

Contents

Acknowledgments	xi
Foreword	xiii
Introduction	xxv

Part I: The Core Language

Chapter 1: Introducing haXe **3**

A Problem with Internet Development	3
What Is haXe and Why Was It Created?	4
The haXe Compiler	5
The haXe Language	7
How Does haXe Work?	8
So What Is Neko?	8
haXe and Neko Requirements	9
Summary	10

Chapter 2: Installing and Using haXe and Neko **11**

Installing haXe	11
Using the haXe Automatic Installer	12
Manually Installing haXe on Windows	12
Manually Installing haXe on Linux	13
Installing Neko	14
Manually Installing Neko on Windows	14
Installing Neko on Linux	16
Hello World! in Triplicate	17
Compiling Your First haXe Application	17
How Your Program Was Compiled	18
HXML Compiler Files	19
Compiling to Neko	19
Compiling to JavaScript	21
Program Structure	22
Compiler Switches	23
Summary	25

Chapter 3: Learning the Basics	27
The haXe Hierarchy	27
The Standard Data Types	28
Variables	28
Type Inference	30
Constant Variables	31
The Simple Value Types	31
Floating-Point Numbers	32
Integers	32
Which Number Type?	33
Booleans	33
Strings	33
Abstract Types	35
Void and Null	35
The Dynamic Type	35
The unknown Type	36
Bypassing Static Typing with untyped	36
Commenting Your Code	37
Block Commenting	37
Line Commenting	37
Converting Data Through Casting	38
The cast Function	38
Simple Type Conversion Functions	39
Comparing the Type of a Value	40
Using Arrays in haXe	40
Arrays	41
Lists	46
Hash Tables	48
Using Dates in haXe	50
Creating a Date Object	51
Retrieving the Components of a Date	52
The DateTools Class	53
Manipulating Data	55
Operators	55
The Math Class	63
Using String Functions	66
Summary	72

Chapter 4: Controlling the Flow of Information	73
Pushing Data Around	73
Conditional Statements	74
if Statements	74
switch Statements	77
Returning Values from Conditional Statements	78
Loops	79
The while Loop	80
The for Loop	81
Looping Over Collections	82
Break and Continue	84
Functions	85
Class Functions	85
Local Functions	90
The Lambda Class	96
Summary	99
Chapter 5: Delving Into Object-Oriented Programming	101
Classes and Objects	102
Instance Fields	104
Static Fields	114
Understanding Inheritance	115
Super	117
Functions Override	117
Constructor Override	118
toString()	119
Abstract Classes and Abstract Methods	119
Static Fields, Instance Variables, and Inheritance	120
Inheritance Roles	120
Using Interfaces	121
Advanced Classes and Objects Features	122
Implementing Classes	123
Type Parameters	123
Anonymous Objects	126
Implementing Dynamic	127
Typedef	128
Extensions	133
Enum	134
Constructor Arguments	136
Summary	138

Chapter 6: Organizing Your Code **139**

Building Reusable Code	139
Using Packages	140
Declaring a Package	140
Implicit Import	141
Explicit Import	142
Enum and Packages	143
Type Lookup Sequence	144
Importing an Entire Package	144
Importing Libraries	145
haXe Standard Library	145
haxelib Libraries	146
Libraries from Other Projects	150
External Libraries	150
Using Resources	163
Documenting Your Code	166
Offline Documentation	167
Online Documentation	169
Unit Testing	170
The haxe.unit Package	171
Writing Tests	172
Summary	177

Chapter 7: When Things Go Wrong **179**

The Trace Function	179
Trace Output	180
haXe Trace versus ActionScript Trace	181
Exceptions	185
Dealing with Exceptions	186
The CallStack and ExceptionStack	191
The Exception Master Class	194
Creating a Complete Exception Class	194
The Exception Class Code	197
Summary	199

Part II: Server Side, JavaScript, and Flash; Oh My!

Chapter 8: Cross Platform Tools **203**

XML	203
XML Anatomy	203
haXe XML API	206

Regular Expression	210
The EReg Class	210
Patterns	214
Timer	222
Delayed Action	223
Queued Actions	223
MD5	224
Summary	224
Chapter 9: Building Websites with HaXe	225
Introduction to Web Development	225
What Is a Web Server?	226
Distributing Contents Using a Web Server	226
An HTML Crash Course	228
How haXe Differs from HTML	230
The NekoTools Web Server	231
Installing mod_neko For Apache	232
Installing Apache and mod_neko on Windows XP	232
Installing Apache and Mod_Neko on Linux (Ubuntu)	233
Your First Website with haXe	234
Using Neko as a Page Controller	235
The neko.Web Class	236
Using Neko as a Front Controller	238
Summary	250
Chapter 10: Separating Design Using Templates	251
What Are Templates?	251
The Template Class	252
The Template Syntax	253
Using Resources	257
When to Use Code in Templates?	258
Templo for Server-Side Templates	258
Installing Templo	259
Using Templo	259
Differences in Expressions Between haxe.Template and mtwin.Templo	261
Attr Expressions	262
Raw Expressions	262
Logical Expressions	263
Loop Expressions	263
set, fill, and use Expressions	264

Contents

Using Macros with Templo	266
Compiling Templates Manually	268
Summary	269
Chapter 11: Performing Server-Side Trickery	271
Introducing Neko	271
The Neko Virtual Machine	272
The Neko Standard Libraries	272
Working with Databases	273
Choosing a Database	273
Connecting to the Database	276
The Connection Class	277
The ResultSet Class	283
Using the SPOD System	286
Getting Started	287
Simple Use of the SPOD System	287
Table Mapping with neko.db.Object	289
The neko.db.Manager Class	291
Adding Table Relations	295
Working with Files	301
Traversing Files and Directories	301
Working with Files and Streams	304
Creating a Logger Class	311
Summary	314
Chapter 12: Building Interactive Content with Flash	315
The Flash Library	315
The Flash Movie	316
The Movie Clip	319
Text	322
Multimedia Contents	323
Loading Values	325
Events	328
Events over Time	331
The Drawing API	337
Extending the MovieClip Class	342
Flash to haXe	344
Data Types	344
Global Constants and Special Values	346
Top-level Functions	347

Operators	352
Statements, Structures, and Keywords	354
Global Properties	356
Top-level Classes Replaced by haXe-Specific Types	357
Flash API Classes	359
Summary	362
Chapter 13: Replacing the Need for an IDE	363
Assets with SWFMill	364
Creating an Image Movie Clip	365
Embedding Fonts	368
User Interfaces	369
How to Implement an UI	369
Events	371
Views, Controls, Containers, and Elements	373
Element	373
Controls	375
Containers	376
The Flash 9 Implementation	381
A Complete Example: the Tree Editor	381
Conclusions	384
Summary	385
Chapter 14: More Interactive Content with JavaScript	387
Dynamic Content in JavaScript?	387
The Web Developer Tools	389
Trace to Firebug	389
JavaScript as haXe Target	390
The JavaScript Library	391
Manipulating the Page with JavaScript	396
The OnLoad Problem	396
Advanced JavaScript Libraries	399
Traversing HTML with JavaScript	400
What Is AJAX?	414
Removing the Myths Around AJAX Complexity	415
Updating a Panel Content Using AJAX	416
Building an Auto-Complete Control	417
Advantages of haXe/JS Development	422
Summary	423

Chapter 15: Putting It All Together with haXe Remoting **425**

What Is haXe Remoting?	425
Remote Connection and Local Connection	426
Synchronous and Asynchronous Communications	426
Pull or Push?	426
The haXe Remoting API	427
Connection	428
AsyncConnection	432
FlashJsConnection	435
LocalConnection	435
SocketConnection	436
NekoSocketConnection	436
AsyncDebugConnection, DelayedConnection, and AsyncAdapter	437
SocketProtocol	437
Proxy Objects	437
TicTacToe Game	443
Summary	452

Part III: Extending the Possibilities

Chapter 16: haXe Advanced Topics **455**

XML	455
Fast	455
Check	458
XML Proxy	463
Reflection API	464
The Reflect Class	464
The Type Class	467
Runtime Type Information (RTTI)	470
Serialization	473
haXe Magic	475
Summary	478

Chapter 17: Desktop Applications with Neko **479**

Why Use Neko for Desktop Applications?	479
Creating a Neko Executable	480
Graphical User Interface Libraries	480
ScreenWeaver HX	480
Xinf: Xinf Is Not Flash	481

nGui	481
hxGtk	481
Installing the Required Library	481
The nGui Library	482
The nGui Manager Class	482
Working with Static Controls	484
The Control Class	485
Container Classes	487
Working with Interactive Controls and Events	488
Working with Menus	496
The hxGtk Library	499
The hxGtk Function Proxy	499
The Gtk Class	501
hxGtk Events	502
Rapid hxGtk GUI Development with Glade	503
The Glade IDE	504
The Glade Class	505
Summary	506
<hr/> Chapter 18: Desktop Flash with SWHX	<hr/> 509
Why SWHX?	510
Installing SWHX	511
Getting Started with SWHX	511
Working with Windows	513
Window Decoration Flags	513
Window Properties	514
Window Events	515
Communication Between Flash and Neko	517
Asynchronous Communication	518
Synchronous Communication in haXe	520
Synchronous Communication in ActionScript	523
Using SysTools	525
Creating a System Tray Icon	526
Creating Menus	526
Using Dialogs	529
Summary	530
<hr/> Chapter 19: Multimedia with Neko	<hr/> 531
Why Use Neko for Games Development?	532
Neko's NME is Your Friend	532

Contents

Getting Started with NME	533
The Manager Class	534
Writing Text to the Display	537
Working with Surfaces	537
Keeping Time with Timers	546
The Wonderful World of Sprites	546
Interaction Through Events	549
Summary	553
<hr/> Chapter 20: Extending haXe with C/C++	<hr/> 555
Neko Libraries	555
Setting Up the Compiler	556
A Simple Hello World Example	556
Building the ndll Library in C	556
Building the haXe Script	557
The Post-Mortem	557
Neko Value Types in C/C++	558
The Neko value Struct	558
Preparing Data for Transit from haXe to C/C++	559
Restructuring Data in haXe from C/C++	560
Neko Type Checking in C/C++	560
Passing C/C++ Values to Neko	562
Abstract Types	563
Object Types	565
Arrays	567
Functions	568
Managing Neko Primitives	570
Garbage Collection	571
alloc_root()	571
free_root()	572
alloc()	572
Garbage Collecting Abstract Values	572
Error Handling	573
val_throw() and val_rethrow()	573
failure()	573
bfailure()	574
val_print()	574
neko_error()	574
Summary	574
<hr/> Appendix A: Semantic HTML	<hr/> 577
Index	603