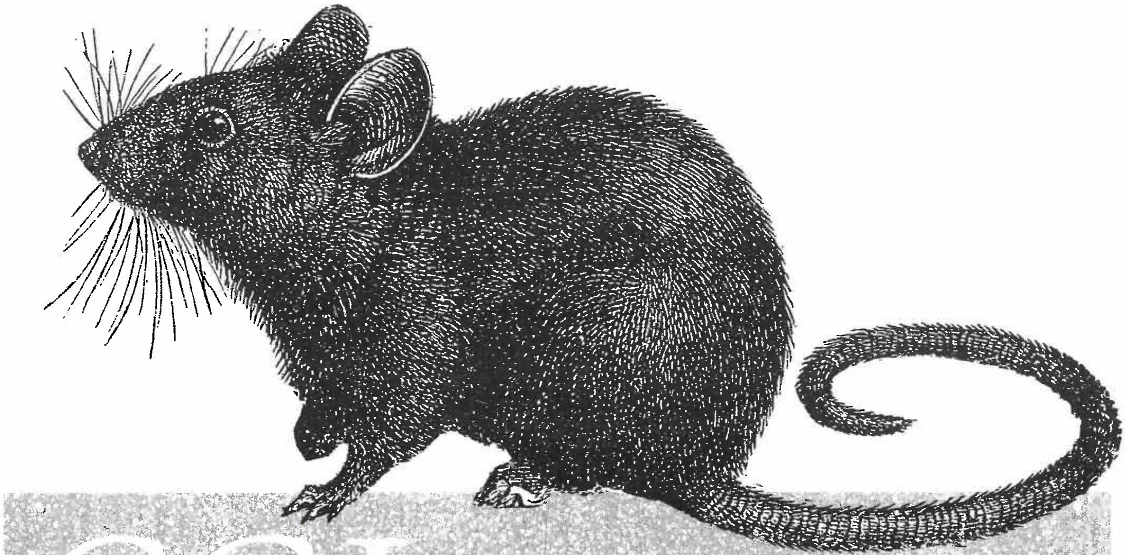
On-the-Spot Information

Examples in Perl



CGI Programming

on the World Wide Web

O'REILLY™

Shishir Gundavaram

CGI Programming
on the World Wide Web

Shishir Gundavaram

O'Reilly & Associates, Inc.
Cambridge • Köln • Paris • Sebastopol • Tokyo

CGI Programming on the World Wide Web

by Shishir Gundavaram

Published by O'Reilly & Associates, Inc., 101 Morris Street, Sebastopol, CA 95472.

Editors: Andy Oram and Linda Mui

Production Editor: Jane Ellin

Printing History:

March 1996: First Edition

Nutshell Handbook and the Nutshell Handbook logo are registered trademarks and the Java Series is a trademark of O'Reilly & Associates, Inc.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and O'Reilly & Associates, Inc. was aware of a trademark claim, the designations have been printed in caps or initial caps.

While every precaution has been taken in the preparation of this book, the publisher assumes no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

This book is printed on acid-free paper with 85% recycled content, 15% post-consumer waste. O'Reilly & Associates is committed to using paper with the highest recycled content available consistent with high quality.

ISBN: 1-56592-168-2 [1/98]

- ArchiePlex Gateway

  A gateway to the Archie search server. Allows the user to search for a specific string and returns a virtual hypertext document. This useful gateway is located at *http://pubweb.nexor.co.uk/public/archie/archieplex/archieplex.html.* A simple Archie gateway is presented in Chapter 10.

- Guestbook with World Map

  A guestbook is a forms-based application that allows users to leave messages for everyone to see. Though there are numerous guestbooks on the Web, this is one of the best. You can access it at *http://www.cosy.sbg.ac.at/rec/guestbook.*

- Japanese <-> English Dictionary

  A sophisticated CGI program that queries the user for an English word, and returns a virtual document with graphic images of an equivalent Japanese word, or vice versa. It can be accessed at *http://www.wg.omron.co.jp/cgi-bin/j-e?SASE=jfiedl.html* or at *http://enterprise.ic.gc.ca/cgi-bin/j-e.*

Although most of these documents are curiosities, they illustrate the powerful aspects of CGI. The interface allows for the creation of highly effective virtual documents using forms and gateways.

Internal Workings of CGI

So how does the whole interface work? Most servers expect CGI programs and scripts to reside in a special directory, usually called *cgi-bin*, and/or to have a certain file extension. (These configuration parameters are discussed in the "Configuring the Server" section in this chapter.) When a user opens a URL associated with a CGI program, the client sends a request to the server asking for the file.

For the most part, the request for a CGI program looks the same as it does for all Web documents. The difference is that when a server recognizes that the address being requested is a CGI program, the server does not return the file contents verbatim. Instead, the server tries to execute the program. Here is what a sample client request might look like:

```
GET /cgi-bin/welcome.pl HTTP/1.0
Accept: www/source
Accept: text/html
Accept: image/gif
User-Agent: Lynx/2.4 libwww/2.14
From: shishir@bu.edu
```

This GET request identifies the file to retrieve as */cgi-bin/welcome.pl*. Since the server is configured to recognize all files in the *cgi-bin* directory tree as CGI programs, it understands that it should execute the program instead of relaying it directly to the browser. The string HTTP/1.0 identifies the communication protocol to use.

The client request also passes the data formats it can accept (www/source, text/html, and image/gif), identifies itself as a Lynx client, and sends user information. All this information is made available to the CGI program, along with additional information from the server.

The way that CGI programs get their input depends on the server and on the native operating system. On a UNIX system, CGI programs get their input from standard input (STDIN) and from UNIX environment variables. These variables store such information as the input search string (in the case of a form), the format of the input, the length of the input (in bytes), the remote host and user passing the input, and other client information. They also store the server name, the communication protocol, and the name of the software running the server.

Once the CGI program starts running, it can either create and output a new document, or provide the URL to an existing one. On UNIX, programs send their output to standard output (STDOUT) as a data stream. The data stream consists of two parts. The first part is either a full or partial HTTP header that (at minimum) describes what format the returned data is in (e.g., HTML, plain text, GIF, etc.). A blank line signifies the end of the header section. The second part is the body, which contains the data conforming to the format type reflected in the header. The body is not modified or interpreted by the server in any way.

A CGI program can choose to send the newly created data directly to the client or to send it indirectly through the server. If the output consists of a complete HTTP header, the data is sent directly to the client without server modification. (It's actually a little more complicated than this, as we will discuss in Chapter 3, *Output from the Common Gateway Interface.*) Or, as is usually the case, the output is sent to the server as a data stream. The server is then responsible for adding the complete header information and using the HTTP protocol to transfer the data to the client.

Here is the sample output of a program generating an HTML virtual document, with the complete HTTP header:

```
HTTP/1.0 200 OK
Date: Thursday, 22-February-96 08:28:00 GMT
Server: NCSA/1.4.2
MIME-version: 1.0
Content-type: text/html
Content-length: 2000

<HTML>
<HEAD><TITLE>Welcome to Shishir's WWW Server!</TITLE></HEAD>
```

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | sales@docketalarm.com | 1-866-77-FASTCASE

fastcase®
Smarter legal research.