	Configuring a Water Server . . . . .	121
	Port . . . . .	122
	root_object . . . . .	122
	Creating an Application Server . . . . .	123
	Content Type and Format . . . . .	123
	Custom Water Server Processing . . . . .	124
	Creating a Pure Water Server . . . . .	126
	Summary . . . . .	127
<b>Chapter 15</b>	<b>Water Namespace: Dividing the Web</b> . . . . .	129
	Using Water Path for Namespaces . . . . .	129
	Using XML Namespaces . . . . .	131
	Using an Attribute Namespace . . . . .	132
	Summary . . . . .	132
<b>Chapter 16</b>	<b>Water Registry: Listing Resources – UDDI</b> . . . . .	133
	Publishing a Water Registry . . . . .	133
	Accessing Registry Resources . . . . .	135
	Viewing a Method as a Resource . . . . .	135
	Converting a Resource into a Local Proxy Method . . . . .	135

Using a Registry Listing . . . . . 136  
 Understanding Water Registry . . . . . 137  
 Summary . . . . . 138

**Part II Water Programming**

**Chapter 17 Water Flow: Controlling the Flow of Execution . . . . 141**  
 Conditional Flow Using if . . . . . 141  
     Executing More Than One Condition . . . . . 143  
     Looping with repeat . . . . . 144  
     Calling if from within a Water Path . . . . . 144  
 Looping and Iteration with for\_each . . . . . 145  
 Returning Immediately . . . . . 147  
 Error Handling . . . . . 147  
     Try . . . . . 148  
     Error . . . . . 148  
     Using get with if\_error . . . . . 149  
 Using Conditional Looping . . . . . 149  
     Using repeat for conditional looping . . . . . 150  
     Using while and until for conditional looping . . . . . 151  
 Summary . . . . . 154

**Chapter 18 Water Logic: Performing Boolean Comparisons . . . . 155**  
 Using Boolean Logic . . . . . 155  
     Defining false . . . . . 155  
     Converting to a Boolean value . . . . . 156  
     Understanding Water Logic Equality . . . . . 156  
     Is Not . . . . . 157  
     Defining 'Is' versus 'Equal' . . . . . 157  
 Making Comparisons . . . . . 158  
 Summary . . . . . 159

**Chapter 19 Water Methods: Learn Once, Use Everywhere . . . . 161**  
 Implementing a Method . . . . . 161  
 Generic Methods on Objects . . . . . 162  
     get . . . . . 162  
     length . . . . . 163  
     has\_key . . . . . 164  
     key\_of . . . . . 165  
     keys . . . . . 166  
     count . . . . . 166  
 Setting Variables . . . . . 167  
     Local variables with scope . . . . . 169  
     Fluid variables . . . . . 170  
 Setting Fields in an Object . . . . . 170  
     set . . . . . 170  
     set\_value . . . . . 171

	Other Generic Methods . . . . .	171
	insert . . . . .	171
	Combining two vectors . . . . .	172
	append . . . . .	172
	remove . . . . .	173
	last . . . . .	174
	subvector . . . . .	174
	subvectors . . . . .	174
	replace . . . . .	175
	rename . . . . .	175
	Summary . . . . .	176
<b>Chapter 20</b>	<b>Water Math: Representing Calculations in XML . . . . .</b>	<b>177</b>
	Understanding Numbers . . . . .	177
	Converting a string to a number . . . . .	177
	Converting a number to a string . . . . .	178
	Testing numbers . . . . .	179
	Using Basic Math Operations . . . . .	179
	Math expressions as a Water Path . . . . .	179
	No need for order of precedence rules . . . . .	180
	Creating custom math operations . . . . .	180
	Other Math Methods . . . . .	180
	Summary . . . . .	182
<b>Chapter 21</b>	<b>Water Date and Time: Managing Time . . . . .</b>	<b>183</b>
	Creating Dates . . . . .	183
	Relative Dates . . . . .	184
	Creating a Date from a Class . . . . .	184
	Date fields and standard formats . . . . .	185
	Custom date formatting . . . . .	186
	Creating Time . . . . .	186
	Relative Time . . . . .	187
	Time fields and formatting . . . . .	187
	Creating Datetime . . . . .	188
	Creating Duration . . . . .	189
	Methods on date, time, and duration . . . . .	190
	Creating a Datetime Interval . . . . .	191
	Summary . . . . .	192
<b>Chapter 22</b>	<b>Water Object System: Understanding</b>	
	<b>Multi-role Objects . . . . .</b>	<b>193</b>
	Understanding Inheritance . . . . .	193
	Understanding Water's Multiple Inheritance . . . . .	195
	Using Constructor and Method Calls . . . . .	195
	Comparing Class-Instance versus Multi-Role	
	Object System . . . . .	196
	Multi-role parts . . . . .	197
	Dynamic object system . . . . .	197
	Summary . . . . .	198

---

Chapter 23	<b>Water Database: Unifying Relational and Object Models</b> ..... 199 Retrieving and Updating Database Records ..... 199 Storing instances in the class ..... 200 Retrieving a field (column) value from an instance (record) ... 200 Returning number of instances (rows) ..... 200 Updating an instance (record) ..... 201 Retrieving an instance (row) from another class (table) ..... 201 Retrieving non-string field values ..... 201 Creating an instance (inserting a row) ..... 201 Creating Database Resources ..... 202 Classes of dbms and database ..... 202 Retrieving Tables and Rows ..... 202 Smart vector ..... 203 Accessing Rows ..... 203 Database table resource ..... 203 Database row resource ..... 204 Updating an instance (row) ..... 205 Selecting Records using Filter ..... 205 Sorting Records ..... 206 Sorting on a field ..... 207 Sort order ..... 207 Sorting on multiple fields ..... 208 Grouping Records ..... 208 Using the SQL Language ..... 211 Defining arguments to SQL method ..... 211 Calling an SQL method as a Water method ..... 211 Summary ..... 212
Chapter 24	<b>Water Bridge to Java: Integrating Water and Java.</b> ..... 213 Executing Java Code from Water ..... 213 Evaluating Java code ..... 214 Modifying the Java environment ..... 214 Water Calling Java ..... 214 Accessing fields of a Java class ..... 215 Manipulating and Creating Java instances ..... 215 Summary ..... 216
Chapter 25	<b>Water Pattern: Introducing Readable Regular Expressions</b> ..... 217 Creating and Using Water Patterns ..... 217 Validating String Data with Matches ..... 218 Defining a set of characters pattern with char_set ..... 219 Defining a set of strings pattern with one_of ..... 219 Matching a Quantity ..... 219 Listing of Predefined Patterns ..... 220 Using a Pattern to Split a String ..... 222

---

	Naming Subpatterns . . . . .	222
	Water Pattern versus Regex String . . . . .	224
	Summary . . . . .	225
<b>Part III</b>	<b>Advanced Water Programming</b>	
<hr/>		
<b>Chapter 26</b>	<b>Water Metafield: Adding Metadata to Fields . . . . .</b>	<b>229</b>
	Viewing a Field as an Object . . . . .	229
	Adding Fields to a Field Object . . . . .	230
	Learning When to Use Metafields . . . . .	231
	How Metafields Work . . . . .	232
	Summary . . . . .	232
<b>Chapter 27</b>	<b>Water Active Value: Changing Values from Static to Dynamic . . . . .</b>	<b>233</b>
	Understanding Active Get . . . . .	233
	Understanding Active Set . . . . .	235
	Active Value Patterns . . . . .	236
	Creating a One-Time Calculation Pattern with Active Values . . . . .	236
	Creating an Alias . . . . .	237
	Creating an Audit Log Pattern . . . . .	237
	Summary . . . . .	238
<b>Chapter 28</b>	<b>Water Fluid Variable: Using Dynamic-Scoped Variables . . . . .</b>	<b>239</b>
	Using Fluids . . . . .	239
	Defining a method in a fluid . . . . .	241
	Overriding a fluid . . . . .	241
	Understanding Where to Use Fluids . . . . .	242
	Lookup Across All Environments . . . . .	243
	Summary . . . . .	244
<b>Chapter 29</b>	<b>Water View: Using A Capability Security Model . . . . .</b>	<b>245</b>
	Understanding Conventional Security Access . . . . .	245
	Understanding Water View . . . . .	246
	Creating and Using a View . . . . .	247
	Creating a view during init . . . . .	248
	Views on classes . . . . .	249
	Creating Access Rights . . . . .	250
	Summary . . . . .	251
<b>Chapter 30</b>	<b>Water Execute: Simplifying How Software Writes Software . . . . .</b>	<b>253</b>
	Delaying the Execution of Arguments . . . . .	254
	Creating Hypertext Tags Using ek_hypertext . . . . .	255
	execution kind of ek_string . . . . .	256
	execution kind of ek_expression . . . . .	256
	Understanding the Stages of Execution . . . . .	257

	Using Execution Kinds . . . . .	257
	Creating a While-Loop . . . . .	258
	Creating a Testing System . . . . .	259
	Summary . . . . .	260
<b>Part IV</b>	<b>Water User Interface</b>	
<hr/>		
<b>Chapter 31</b>	<b>Water Hypertext: Using XHTML for Presentation . . . . .</b>	<b>263</b>
	Understanding String versus Hypertext . . . . .	263
	Manipulating hypertext objects . . . . .	264
	Embedding Water code in hypertext . . . . .	265
	Using hypertext for concatenating values . . . . .	265
	Creating Custom Presentation Methods . . . . .	265
	Summary . . . . .	266
<b>Chapter 32</b>	<b>Water App: Building Applications with Water . . . . .</b>	<b>267</b>
	Creating a Simple Application . . . . .	267
	Calling the start page . . . . .	267
	Creating an Application Server . . . . .	268
	Extending a Simple Application . . . . .	268
	Multiple pages . . . . .	269
	Adding a page method . . . . .	270
	Calling a dynamic page with an HTML form . . . . .	271
	Subapplications . . . . .	272
	HTML Form Inputs . . . . .	273
	An Example Application . . . . .	275
	Summary . . . . .	276
<b>Chapter 33</b>	<b>Water Bridge to JavaScript: Adding Logic to Web Pages . . . . .</b>	<b>277</b>
	Using JavaScript in Water . . . . .	277
	Standalone JavaScript . . . . .	278
	JavaScript in Attributes . . . . .	279
	JavaScript Calling Water Server . . . . .	279
	Encoding Water Objects as JavaScript . . . . .	281
	Running Water Script on a Browser . . . . .	283
	Summary . . . . .	284
<b>Chapter 34</b>	<b>Water Style: Adding Water to CSS . . . . .</b>	<b>285</b>
	Introducing Water Style . . . . .	285
	Inheriting Style Properties . . . . .	286
	Summary . . . . .	287
<b>Chapter 35</b>	<b>Water Color: Making Color an XML Object . . . . .</b>	<b>289</b>
	Using Color in HTML . . . . .	289
	Creating Custom Named Colors . . . . .	291
	Summary . . . . .	291

<b>Part V</b>	<b>Water Tools</b>	
<b>Chapter 36</b>	<b>Installing Water</b> . . . . .	295
	Installing Water . . . . .	295
	Running Your First Water Program . . . . .	296
	Summary . . . . .	297
<b>Chapter 37</b>	<b>Steam IDE: Using the Editor</b> . . . . .	299
	Executing Selections of Code . . . . .	299
	Highlighting a Water expression . . . . .	299
	Displaying an error . . . . .	300
	Understanding the Menus . . . . .	301
	Summary . . . . .	301
<b>Chapter 38</b>	<b>Water Debug: Integrated Debugging with Development</b> . . . . .	303
	Debugging Water Programs . . . . .	303
	Print to Console . . . . .	304
	Inspecting Objects . . . . .	304
	Other Features . . . . .	306
	Stepping . . . . .	306
	Viewing Past Values of an Expression . . . . .	306
	Remote Inspection . . . . .	306
	Summary . . . . .	307
<b>Chapter 39</b>	<b>Water Test: Building Self-Testing Software</b> . . . . .	309
	Creating Test Cases . . . . .	309
	Testing with a custom comparison . . . . .	310
	Testing with setup code . . . . .	311
	Testing for an error . . . . .	311
	Associating a test with an object . . . . .	311
	Other Testing . . . . .	312
	Test suite . . . . .	312
	Storing tests and running multiple tests . . . . .	312
	Timing Tests . . . . .	313
	Assertions . . . . .	313
	Summary . . . . .	316
<b>Chapter 40</b>	<b>Water Doc: Integrating Documentation and Code</b> . . . . .	317
	An Example of Water Documentation . . . . .	317
	Where to Keep Documentation . . . . .	318
	Documentation Examples . . . . .	319
	Additional options for associating documentation . . . . .	319
	Multiple documentation objects . . . . .	320
	Creating and Viewing Documentation . . . . .	321
	Custom Documentation Groups . . . . .	321
	Documentation Arguments . . . . .	322
	Summary . . . . .	322

---

<b>Part VI</b>	<b>Water in Practice</b>	
<b>Chapter 41</b>	<b>Water Process: Using Water Throughout the Software Lifecycle</b>	325
	Using Water for Building Software	325
	Gathering Requirements	326
	Using Water Across All Stages	326
	Analysis	327
	Design	327
	Implementation	327
	Testing	327
	Documentation	328
	Deployment	328
	Maintenance	328
	Summary	328
<b>Chapter 42</b>	<b>Water Pics and Sketch: Unifying Diagrams and Code – UML</b>	329
	Using Water Sketch for Business	330
	Visualizing Objects	330
	Object views and identity	332
	A vector	332
	Primitive objects	333
	Understanding Classes and Fields	334
	Fields and relationships	334
	Field type	335
	Visualizing Logic and Flow	336
	Method as a box	338
	Call	338
	Visualizing an Implementation	340
	Dotted circle for do and set	344
	Fluid variables	345
	Showing a Class	346
	Viewing a State Machine	347
	Water Pics and UML	348
	Summary	349
<b>Chapter 43</b>	<b>Building a Water App for Frequently Asked Questions</b>	351
	Reviewing the Requirements	351
	Implementation	352
	Walking through the Design and Implementation	354
	Summary	358
	<b>Index</b>	359

