Introduction

It was almost eight years ago on a balmy Southern California January at the beautiful La Jolla Shores Hotel. And there we were. Indoors. Arguing the merits of assorted remote procedures called APIs: Sun Microsystems' ONC RPC versus everyone elses'. The occasion was an X/Open Systems Management Working Group meeting. The task before us was to select a standard RPC API. With it, we could establish a base platform upon which we could further accept implementations that would form the basis for a standard management platform. But a resolution never came. The "RPC Wars" were too fraught with N.I.H. (Not Invented Here) and things political. By week's end, there was no resolution or progress. The folks from the east coast would have to head back and dig their way out of the airport parking spots that they had to dig into just to park a few days before. The Northridge Earthquake that had thrown me out of my bed the night before the meeting began failed to assert itself as a good omen that things were about to get shaken up. Instead, it was the same old standards process, guided by large companies, politics, and feature creep. (That is, should we boil the oceans or solve a more narrow issue?)

Then, to paraphrase the prophetic bumper sticker, "a miracle happened." A cool little desktop client called a browser led the world to vote Internet. Suddenly, all the wannabe distributed computing standards—OSI, Novell, Microsoft, and DCE—were looking suspect. What's more, the desktop standard Microsoft was suddenly vulnerable.

I am a refugee of the old-style standards process and I couldn't be happier that the Internet is here to stay. Today, there is a new standards mechanism, powered by a 25-year old Internet mechanism, dubbed only a few years ago by Eric Raymond as *open source*. Thanks in part to the open source process, the standards process is now more empowered. W3C.org is a great example. Wisely, they use the open source community to refine and process standards specifications through worldwide implementations. We all benefit when standards-defining activities are forged on the anvil of world application.

Enhydra

Since its introduction as an open-source technology on January 15, 1999, Enhydra has steadily expanded its reach as a favorite set of technologies in the application server market.

The first audience to appreciate its value were the consultants of the world, represented by companies like Tivano in Germany, I-exposure of the US, Plugged In Software of Australia and Digitalsesame of Taiwan. Drawn by Enhydra's elegant and pragmatic architecture, simple development philosophy, and well-organized source code, these consultants, and eventually, developers of corporate IT, have taken Enhydra from its simple application server roots to a broad set of open source technologies for areas Lutris never anticipated, namely voice and wireless development.

This book is about perhaps the most popular of the Enhydra projects—Enhydra XMLC for the development of wired and wireless Web presentations. Enhydra XMLC is powering Web sites powered by open source Enhydra as well as commercial application servers, including BEA WebLogic. As you'll soon read, Enhydra XMLC appeals as much to project managers and presentation architects as it does to presentation developers.

Who Should Read This Book

This book is written for computer professionals, with a special focus on application architects, Java Web application developers, and those who are just ramping up on Java and are excited about immersing themselves into Web application development.

Taking a task view wherever possible, this book is written to support those seeking a more elegant, maintainable, and flexible mechanism for building Web application presentations. While we spend some time introducing the Enhydra application server for those who are new to the topic of application server development, this book is focused primarily on the topic of Enhydra XMLC and how to use it to improve the lifecycle requirements of your Web application.

A modest knowledge of the Java language is assumed, but key supporting topics, such as XML and DOM development, are introduced for those who are new to them. There is even a chapter dedicated to thinking out the requirements of an application based on an application service-provided (ASP) model. For those who are already well-versed in presentation frameworks, a technical overview of XMLC with other presentation technologies is provided.

For those who are curious about Enhydra XMLC and Enhydra in general, we've presented sufficient information to appreciate its value and unique approach to building dynamic, Web-based presentations for browsers, mobile devices, and even good old telephones when taking advantage of freely available voice portal services from TellMe or Voxeo. More importantly, it is also written to demonstrate how to build Web applications and some of the strategies you could employ.

We've also used this opportunity to explain and compare supporting concepts such as XML and HTML, explaining their historical differences as well as their basic reasons for being. The use of a pronounced demonstration application modeled after a conference showfloor that you'd encounter at any major computer show is targeted at the reader who is just embarking on the topic of building Web applications. It's used as a means for exploring the process of how you might integrate different display devices into the overall application.

Enhydra definitely carries a Unix flavor, although the Enhydra Kelp project has done a lot to integrate Enhydra with popular Windows Interactive Design Environments, such as JBuilder and Forte. In order to focus more on the `xmlc` command itself, we have chosen to use RedHat's Cygwin tools to emulate a Unix command line environment for the Windows environment.

Finally, I love new innovations. So, at the end of the book, I leave you with an introduction to Barracuda, which was a lot of fun to discover and write about. The promise of this new technology is incredible.

Lutris Technologies, Steward of Enhydra.org

Enhydra.org, one of the world's most popular and well-known open source efforts, is the result of the efforts of the Java developers at Lutris Technologies. After a great deal of preparation, Lutris introduced Enhydra.org on January 15, 1999. The bootstrapping technology of Enhydra.org was Enhydra, a Java application server, developed by Lutris' consulting organization and used for engagements with many dot coms as well as large IT organizations, including the Kinko's Corporation.

The emergence of the popular Internet has been a blur to most of us. Few are aware that it wasn't until the move by Netscape to define the Mozilla license for the Netscape browser when the term *open source* was coined by Eric Raymond.

Lutris was no stranger to open source. Most of Enhydra's early development was conducted on FreeBSD. Mark Diekhans, the creator of Enhydra XMLC, was well-known in John Ousterhout's Tcl community, having co-authored the TclX extensions, influencing the core Tcl APIs. He and his Lutris colleagues knew what open source was about, what its value was, long before IDC and others started trying to explain this emerging concept to the business community.

Lutris also gives Enhydra its business flavor. Noted earlier, Enhydra was developed in support of professional consulting engagements. One of the business considerations that drove Enhydra's design was to make it easily adaptable to the favorite tools of Java developers.

Today, despite the fact that Lutris is still very much a leading force behind this unique open source effort, Enhydra has a life of its own. Much of the scope of Enhydra.org will be covered in Chapter 1, "Enhydra and XMLC."

Servlet Programming

Enhydra 3 features two types of servlet development. The first type, Enhydra Application Framework (EAF; more commonly referred to as the *superservlet*), was developed by Lutris in advance of the introduction of the standard servlet programming. The other type, of course, is Sun Microsystems' standard servlet development based on servlet 2.2. As we'll explain in Chapter 10, "Servlet Web Applications," you can use Enhydra 3 to develop and generate a WAR file, then deploy it with a few clicks of the mouse in a BEA WebLogic and Lutris EAS 4 enterprise-class application server. Or you can simply download the XMLC-only package from xmlc.enhydra.org and use XMLC as a part of your native WebLogic development environment. With some exceptions, we will be using Enhydra EAF-style servlet programming, characterized by presentation objects. As you'll quickly see, EAF looks very much like standard servlet development. The two styles of programming are so similar that servlet programmers won't be put off.

Organization

This book was written to give you a gut feeling of what Enhydra XMLC is all about. To support this goal, there are numerous notes that explain the background on what motivated certain features of XMLC. If you are new to Web development, then we encourage you to focus on Chapters 1–9, which explain the basics of XMLC, its supporting technologies, and how they relate to other technologies that address presentation development. Any experienced Web developer can pick and choose chapters depending on, for example, her level of experience with DOM programming. The latter third of the book should be interesting for those exploring wireless, voice, and Flash development, as it is driven by data shipped from a back-end application server. We encourage everybody to read the last chapter on Barracuda, a very promising next step in the evolution of XMLC to a full presentation framework.

Chapter 1, "Enhydra and XMLC"—High-level introduction to Enhydra XMLC and the Enhydra.org family of technologies.

Chapter 2, "XMLC Development"—A first introduction to the experience of XMLC development, with a walkthrough of development examples. The concept of DOM development is introduced.

Chapter 3, "Presentation Technologies"—A survey of presentation technologies, including XSLT, JavaServer Pages, Servlet Development, and Cocoon. A comprehensive set of observations on the comparative strengths of XMLC concludes this chapter.

Chapter 4, "The ShowFloor ASP Application"—An application reflecting an application server provider business model is introduced. The concept of UML design is introduced. The rest of the chapter speculates on the potential uses of wireless technology for enhancing the features and attractiveness of this fictitious application.

Chapter 5, "Enhydra, Java/XML Application Server"—A presentation technology is useless without an application server behind it. This chapter introduces Enhydra and its architecture for the development of both superservlets, as well as standard Web application servlets. Enhydra DODS for creating Java object-to-RDBMS mappings is also introduced.

Chapter 6, "XMLC Basics"—The basic elements of XMLC development are introduced, with a focus on comparing and contrasting the XML and HTML markup technologies and how they feed into DOM development. The remainder of the chapter focuses on the value that XMLC brings to DOM development.

Chapter 7, "The xmlc Command"—All of the aspects of the `xmlc` command for generating DOM templates and the features of runtime XMLC programming are addressed here.

Chapter 8, "HTML Presentations"—An introduction to the unique nature of HTML development with XMLC. Template cloning, table construction, and DOM operations are covered. A questionnaire application, VendQuest, is introduced to illustrate template-driven forms and HTML control development.

Chapter 9, "Presentation Strategies"—More complex XMLC development situations are addressed. Strategies for designing and implementing composite views introduce the ability to import nodes and leverage the XMLC `-ssi` option. The VendQuest example is enhanced with the introduction of Zeus XML data binding, another Enhydra.org open source technology.

Chapter 10, "Servlet Web Applications"—A focus on Enhydra XMLC's ability to address standard servlet programming. A WAR built with Enhydra is migrated to both the Lutris EAS and BEA WebLogic J2EE application servers.

Chapter 11, "Wireless Markup Presentations"—Introduces the concepts and strategies behind XMLC's support for WAP's WML language. This chapter also introduces VoiceXML and how XMLC can drive voice presentations through voice portals.

Chapter 12, "Client-Server Development with J2ME and Flash"—Introduces the concepts and capabilities of these smart client presentations, as driven by XML and generated and consumed by Enhydra XMLC. An introduction to using XMLC with the new Scalar Vector Graphics (SVG), the XML standard, is presented.

Chapter 13, "Barracuda Presentation Framework"—An introduction to an impressive new open source presentation framework constructed on top of Enhydra XMLC. Barracuda introduces a Swing-like component view of DOM development.

About OtterPod Productions

OtterPod Productions is this book's model ASP company. The Web site http://www.otterpod.com is real. It is operated by the author.

You can find the following at www.otterpod.com:

- A working version of the ShowFloor application, source code and all.
- Notes and updates relevant to the book.
- Announcements about Enhydra XMLC, as learned by the author.
- Hints/Tips as acquired by the author.
- A page for registering your comments, feedback, and suggestions for the book.

The entire site is a combination of static HTML and dynamic pages powered by open source Enhydra XMLC and the Enhydra Java/XML Application Server.

The www.otterpod.com site is wireless friendly! That's because the index page, `index.po`, is an Enhydra presentation object whose chore it is to detect the device type that is accessing the site. Why access this site with your i-mode phone? That will be a surprise, but a well-formatted surprise delivered in the appropriate markup language for your mobile device.

Conventions and Tools

Unless indicated otherwise, all of the commands illustrated in this book are given in standard Linux/Unix form. What good does this do for those using Windows environments? I recommend downloading the Cygnus tools, a UNIX shell emulation environment.

Cygnus can be downloaded from http://sources.redhat.com/cygwin/.

If you are looking for a Java IDE environment to simplify Enhydra servlet and Enhydra XMLC development, be sure to check out http://kelp.enhydra.org. This open source project provides the tools needed for the integration of Enhydra technology into JBuilder 4 and 5, as well as Sun Microsystems' Forte.

Enhydra 3 Versus Lutris EAS 4

Enhydra XMLC development is enhanced by the inclusion of sub-DOM class implementations that address both HTML and WML. To put their value simply, they support a set of methods that help insulate the developer from raw DOM API programming. Having said that, there's nothing about XMLC programming that requires the use of per-XML language DOMs. If the XMLC compiler can coerce your targeted markup file into a DOM, you're in business. You just need to be comfortable with DOM API development.

If you want to take advantage of the sub-DOMs available for cHTML (the i-mode standard), XHTML (the new W3C.org standard), or VoiceXML 1.0 (also from W3C.org), then you have a choice of either purchasing Lutris Enhydra 3.5 from Lutris Technologies, or using the 45-day evaluation copy of the Lutris EAS 4 J2EE Application Server that's included with this book. This is not a requirement to enjoy the chapters that focus on wireless and voice topics.

Downloads

If, like me, you want to assure yourself of working with the latest and the greatest, everything on the book's CD can be downloaded from a number of locations.

- **http://enhydra.enhydra.org—** The open source home for the Enhydra 3 Java/XML Application Server.
- **http://xmlc.enhydra.org—** The open source home for the Enhydra XMLC.
- **http://barracuda.enhydra.org—** The open source home for Barracuda, a presentation framework built on top of the XMLC layer.
- **http://kxml.enhydra.org—** The open source home for Stefan Haustein's micro XML parser, key to bringing J2ME and Enhydra XMLC together for a client/server relationship.
- **http://zeus.enhydra.org—** The open source home for Zeus, a data binding facility for marshalling and unmarshalling XML into and from Java logic.

The Book's CD

A CD is included with this book to simplify the chore of putting all the elements together that you'll need. All of the examples presented in this book are available on the accompanying CD.

- Open Source Enhydra 3 Application Server
- Open Source XMLC.zip (Standalone portable XMLC environment)
- BEA WebLogic 6.1
- RedHat Cygwin Tools
- Open Source Enhydra Barracuda 1.0
- Lutris EAS 4.1

Be sure to read the `index.html` file for instructions on how to install each component.