

ter

telecommunications electronic reviews

Volume 5, Issue 6, July 1998

Telecommunications Electronic Reviews (TER) is a publication of the Library and Information Technology Association.

Telecommunications Electronic Reviews (ISSN: 1075-9972) is a periodical copyright © 1998 by the American Library Association. Documents in this issue, subject to copyright by the American Library Association or by the authors of the documents, may be reproduced for noncommercial, educational, or scientific purposes granted by Sections 107 and 108 of the Copyright Revision Act of 1976, provided that the copyright statement and source for that material are clearly acknowledged and that the material is reproduced without alteration. None of these documents may be reproduced or adapted for commercial distribution without the prior written permission of the designated copyright holder for the specific documents.

Contents:

- REVIEW OF: Julie M. Still, ed. *The Library Web: Case Studies in Web Site Creation and Implementation*. by Matthew M. Benzing
- REVIEW OF: Nate Patwardhan, Clay Irving, Brian Jepson, Ellen Siever, David Futato, and Larry Wall. *Perl Resource Kit - UNIX Edition*. by Ray Olszewski
- About TER

 [ter issues \(/lita/publications/archive/ter\)](/lita/publications/archive/ter)

REVIEW OF: Julie M. Still, ed. *The Library Web: Case Studies in Web Site Creation and Implementation*. Medford, NJ: Information Today, 1997.

by Matthew M. Benzing

Some librarians become involved in Web design out of personal and professional interest; some have Webmastering thrust upon them. This book reviews Web creation projects at a number of academic, public, and special libraries; it provides readers with not only practical advice for creating a library Web presence,

but also a picture of the ways in which the Web has impacted library services.

One of the more interesting aspects of this collection is its description of the various motivations that drive the development of a library Web presence. Some libraries are interested in the Web's usefulness as a marketing tool; others just wanted to provide a common interface to a number of online products. Some libraries see their Web pages as a way to coordinate Internet collections of relevant sites, while others see it as a way to publish their own unique material. The catalyst of the Web has sparked numerous different reactions.

The University of Saskatchewan Library U-SEARCH project is an example of one approach. Douglas Fox, head of systems for the library, describes how in 1994 his library designed not only a Web page, but also a Lynx-type browser expressly for the purpose of providing access to all library resources through a single interface. As any reference librarian who has had to patiently guide a frazzled patron through multiple and confusing search screens on multiple machines will attest, this is a worthy goal. Later, as Netscape seemed to be winning the browser wars, U-SEARCH was adapted to work on this widely available platform. Fox logically and coherently describes the evolution of his library's Web site and gives insight into why libraries should be on the Web. Technical issues are dealt with cleanly; the presentation reflects this library's levelheaded approach to creating a Web presence.

Judith Edwards' experiences at the University College London Library differ markedly. In this case the decision to go online came from on high, as the library was directed to come up with a Gopher and then Web page to integrate with the Campus-Wide Information System that was being administered by the Information Systems Division. Edwards had little technical training and had to teach herself HTML (Hypertext Markup Language) and other skills in order to keep up.

Whereas Saskatchewan benefited from a technically savvy systems librarian working with a team, the UCL site was thrown up in a "quick and dirty" way by a one-woman show with skill learned on the job. Edwards' account gives another perspective on the growth of Web services in libraries: that of the many librarians who do not have technical backgrounds but who are forced to pick up Web design skills on the fly. Edwards' description of her evolution from assistant reference librarian to Webmaster is well told; her growing confidence in networking technology is palpable.

While both of the previously-described libraries set out to create Web pages in order to provide better access to information, many libraries also see the Web as a means of gathering support and of creating the impression of technological relevance. The University of Southern Mississippi created its Web site with four factors in mind: on-campus public relations, off-campus public relations, staff interest, and availability of equipment. Creating broader or more efficient access to information is not listed as a concern.

This is probably typical of many libraries' first foray into cyberspace. Unfortunately, this lack of focus on appropriate goals often results in ill-designed Web sites; however, in the case of the University of Southern Mississippi the library benefited from the work of a recent library school graduate who had experience with Web site design and maintenance. David King, who developed both the Web site and the article describing its creation, describes the process of planning and implementing a Web site clearly and with an eye to the difficulties that a novice might have.

The Library Web also includes discussion of librarians' experiences with using handouts on the Web, Web-based library instruction, and other case studies. The main drawback to this book as a practical guide to Web creation for libraries is that it is dated; how many libraries are just now creating their own Web sites? On the other hand, as a snapshot of the early attempts of libraries to find uses for the Web it is a valuable

guide to the motivations and methods with which libraries adopt any new piece of technology. The writing skill of the contributors varies greatly, but most essays are clear and concise while accurately reflecting the variety of temperaments and talents that make up the world of library Webmasters.

Matthew M. Benzing (benzim@rpi.edu (mailto:benzim@rpi.edu)) is Information Systems Librarian at Rensselaer Polytechnic Institute.

Copyright © 1998 by the American Library Association. This document may be reproduced in whole or in part for noncommercial, educational, or scientific purposes, provided that the preceding copyright statement and source are clearly acknowledged. All other rights are reserved. For permission to reproduce or adapt this document or any part of it for commercial distribution, address requests to Office of Rights and Permissions, 50 East Huron Street, Chicago, IL 60611.

[table of contents](#)  [ter issues \(/lita/publications/archive/ter\)](#)

REVIEW OF: Nate Patwardhan, Clay Irving, Brian Jepson, Ellen Siever, David Futato, and Larry Wall. Perl Resource Kit - UNIX Edition. Four volumes. Sebastopol, CA: O'Reilly & Associates, 1997.

by Ray Olszewski

Perl is a computer programming language. Originally developed as a tool to automate systems administration tasks on Unix servers and workstations, it has grown to become a full-service programming tool. It combines the versatility of the best applications programming languages, such as C and C++, with uncommonly strong components for file processing and pattern matching. Perl is available for all major operating systems, including Unix, Windows 95 and NT, the Macintosh OS, and IBM's OS/2.

Its main use, however, remains in the Unix world, where it is the language of choice for systems administration, "glue" code that links other applications, and back-end programming for Internet services (I've seen it called "the duct tape of the Internet"). Several major Internet applications, such as majordomo (a widely-used system that manages automated email discussion groups) are written entirely in Perl. In my own work as an applications programmer, I have written one major (proprietary) application in Perl, along with too many system management tools to count. Perl's pattern-matching tools make it a particularly convenient way to write programs that process complex data inputs, such as system log files, CGI (Common Gateway Interface) forms on the Web, and small databases.

The name Perl is an acronym; officially, it stands for Practical Extraction and Report Language, though one sees many variants on this name. Perl is also unusual in that it is the only major programming language that developed entirely as freeware. To this day, Perl interpreters, libraries, modules, and application examples are available without charge for every operating system on which Perl runs. The complete Perl archive is available online (<http://www.perl.com/> (<http://www.perl.com/>), a site supported by O'Reilly & Associates, and at many other mirror sites) and on CD-ROM (for example, from Walnut Creek CD-ROM, <http://www.cdrom.com/> (<http://www.cdrom.com/>)).

The publisher O'Reilly & Associates has played a major role in the growth of Perl, publishing several important reference and learning tools, helping to maintain the Internet archives of Perl resources, and sponsoring conferences and tutorial sessions on the language. Perl Resource Kit is O'Reilly's latest

contribution to the material on Perl. The present Unix edition provides direct support for the Solaris (Sun Microsystems) and Linux (a freeware package) versions of Unix and indirect support for other Unix variants. A version for Windows 95 and NT is also available.

The kit consists of five main components:

Programming with Perl Modules (Nate Patwardhan with Clay Irving), an introduction to the use of specialized modules to extend the capabilities of Perl;

Perl Utilities Guide (Brian Jepson), an overview of the kit and detailed instructions in the use of the JPL software package (see below);

Perl Module Reference (Ellen Siever and David Futato; 2 volumes), one volume of which provides detailed documentation of most of the modules in the CPAN (Comprehensive Perl Archive Network) archive, with the second volume including an October 1997 "snapshot" of the CPAN archive, a collection of Perl modules and sample applications created by dozens of programmers; and

JPL (Java Perl Interface), a software package that integrates Perl with Java, the well-known programming language originated by Sun Microsystems and now available for a wide variety of platforms.

It also includes source code and executables (for Solaris and Linux) of version 5.00404 of Perl, the version current as of the kit's October 1997 publication date.

Taken as a whole, the kit is a valuable addition to the support available for Perl. Looked at one by one, all the components are useful, though their values vary.

Programming with Perl Modules

Programming with Perl Modules provides an overview of how to use Perl modules, along with a detailed introduction to several of the more important sets of modules. Modules in Perl are similar in purpose to C libraries in that they provide sets of standardized routines a programmer can call upon to provide capabilities not included in the basic Perl distribution.

This introduction emphasizes modules that assist Perl programmers in developing Internet applications. Modules that provide direct support for Internet services--such as email, Telnet, FTP, Usenet news (Network News Transfer Protocol or NNTP), CGI script support, and Web services--are of course important here. But so are the modules that let Perl programmers create the GIF (Graphics Interchange Format) graphics that are the standard for display on Web pages and the modules that provide an interface to standard database packages.

While coverage of these areas is not exhaustive, it is intelligently selective, focusing on useful, up-to-date modules such as GD (graphics), the many components of the Net (Internet services) package of modules, and interfaces to SQL-compliant (Structured Query Language) database packages. Other modules featured here support Unix's graphical interface (X Window) using the Tk toolkit, filesystem access and management, and date and time.

To test this introduction to modules, I followed its instructions for a module I already use a lot (GD, the graphics module) and for one I wanted to learn (Net::FTP, which supports automated file transfers between Internet hosts). In both cases, I was pleased with the coverage. In the case of the Net::FTP module in particular, I was quickly able to adapt their example to use on an Internet host I had access to. After

correcting a small error in the example code, I was readily able to modify it to become a component in an application I was developing. I had similar good experiences with Net::Telnet and Net::Ping, other modules that provide access to Internet services.

The authors end with a brief introduction to the process of creating modules and a few extended examples of applications using the featured modules. The discussion of the procedure for contribution modules to CPAN is well described, but the short chapter only scratches the surface of the process involved in the actual programming of a module. The extended examples are useful illustrations of the process of creating real, purposeful programs (not mere teaching examples) that draw on many of the modules discussed.

While the introduction is good, it is not without its flaws. Most serious is the inclusion of occasional errors in the examples, errors distinct enough that the code could not have run as intended. Additionally, some of the modules require installation of other modules to run--the Net::FTP package, for example, requires two others--but the descriptions of the modules do not identify these dependencies.

Overall, though, this volume is a model of what software documentation should look like. It is well written, rich with both short and extended examples, and clear and direct in its presentation. While there is always room for improvement, Programming with Perl Modules earns high marks for its intelligent selectivity and clarity.

Perl Module Reference

Perl Module Reference collects in print form the documentation associated with most of the modules available from CPAN. Based on spot comparisons, there appears to be little added here beyond formatting and packaging; the text of the module documentation seems identical in content to the documentation files that are included in the archives for each package.

While the documentation is essentially unedited except for format, having it in print form is a real value. The formatting is good, and hard copy is easier (for me, at least) to read and annotate than online documentation, especially the mishmash of "pod," HTML, and man-page format files that the online versions of module documentation actually come in.

Compared to the prior volume, the virtue of this two-book set is its near completeness. Puzzling is the occasional omission of a module. For example, I could find no documentation for the BigInt module (used to do integer arithmetic on numbers larger than the normal maximum built into Perl), even though it is one of the "core modules" included with most Perl distributions.

Perl Utilities Guide and JPL

This part of the kit proved the most difficult for me to review. Although I am an experienced Perl programmer and applications developer, I am a beginner at Java--just another member of the faceless army of programmers who "know some Java" but have not written any substantial applications in it.

JPL is a toolkit for creating applications that combine Perl and Java code. Written by Larry Wall, who also created Perl, this is a commercial product developed specifically for the Perl Resource Kit. To run, it requires a host computer that runs both Perl 5.x and JDK (Java Development Kit) 1.1. A workstation with the X Window graphical interface is desirable. With these facilities, you can use JPL to compile applications that combine Java and Perl code into applications that run on your Unix computer. Add an Internet connection and the RMI (Remote Method Invocation) capabilities of Java, and you can write applets that run standard Java on a browser but connect to the Unix server to run Perl components.

What does all this mean for the Java programmer? Mainly, it offers the ability to extend Java by "borrowing" the capabilities of Perl that Java lacks. Generally speaking, these capabilities are of two sorts.

Flexible processing of text: Perl was originally written to simplify systems administration tasks, many of which involve processing logs. As a result, Perl excels in its ability to find patterns in lines of text, do search-and-replace, parse strings into components, and perform similar activities. Perl's strengths here are distinct, perhaps unique.

Extensibility: like many programming languages, Perl allows for easy extension through its system of modules. Java's commitment to machine independence, in contrast, limits one's ability to add specialized libraries of functions to it, especially in connection with applets, the Java programs designed to run in Web browsers. Access to Perl modules facilitates a range of programming tasks.

Examples that come with the JPL package illustrate using Perl components to manipulate text, expedite the production of graphs and other images, and access Internet services.

What does all of this mean for the Perl programmer? For those of us who do move to Java, JPL will allow us to transfer some of our experience to the new programming environment. They allow us access to the Java interface components, perhaps the best features of Java (and among the worst of Perl). The JPL examples offer convincing evidence that combining the strengths of the two languages can allow for quick development of serious, valuable programs.

Despite this, I doubt that JPL will catch on outside a small niche. Other mechanisms are available to let Java and Perl applications communicate--sockets and CGI scripts are two that come quickly to mind--and the attempt to establish PerlScript (a way to let Web browsers execute a subset of the Perl language) may prove successful. While this package offers some features that will be broadly useful to programmers--experience using RMI is particularly attractive--I would hesitate to invest much time mastering the intricacies of this particular approach to integration.

The Perl Utilities Guide also provides instructions for using the CD-ROM. For the most part, use of the CD-ROM is self-evident to readers in the target audience, but the instructions here are clear and direct.

One significant limitation related to CD-ROM installation is that the setup program included on it relies on the X Window graphical interface. While this is not a serious limitation for users of Unix workstations, it is for those of us who access Linux servers remotely, or from a PC running Windows 95 or some other operating system/user interface. In my experience, these servers often do not support X Window, making the tool unavailable. Still, the fallback procedures for installation without the interface do work and are not particularly burdensome to anyone accustomed to working from the Unix command line.

The CPAN "Snapshot"

The CPAN (Comprehensive Perl Archive Network) archive is a collection of Perl modules and sample applications created by dozens of programmers. For Perl developers, it is the essential set of extensions to Perl, a set that includes code to provide network services, interface with other applications (such as databases), create graphs and other images, handle advanced mathematics, and more. At about 400 megabytes of code (uncompressed), CPAN is too large for almost any site to install in its entirety. For that reason, it distinguishes a smaller number of "core modules" from its larger collection.

Each module is provided as a "*.tar.gz" file, requiring use of the standard Unix packages gzip and tar to expand the archive into a collection of files. A typical distribution includes source code in C, the .pm header code needed by Perl, a "Makefile" that controls installation of the package on a Unix computer, and package

documentation. For users familiar with Unix, installation is routine. Many of the newer packages use the Perl "MakeMaker" preprocessor to create the needed Unix Makefile; the added step seems to make installation smoother overall.

For the most part, the strengths and weaknesses of this CD-ROM are those of the underlying CPAN collection. The modules themselves are of high quality--they're astoundingly consistent in quality given that almost all are freeware and the archive itself is essentially a labor of love for its operators. Documentation varies from crystal clarity to troubling obscurity. The installation routines for the half-dozen packages I installed ran flawlessly.

In all, CPAN is a vital resource for any site that wants to support serious applications development in Perl. While this CD-ROM delivers CPAN in convenient form, though, it does have a couple of problems.

One serious limitation of the CD-ROM is that its filesystem works properly only under Unix. Although this might not seem like a problem for a collection of Unix software, it actually can be a significant limitation for many potential users. I, like many users of Unix, access servers remotely; to test some of the networking-oriented modules, I needed to install them on a remote server (physically located about 30 miles from my home office) through my Windows 95-based client. The filesystem incompatibility--particularly the fact that Unix symbolic links do not work under Windows 95--made it impossible to upload the files directly from the CD-ROM. Instead, I had to load the CD-ROM on a nearby (non-networked) Linux host, copy the needed modules to floppies, then use the floppies to FTP the modules to the remote server. Other CD-ROMs that distribute the CPAN archive (for example, the collection available from Walnut Creek CD-ROM) do not share this limitation.

A second problem is common to all CD-ROM collections: they get old and out of date. Although this CPAN snapshot was reasonably fresh at the time of publication, it is already out of date. Unless O'Reilly plans to update the CD-ROM component of the package regularly, I cannot recommend relying on it as a CPAN source for very long, especially given that the up-to-date version of the archive is on the Web at many locations (e.g., <ftp://ftp.digital.com/pub/plan/perl/CPAN/> (<ftp://ftp.digital.com/pub/plan/perl/CPAN/>)) and less expensive CD-ROM versions of CPAN are issued regularly by Walnut Creek and others.

Other Odds and Ends

The Perl Resource Kit includes a few other treats that are hard to categorize. You get a sample issue (Fall 1997) of The Perl Journal, a quarterly magazine that supports Perl programming, the non-Unix pieces of CPAN (mainly versions of Perl for other operating systems), many functioning Perl programs from the CPAN "scripts" archive, and an assortment of O'Reilly promotional materials. These incidentals are generally useful but incidental to the package and its purpose.

Conclusion

Perl Resource Kit is a useful addition to the toolkit of every serious Perl programmer. It is also worth a look by system administrators of Unix sites that support significant Perl development work. It is not for the beginner, though--it includes no real tutorial component (except for the specialized JPL package), and beginners would do better to work at a site that already has support for at least the best of the CPAN modules in place.

The JPL component makes the kit a good buy for Java programmers as well, who will appreciate getting access to Perl's text-manipulation features and the many features available through Perl modules. Perl programmers will similarly benefit from access to the Java user interface. But programmers who hope to use

JPL as a transition to Java will be disappointed--JPL assumes the programmer already is well versed in Java as well as Perl.

Ray Olszewski (ray@comarre.com (mailto:ray@comarre.com)) is a consulting economist and statistician. He spent three years as Network Manager at The Nueva School, a private K-8 school in Hillsborough, California. His work includes development of custom Web-based software to support online research.

Copyright © 1998 by Ray Olszewski. This document may be reproduced in whole or in part for noncommercial, educational, or scientific purposes, provided that the preceding copyright statement and source are clearly acknowledged. All other rights are reserved. For permission to reproduce or adapt this document or any part of it for commercial distribution, address requests to the author at ray@comarre.com (mailto:ray@comarre.com).

[table of contents](#)  ter issues (/lita/publications/archive/ter)

About TER