Professional
haXe and Neko

Franco Ponticelli
Lee McColl-Sylvester

Wiley Publishing, Inc.

Professional
haXe and Neko

Part I: The Core Language

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

Part III: Extending the Possibilities

Professional
haXe and Neko

Franco Ponticelli
Lee McColl-Sylvester

Wiley Publishing, Inc.

Professional haXe and Neko

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

Copyright © 2008 by Wiley Publishing, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-0-470-122135-6

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Library of Congress Cataloging-in-Publication Data:

Ponticelli, Franco, 1972 -
Professional haXe and Neko / Franco Ponticelli and Lee McColl-Sylvester.
p. cm.
Includes index.
ISBN 978-0-470-12213-6 (pbk. : website)
1. Object-oriented programming (Computer science) 2. Virtual computer systems.
I. McColl-Sylvester, Lee, 1976 - II. Title.
QA76.64.P645 2008
006.8—dc22

2007045705

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) 646-8600. 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) 572-4355, or online at http://www.wiley.com/go/permissions.

Limit of Liability/Disclaimer of Warranty: The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Website is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or Website may provide or recommendations it may make. Further, readers should be aware that Internet Websites listed in this work may have changed or disappeared between when this work was written and when it is read.

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

Trademarks: Wiley, the Wiley logo, Wrox, the Wrox logo, Wrox Programmer to Programmer, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. 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.

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

To Cristina and Gabriel, the meaning of my life
—Franco Ponticelli

For Jay, Tabitha, Reuben, and Bradley
"The best feelings are those that have no words to describe them . . ."
—Lee McColl-Sylvester

About the Authors

Franco Ponticelli is an experienced developer and solution architect. An Architecture Graduate with specialization in Industrial Design, he performed many different activities in the Information Technology area from 3D Computer Graphics to hard-core software development. In his continual research for the perfect development environment, he found haXe and fell in love.

Franco is currently a self-employed developer and you can reach him on his personal website `www.weblob.net`.

Lee McColl-Sylvester is an expert ActionScript developer as well as a seasoned master in systems integrations. A student in Visual Communications, he evolved his career specializing in advanced graphical interface development, as well as information management systems, database architecture, and hardware communications implementation. An inventor at heart, Lee discovered haXe while tinkering in the Open Source Flash arena.

Lee is self-employed and can be reached through the haXe mailing list, or his website `www.designrealm.co.uk`.

Credits

Executive Editor
Chris Webb

Development Editor
Ed Connor

Technical Editor
Daniel Fischer

Production Editor
Martine Dardignac

Copy Editor
Mildred Sanchez

Editorial Manager
Mary Beth Wakefield

Production Manager
Tim Tate

Vice President and Executive Group Publisher
Richard Swadley

Vice President and Executive Publisher
Joseph B. Wikert

Project Coordinator, Cover
Lynsey Stanford

Proofreader
Sossity Smith

Indexer
Melanie Belkin

Acknowledgments

My first and foremost thanks go to Gisella, without her support, it would have been impossible for me to realize this work.

Special thanks goes to co-author Lee McColl-Sylvester, who involved me in this project in the first place. I sincerely hope to work with him again because of his professionalism, kindness, and competence.

Many thanks also to Nicolas Canasse, the haXe author, whose expertise and talent are incomparable. Many thanks to all of the members of the haXe mailing list too, they are always ready to give you an answer to every doubt or concern you can have about the haXe experience. You can reach the mailing list at this address: `http://lists.motion-twin.com/mailman/listinfo/haxe`.

A big hug goes to Ed Connor, Daniel Fischer, and Chris Webb whose help has been fundamental. I really appreciated their aid and guidance.

Finally, I have to thank my family and friends who always support my choices unconditionally.

Franco Ponticelli

From the deepest of my heart, I'd like to thank my wife, Jay, for her patience and understanding at a time without reason. Also, my children, Tabitha, Reuben and Bradley, who never fail to make me smile.

I'd like to thank Franco Ponticelli for being a fantastic co-author. This book would never have been possible without his support, enthusiasm, dedication, and friendship. I'd also like to thank Nicolas Cannasse for his help and advice, and for the haXe language and Neko platform.

A big thank you to Daniel Fischer for his excellent efforts as a Technical Editor. Also, Ed Connor and Chris Webb for their understanding, motivation and tireless support as editors.

Thanks to Ritchie "Blackdog" Turner, Grant "Resident Alien" Davies, and Kevin "The Masked Coder" Murphy for their advice and friendship.

More thanks to Wesley Russell Blampied for giving me the idea to write this book. Also, Glynn Hayward, James Jackson and Nigel Britton of Complete Control UK Limited for their support, kindness, generosity, and understanding. I did it, guys!

Thanks to my old team at Element 78 Ltd for joining me on many years of Flash coding. It was always a pleasure working with them, and I miss every minute of it.

Finally, thanks to the whole haXe and Neko mailing lists for their dedication to a fantastic family of development tools and to those who learn a little something from the words contained in this book.

I thank you. . .

Lee McColl-Sylvester

Foreword

Writing a book about haXe is not an easy task. Not because there are complicated things to explain: haXe is a new programming language. Its features are mostly coming from classic object-oriented languages such as Java, with some having been taken from more dynamic Scripting languages, and some others from Functional languages. This mix makes haXe a unique language, but all these features one by one are pretty easy to explain and understand.

What makes a haXe book difficult to write is the amount of possibilities that haXe opens. By being able to target three different platforms — JavaScript, Flash, and Neko — haXe opens a lot of doors for the web developer. And because Neko is an extensible virtual machine, using haXe opens more doors behind pure web development, such as real-time servers, console and desktop applications, and all the things that a modern programming language can do.

This is the main reason why the haXe community is so diverse. People are coming from different languages with different goals. Some are tired of JavaScript and want to use a modern language such as haXe to speed up web development, some want to develop Flash content without relying on proprietary tools, some want to experiment with scalable technologies for website development and database handling, some even want to develop desktop OpenGL games using haXe!

This diversity is a gift. By combining different technologies, it's now possible to create things that were yesterday very difficult and costly to realize because of the difficulty of learning and integrating these different technologies together. By using haXe everywhere, it's now possible to quickly develop multitechnologies applications, such as highly interactive Flash-JS-Server websites. By using haXe everywhere, people with knowledge in one of these platforms can now leverage their skills and use it to develop for other platforms as well, as soon as they need it. This is the biggest success of haXe: opening doors for developers so that different technologies and platforms are no longer a limitation for creativity.

But that's also a problem when it comes to presenting haXe clearly: explaining its different possible applications already takes a few lines. Entering into details and showing examples of all the possibilities that haXe offers is very time consuming and requires time to learn and experiment with a lot of technologies. While this book mostly will focus on web development: web servers, databases, Flash and JavaScript, it will also give you the keys for other technologies, such as real-time servers, that will open as many doors.

It's for these reasons that I really want to thank the book's authors: Lee McColl-Sylvester and Franco Ponticelli, for their work on this book, which I sincerely think will provide you with a very good introduction, samples, and detailed explanations of the most important aspects of the haXe Programming Language. I thank them for their hard work in putting all these pieces together.

Last thing: If I can give you a tip for reading this book, it's not to try to read every single page of it before starting using haXe. First focus on the basics, then read the parts that you are more familiar with or that you are most eager to learn, then experiment, modify, and try it by yourself. If you have questions, contact the haXe mailing list; there will be people there ready to help you. Don't read this book like you would read a novel, but more like an encyclopedia.

Thank you, and don't forget to have fun,

Nicolas Cannasse
haXe creator

Contents

Introduction

haXe is a language, a compiler, and a framework of classes designed to empower the developer to write clean code targeted at multiple platforms on multiple operating systems. Based on the famous open source MTASC ActionScript compiler and written by the author of MTASC, haXe opens up the world of applications development while reducing the learning curve and minimizing roadblocks.

Neko is a cross-platform virtual machine, and one of the targets of the haXe compiler. It is lightweight, as well as incredibly fast, flexible, extensible, and reliable. It can be embedded or extended and requires a footprint so small as to appear almost invisible.

Professional haXe and Neko aims to target these two tools and the relationship between them. Within the pages of this book, you will learn how to command the awesome power of the Flash platform, how to develop server-side web applications with ultimate functionality, how to manipulate client web browsers with unmatched JavaScript, and how to combine all three into unprecedented desktop applications. Professional haXe and Neko will take you on a ride through the leading technologies in the industry and show you how to bend them to your will. Plus, you'll have fun doing it.

Written by two guys who love haXe and Neko with a passion, this book aims to reveal how haXe and Neko can render all other web-related technologies obsolete. From the very first chapter, you'll be taken on a detailed journey through the capabilities of these technologies and you'll stretch them to the limit. You'll learn how to master graphical user interface development for a multitude of purposes, program reusable code modules using a true ECMAScript language, tackle the browser DOM with object-oriented JavaScript, and more.

Professional haXe and Neko is the first book to cover these fantastic new technologies, and may possibly be the last book you ever need.

Whom This Book Is For

This book is for existing web developers who wish to work with a unified environment and reduce the number of languages required to produce a single software solution. You don't have to be a seasoned programmer to understand the concepts within this book, though if you have already programmed in ActionScript, Java, or some other ECMA standard language, then things will come to you a lot faster. As strange as it sounds, most developers who come to haXe are those who do not want to learn a new language. Many ActionScript 2.0 developers, for example, may choose not to progress to ActionScript 3.0, simply because it follows a slightly different syntax. haXe, however, maintains a singular language, regardless of target platform, and therefore preserves the developer's language experience.

What This Book Covers

At the time of writing, the versions of the software covered by this book are haXe version 1.15 and Neko version 1.6.0. Both haXe and Neko are in constant evolution as there is always a constant activity in the various community meeting places, and it would not be surprising to see two or more minor releases

appear between the book's completion and the publishing date. However, new releases do not usually break backward compatibility, so you need not worry about information within this publication being out of date. If you are in any doubt about the content of this book, just check the haXe website, which will list the features added and direct you to how these changes have affected the book.

This book describes the use of haXe to target the generation of output for the following platforms: Neko, Flash Player versions 6 to 9, and JavaScript. Within the pages of this book, you will learn everything you need to produce first class applications for any of these environments.

How This Book Is Structured

Professional haXe and Neko is split into three parts, depending on the content being delivered.

The first part of the book is dedicated to the basics of the haXe language, providing a thorough detail of the language structure, standard libraries, and programming best practices. An experienced haXe programmer can skip this section entirely while a seasoned programmer learning haXe for the first time can scan it rapidly and take a deeper look at both the code examples and the tables detailing the classes and commands required to develop in haXe.

The second part of the book covers practical aspects of haXe, such as producing Flash movies and dynamic content for websites. This part is of interest to programmers of any skill level. It also contains references to the vast majority of platform-specific classes defined in the standard library, which is the core framework that comes with the haXe base installation. Some external libraries from the haXeLib repository are also described.

The third part of the book is dedicated to the more advanced developer who wants to push the haXe language to its limit, and seeks to extend haXe with existing libraries or use haXe outside of the conventional web environment.

What You Need to Use This Book

Developing with haXe and Neko is a cheap affair, as all the tools described in the book are open source and available for free on the Internet. You will find there aren't any particular requirements, and both haXe and Neko will operate on Windows, Linux, and both types of Apple Macintosh. All you need to get started with haXe and Neko is to follow the step-by-step procedures as outlined in Chapter 2.

The use of a more sophisticated editor can improve your productivity dramatically. *FlashDevelop* (`www.flashdevelop.org/`) is a very good alternative to tools such as Notepad in the Windows environment, and can certainly save a lot of headaches when you first start out. The current version of FlashDevelop supports haXe directly out of the box, while older versions can install the *haxefd* plug-in from the *haxelib* repository. An alternative development environment for the Linux/Mac OS X operating systems is the promising *haxeDevelop* (`http://code.google.com/p/haxedevelop/`). As more and more editors support the haXe language, they will begin to appear on the haXe mailing list, so be sure to keep an eye open.

Conventions

To help you get the most from the text and keep track of what's happening, we've used a number of conventions throughout the book.

> **Boxes such as this one hold important, not-to-be forgotten information that is directly relevant to the surrounding text.**

Notes, tips, hints, tricks, and asides to the current discussion are offset and placed in italics like this.

As for styles in the text:

- We *highlight* new terms and important words when we introduce them.
- We show filenames, URLs, and code within the text like so: `Reflect.field()`.
- We present code in two different ways:

```
We use a monofont type with no highlighting for most code examples.
```

```
We use gray highlighting to emphasize code that's particularly important in the
present context.
```

Commands that must be executed in the DOS prompt (for Windows) or in the terminal/console (for Mac OS X and Linux) are shown prefixed with an angular bracket followed by a whitespace.

```
> haxe build.hxml
```

Source Code

As you work through the examples in this book, you may choose either to type in all the code manually or to use the source code files that accompany the book. All of the source code used in this book is available for download at `www.wrox.com`. Once at the site, simply locate the book's title (either by using the Search box or by using one of the title lists) and click the Download Code link on the book's detail page to obtain all the source code for the book.

Because many books have similar titles, you may find it easiest to search by ISBN; this book's ISBN is 978-0-470-12213-6.

Once you download the code, just decompress it with your favorite compression tool. Alternately, you can go to the main Wrox code download page at `www.wrox.com/dynamic/books/download.aspx` to see the code available for this book and all other Wrox books.

Errata

We make every effort to ensure that there are no errors in the text or in the code. However, no one is perfect, and mistakes do occur. If you find an error in one of our books, such as a spelling mistake or faulty piece of code, we would be very grateful for your feedback. By sending in errata you may save another reader hours of frustration and at the same time you will be helping us provide even higher quality information.

To find the errata page for this book, go to `www.wrox.com` and locate the title using the Search box or one of the title lists. Then, on the book details page, click the Book Errata link. On this page you can view all errata that has been submitted for this book and posted by Wrox editors. A complete book list including links to each book's errata is also available at `www.wrox.com/misc-pages/booklist.shtml`.

If you don't spot "your" error on the Book Errata page, go to `www.wrox.com/contact/techsupport.shtml` and complete the form there to send us the error you have found. We'll check the information and, if appropriate, post a message to the book's errata page and fix the problem in subsequent editions of the book.

p2p.wrox.com

For author and peer discussion, join the P2P forums at `p2p.wrox.com`. The forums are a web-based system for you to post messages relating to Wrox books and related technologies and interact with other readers and technology users. The forums offer a subscription feature to e-mail you topics of interest of your choosing when new posts are made to the forums. Wrox authors, editors, other industry experts, and your fellow readers are present on these forums.

At `http://p2p.wrox.com` you will find a number of different forums that will help you not only as you read this book, but also as you develop your own applications. To join the forums, just follow these steps:

1. Go to `p2p.wrox.com` and click the Register link.
2. Read the terms of use and click Agree.
3. Complete the required information to join as well as any optional information you wish to provide and click Submit.
4. You will receive an e-mail with information describing how to verify your account and complete the joining process.

You can read messages in the forums without joining P2P but in order to post your own messages, you must join.

Once you join, you can post new messages and respond to messages other users post. You can read messages at any time on the Web. If you would like to have new messages from a particular forum e-mailed to you, click the Subscribe to this Forum icon by the forum name in the forum listing.

For more information about how to use the Wrox P2P, be sure to read the P2P FAQs for answers to questions about how the forum software works as well as many common questions specific to P2P and Wrox books. To read the FAQs, click the FAQ link on any P2P page.