Mark Cyzyk. Book Review: Erik Hatcher and Steve Loughran's *Java Development with Ant*, in *Technology Electronic Reviews*, v12 (2), May 2005.

http://www.ala.org/ala/mgrps/divs/lita/ter/volume12no2.cfm#cyzyk

ter
Technology Electronic Reviews
Formerly *Telecommunications Electronic Reviews*

*Volume 12, Number 2, May 2005*

# REVIEW OF: Erik Hatcher and Steve Loughran. (2003). *Java Development with Ant*. Manning: Greenwich, CT.

by Mark Cyzyk

As a long-time web application architect and developer, yet someone who is a novice at Java programming, I must say that server-side Java web application programming and deployment is incredibly complicated, particularly the deployment aspect. Other web application platforms, such as ASP, PHP, Perl, Python, and Cold Fusion, don't really have the same notion of application "deployment" as what holds in the Java world. With these other application platforms one simply places uncompiled files of code on a suitably configured web server, configures whatever data source names (DSNs) may be required for backend database connectivity, and configures other extraneous services, e.g., scheduled jobs, full-text indexing engines, third-party libraries, etc., and the application is ready for use. With Java this is different. Very different.

First, Java requires not only just an external language interpreter for compilation and execution of application code, it requires a full-blown external application server, something as large, complicated, and resource-intensive as a web server. The application server can either run in tandem with a web server or can run on its own and serve up Java applications without any other intervening technology. Secondly, Java is not a scripting language like the languages listed above. Scripting languages are dynamically compiled. Java code, servlets at least, must be pre-compiled into bytecode before they can execute within the bounds of a Java application server. And third, the way this compilation and deployment to the Java application server happens is somewhat complicated, with a fairly complex series of conventions governing it, e.g., a directory tree containing suitably-named folders and files for each individual application. Pre-compiled Java class files must be placed precisely into this directory tree to be properly executable by the parent Java application server.

Ant is a tool that was built to automate compiling and deploying code, not only for Java web applications, but for standalone Java software as well. As I learn more about Ant, and read this nicely written book by Erik Hatcher and Steve Loughran, co-committers on the open source Ant project, I've come to think of Ant as a large batch file or system macro interpreter.

Ant works by reading in a structured XML file, build.xml, and responding to the commands ("directives") contained within./p>

As someone who is new to Java web application development and deployment, and someone who is really not at a point to be capable of generating custom, from scratch Ant build files, I found the three chapters particularly useful in bringing me quickly to an understanding of how it all works; Chapter 7: Deployment, Chapter 12: Developing for the Web and Chapter 18: Production Deployment.

Chapter 7: Deployment illustrates how to deploy a Java application using Ant in four different scenarios: FTP-based distribution of a packaged application; Email-based distribution of a packaged application; Local deployment to Tomcat; and Remote deployment to Tomcat.

Copying a directory tree to a remote server via FTP seems, at first, to be a pretty easy task to do programmatically, until, that is, you stop and think about everything that could go wrong. Here the authors focus on how to connect to an FTP server using the directive, how to first determine whether that server is even available using the directive, and how to pause the build process at appropriate times using the and directives.

Distributing a zipped up application via email is straightforward. The directive contains all the email-related attributes you would expect, e.g., "from"; "bcclist"; "mailhost"; "subject".

The heart of this chapter, however, is deploying web applications to local and remote Tomcat servers. Here the authors provide nice examples of using the Ant directive to install either a single WAR file into its appropriate location in the Tomcat directory tree, or alternatively using the directive to do likewise on a remote Tomcat installation.

Chapter 12: Developing for the Web provides a nice illustration and explanation of how server-side Java web applications are structured, the directory conventions used and meta-data files required for proper deployment of a web application to a Java servlet container. The chapter then delves deeper into automated pre-compilation of Java Server Pages (JSP), including JSPs that rely on tag libraries, generation of "deployment descriptor" files, i.e., small XML files that contain metadata about the web application, and using HttpUnit or Cactus (two open-source projects) to automatically test the deployment.

Chapter 18: Production Deployment tied everything together for me. The authors briefly address the challenges of deployment to different application servers/servlet containers, each of which may have slightly different implementations of official specifications, differing demands for underlying versions of Java, etc. The authors then proceed to locate and define how and where in the development-to-production software engineering process Ant fits in. The thought here is that, in a production environment, responsibility for the software is passed from developers and a test system to operators and a bona fide production environment. Ant can and should be used to streamline this movement of code out of development and testing and into full-blown production. The rest of the chapter provides a nicely detailed illustration of a sample Ant build and deploy scenario, moving code from test to development using Ant's various "Power Tools." It concludes with suggested techniques that could be used to verify that the build and deployment processes were successful. Again, this chapter provided, for me, a nice finale to the whole Ant deployment process, despite the fact that it is only about two-thirds of the way through this encyclopaedic book!

Although I have focused this review on these three chapters and on web application deployment techniques, both Ant and this book include much more. Other topics and capabilities of Ant broached in the book include:

- Automated code testing with JUnit
- Executing external programs
- XDoclet (which appears to be some sort of code generator for something called "Attribute-Oriented Programming" in Java)
- Ant and XML data
- Ant and Enterprise Java Beans (EJB) development
- Ant and Web Services
- And a set of very useful appendices, including one addressing the integration of Ant with your local IDE.

As with the O'Reilly books, technical books published by Manning are in general excellent. I've read and reviewed Manning publications in the past, and have always enjoyed their chatty style and superb typography and layout, something crucial to the usefulness of any technical book, in my opinion.

This book is recommended for the experienced Java programmer as well as for the novice wishing to tweak a pre-existing build file or simply attempting to gain an understanding of the culture and conventions of Java application programming and deployment.

*Mark Cyzyk, formerly Head of Information Technology in the Albert S. Cook Library at Towson University, is the Web Architect at the Johns Hopkins University in Baltimore, Maryland.*

## About *TER*

Technology Electronic Reviews (TER) is an irregular electronic serial publication of the Library and Information Technology Association, a division of the American Library Association, 50 E. Huron St., Chicago, IL 60611. The primary function of TER is to provide reviews of and pointers to a variety of print and electronic resources about information technology. Resources include books, articles, serials, discussion lists, training materials, bibliographies, and other items of interest to librarians and information technology professionals. The topics covered may include, but are not limited to, networking technologies and standards; hardware and software; operating systems; databases; specific programming languages; management tools and utilities; technical project management; training and personnel issues; library perspectives; and research and development.

TER is distributed electronically via Internet. There is no subscription fee. Currently it is available via World Wide Web (http://www.lita.org/ter/) and new-issue announcements are posted on the LITA-L electronic discussion list. To subscribe, send an email message to listproc@ala1.ala.org that says: subscribe LITA-L First-Name Last-Name. Other distribution arrangements may be made in the future.