Operate your own World-Wide Web server

WILLIAM C. SCHMIDT, RON HOFFMAN, and JOHN MacDONALD
Dalhousie University, Halifax, Nova Scotia, Canada

Although many researchers wishing to use the World-Wide Web for academic purposes rely on centralized Web services, they should be aware that it is neither expensive nor difficult to operate their own server. Doing so provides research-related benefits such as complete control over their host name and documents provided, the guaranteed ability to execute common gateway interface and server-side include programs, immediate access to their collected data, and the ability to better control who participates in their experiments. This paper surveys Web-server software features likely to be of interest to psychologists and conceptually summarizes their operation and use. The basic steps required to set up a Web server on popular microcomputers are reviewed, and security issues concerning Web-server operation are discussed. An accompanying resource Web page can assist users in setting up their own servers.

Numerous applications of Web technology promise to assist psychologists. A wide range of uses for psychology have already been promoted; these include using the World-Wide Web (WWW) as an instructional aid (Welch & Krantz, 1996) and presenting lab and departmental Web pages (Krantz, 1995). Some of the more attractive uses of the WWW for psychological research involve interaction with remote participants for the purpose of gathering data in experiments conducted via the Web. Work along this line has already been reported for psychoacoustical experiments (Welch & Krantz, 1996) and for survey research (Schmidt, 1997). Other research and teaching applications are sure to follow.

This paper provides researchers who wish to conduct experiments via the Web with a conceptual-level description of the operation of a Web server and familiarizes the reader with features of servers that may be of use in creating well-implemented experiments. Specific Web-server software is not discussed (due to the rapidity with which details associated with such software change and the variations that exist across computer platforms). However, a Web page has been created (<http://or.psychology.dal.ca/~wcs/WebServerLinks.html>) to help users get started in finding server software that satisfies their goals and is appropriate for their machines. This resource page also includes rudimentary directions for setting up a server on a number of popular platforms and provides links to other Web-server resource pages.

After discussing the basic concepts underlying the client/server relationship, a number of software features provided by many Web-server programs will be presented.

These include the ability to use common gateway interface (CGI) programs, server-side includes (SSI), and image maps. The benefits of operating your own Web server, as opposed to entrusting this task to others, will then be examined and security issues pertaining to server operation discussed.

THE CLIENT/SERVER RELATIONSHIP

Figure 1 depicts a small portion of the Web. A client is a computer that runs a Web browser, requests information from a Web server, and displays received information to the user according to a predefined convention (hypertext markup language, or HTML).

Within HTML code, links may be found. Such links provide URLs (universal resource locator—a specially formatted address of a document on the Web) to other Web documents or programs, and following such links presents the user with the content referenced by these. Because of the versatility of the URL, any HTML document can reference any document or program on other Web servers anywhere on the Web.

Web-server software runs in the background on the host computer, waiting for document requests to come from various clients. The server responds to such requests by sending HTML-formatted information to the client. The specific details of such requests conform to the hypertext transfer protocol (HTTP) standard. Clients send requests to receive the contents of a file containing HTML (which is then displayed) or the output of a program referenced by the URL (the so-called CGI program, the output of which is displayed).

WEB-SERVER FEATURES

Different server programs do not have identical capabilities. For this reason, it is important for experimenters to understand precisely what their server software is capable of doing. Understanding these server features can

---

The writing of this paper was supported by NSERC Canada and the Izzak Walton Killam Foundation. A page maintained by the first author containing all of the links to the resources referenced in this paper, as well as topically related links, can be found at <http://or.psychology.dal.ca/~wcs/WebServerLinks.html>. W. C. Schmidt's address is: Department of Psychology, Dalhousie University, Halifax, NS, Canada B3H 4J1 (e-mail: wcs@or.psychology.dal.ca).

Figure 1. The client/server relationship.

assist experimenters in overcoming many obstacles to successful data collection by taking appropriate measures in the implementation of their experiments.

Common Gateway Interface (CGI) Programs

The CGI program is the most powerful component behind experiments conducted via the Web, and much useful information has already been reported on them (see Kieley, 1996). These programs enable one to dynamically use information supplied by the user of the client computer, to transform that information, and to supply

specific information back to the user. The capabilities of the CGI are limited only by programmers, their imagination, and the limitations inherent in HTML. Whenever a user clicks on a button, for instance, a CGI program is called to process the information submitted.

Whenever a CGI program is run, privileged information about the HTTP transaction or the environments of both the host and client computers becomes available to the CGI program. Inferences based on such information can help determine actions to take in experimental delivery. Table 1 lists some environmental information available to CGI programs and a few ways that such information could be used by designers of psychology experiments to improve the integrity of the data collected. For a complete list of environmental information available to CGI programs, consult <http://hoohoo.ncsa.uiuc.edu/cgi/env.html>.

To be able to run CGIs at all, the experimenter needs to be using a Web server that is able to execute CGI programs, and must have permission to install CGI programs on the host. Not all servers have CGI capabilities, particularly some free or inexpensive server applications.

Server-Side Includes (SSI)

Just as CGI programs add the ability for a Web server to dynamically deliver information in response to a user's actions (i.e., clicking on a button), SSIs add the ability to execute programs or other directives, but at the time of document delivery and without any actions on the part of the user. SSIs allow the Web server to deliver documents with tailored information to the client. Built-in instructions or specially written programs similar to CGIs are

Table 1
Environment Variables Available to CGI and SSI Programs

Variable Name	Description	Potential Use
HTTP_REFERER	-the URL of the document referencing the CGI on host	-ensure that only clients using authentic documents are sending data to your CGI
HTTP_USER_AGENT	-the browser the client is using to send the request -contains not only browser name, but version and system the user is running (i.e., Windows version, MacOS CPU)	-knowledge of the user's browser can assist in sending properly formatted HTML (i.e., lynx text browser versus graphical browsers)
REMOTE_ADDR	-the IP address of the remote computer hosting the client that is making the request	-track who is sending information to your CGI program, or whom your SSI is sending information to
REMOTE_HOST	-if available, the name of the REMOTE_ADDR computer -if unavailable, same as REMOTE_ADDR	-track who is sending information to your CGI program, or whom your SSI is sending information to
SERVER_NAME	-the server's hostname	-may be used by self-referencing URLs
AUTH_TYPE	-if server supports password protection, then this variable holds protocol	-useful to know how to decode user/password information
REMOTE_USER	-if server supports password protection, and the script is protected, this is the username approved	-useful to access identity of user from within CGI/SSI
SCRIPT_NAME	-holds virtual path to the script	-may be used by self-referencing URLs
PATH_INFO	-URLs may contain additional info beyond the SCRIPT_NAME -PATH_INFO holds this raw information	-allows extra information to be passed in the URL request

executed by the Web server when they are encountered within an HTML document and are expanded inline. That is, before delivering information to the client, the server software filters each document requested, looking for specific keywords that instruct it to carry out SSI tasks. The extra information included can be dynamically computed at the time that the document is delivered, allowing specialized or up-to-date information to be delivered to targeted users.

The SSI Web-server component fits into the server-client relationship on the server side (see the SSI loop of Computer B in Figure 1). The client first makes a document request. This document is fetched and examined by the Web-server software for included instructions to the server. Any instructions found are executed and replaced by the HTML output of the SSI instruction on the server side, all before the client is sent any information. Once the HTML code to be sent to the client is assembled, the server transmits this information. It is unlikely that the client will distinguish between having been sent a dynamic SSI document or a regular static HTML document.

With the use of SSIs, specialized documents can be generated on the fly to satisfy certain goals of the experimenter. For example, suppose that the server administrator is conducting a survey, and wishes to present a differently formatted page to users of text-based versus graphics-based browsers. SSI programs have access to all of the environment information that CGI programs do (see Table 1), as well as to extra pieces of information about the client's request (see Table 2). Because of this environment information, an SSI program can determine the type of browser being used by the client (by examining the `USER_AGENT` environment variable—see Table 1) and select different HTML content to send, depending upon the browser in use. This method allows for the graphical delivery of information despite inconsistencies that different browsers (and sometimes different versions of the same browser) have in the presentation of certain types of graphic information. As another example, suppose that an experimenter wished to ask international users a set of questions that differed slightly from those asked of domestic users. Again, an appropriately written SSI program could determine the user's geographical locale by using the client's domain name or internet address, and could then deliver the appropriate material. Other uses for SSIs include randomly selecting images, sounds, links, or questions to present, and updating internal hit counters.

Most SSI-capable servers provide facilities for a number of functions to Web administrators without programming, such as including the contents of files in the document being distributed to the client (i.e., embedding a file within another), the date or time that the client's request was answered, a file's last modification date, or any of the environment information listed in Table 2.

Not all Web servers support SSIs, so this is a feature to consider when selecting Web-server software. Keep in mind that there is a small time overhead in the delivery of SSI documents, because they must be interpreted by the server, and actions taken, prior to sending the client code. For this reason SSIs are not usually enabled by default; to be enabled, they generally require some access to the Web-server program. However, the time delay is negligible and unlikely to be noticeable in the low traffic that psychology experiments would produce.

Image Maps

Another feature that some Web-server software provides that may be of interest to psychology experimenters is the image map. Readers with graphical browsing experience on the Web are likely to be familiar with image maps—pictures that have different URLs associated with different locations or objects in the picture. By clicking on the object, one is transferred to the site of a designated URL associated with that object.

Image-map capabilities can be of use to psychology experimenters in a number of ways. Collecting the coordinates of the mouse cursor when the user clicks along a picture of a scale (categorical or continuous) can be implemented by way of an image map. Similarly, one could use image maps to collect categorical responses among a set of provided options.

There are two types of image maps: server and client side. Client-side image maps do not involve the Web-server program, so no special Web-server feature is required. However, client-side image maps do not give the server the opportunity to collect information about the user's click. On the other hand, server-side image-map programs, which are CGIs, process information about the click and make decisions about how to respond. Psychologists may wish to use the server-side method so that they have control over precisely where participants clicked and can record such clicks as data points rather than simply have them trigger the presentation of new Web pages.

Password Protection and Domain Restriction

For some experiments conducted via the Web, it may be desirable to restrict document access to authorized or desired participants. Some Web servers have built-in password protection (or user-authentication) facilities that can fulfill precisely this demand. Servers with password-based document access require that the administrator create a user and password list for document directories that are to be accessed only by authorized users. For Web servers that have this feature and are run on microcomputers, utility programs are often supplied to assist in easily adding or removing users and/or changing passwords.

Table 2
Additional Environment Variables Available to SSI Programs

Variable Name	Description
DOCUMENT_NAME	-the html document name
DOCUMENT_URI	-the path from server root to this document (such as /docs/tutorials/foo.shtml)
DATE_LOCAL	-the current date, host's time zone
DATE_GMT	-same as DATE_LOCAL but in Greenwich mean time
LAST_MODIFIED	-the last modification date of the current document

Whenever a client attempts to access a password-protected Web page, they are asked for a user name and password which is compared with the server's database. The requested documents are delivered only to registered users with correct passwords.

Independent of password protection capabilities, some Web servers can deny clients access on the basis of their network addresses. This feature is known as domain restriction, and it could potentially assist psychology experimenters by allowing them to restrict the population that they access on the basis of the potential client's geographical information. For instance, suppose that one were restricting one's survey to members of a specific university. In this case, the experimenter could arrange the server so that it would not deliver their survey to clients whose domain of origin did not match what was desired. If the survey were to be restricted to Stanford University, then only clients from computers whose domain ended in .stanford.edu could be serviced. It should be noted that this method is rather coarse: users with telnet access to Stanford systems could participate and Stanford University community members attempting to access the server from other domains would be rejected. Nonetheless, domain restriction provides an easy way to better control the population that contributes data to a Web research project.

BENEFITS OF OPERATING YOUR OWN SERVER

Many of the features discussed above (CGI and SSI program execution, domain restriction, and password protection) require that the Web-server program be appropriately configured. Access to accomplish this is not available to the casual user on multiuser systems such as those that are commonly supplied for free at many universities or companies, or for a fee to an internet-service provider for smaller research institutions or individuals.

For instance, on many multiuser systems, allowing any user access to running CGIs or reconfiguring the server software in any way they wish, is viewed as a security risk, because the user must be given access to a wider range of the system's resources than they would normally be allowed. Furthermore, reconfiguring changes the operation of the server for other users. Such increased privileges jeopardizes the integrity of the server system. Running CGIs on such systems requires placing the executable CGI program in an authorized directory (CGI bin by default) and making a reference to it as appropriate from an HTML document. The CGI-bin directory is commonly located in an area where common users cannot gain access, and installing CGI programs or reconfiguring any aspect of the server is under the system administrator's control.

Requiring that a system administrator act as an intermediary between you, your experiment, and your data is a potentially awkward undertaking. This arrangement is particularly limiting during the early development of the data-collection process which necessarily involves repeated testing and modification of CGI and SSI programs,

or reconfiguring and testing the server on line. It is not technically difficult or expensive to operate one's own Web server, thereby excluding the intermediary and exerting complete control over one's experiments. All that is required to operate your own server is a popular microcomputer (i.e., a Macintosh or PC-compatible system) configured for internet access¹ and running appropriately chosen server software (which is often freely available).

Assuming that researchers are going to use CGI programs to collect data, or that they will take advantage of server features that can assist them in confidently collecting data through the Web, then the benefits of operating their own Web servers are obvious. They will not have to rely on others to configure their experiments precisely as they want them, and they will have instant access to the ability to track the experiments under way. Operating their own servers provides researchers with a savings of time and frustration, particularly during development. It can also enable researchers to attempt broader and more complex methods for collecting data over the WWW medium and potentially save them some money.

THE BASICS OF INSTALLING A SERVER ON A MICROCOMPUTER

An important first step in configuring your Web server is to find a server application that will fit your needs. Consult the earlier discussion on running CGI programs, SSIs, and image map use, as well as domain restriction and password protection, and determine what options you need in a server. Next, consult the Web page accompanying this paper (<http://or.psychology.dal.ca/~wcs/WebServerLinks.html>) for helpful links to databases that are available for comparing various server products. Acquire the server software for your platform, and follow the detailed instructions that should be provided with it.

The installation of server software on Macintosh and PC-compatible servers for Windows is quite straightforward. Recall that server applications are programs that await incoming requests from clients. On these microcomputer platforms, server programs need simply to be running in order to activate your Web site, and they generally do not interfere with the ability to run other applications. Web software is commonly distributed in installation packages that simply need to be executed, and the installer will set up everything the server needs to operate (except your HTML documents, the custom CGIs that you wish to use, and any resources that CGI or SSI programs may need). Configuring your server to do things like SSIs, image maps, password authentication, and domain restriction will require reading some documentation that provides details for your server package.

SECURITY ISSUES PERTAINING TO SERVER OPERATION

Web security is a burgeoning field. Of particular worry are security loopholes in servers housed on large multiuser systems. Small microcomputer servers are gener-

ally the most secure, because they do not have extensive schemes for protecting resources from multiple users, and the server software is not dependent on many shared system resources.

Security concerns on single-user microcomputer servers are primarily related to the execution of insecurely written CGI programs. Because such programs are flexible in their operation and can immediately access all of the system's resources, loopholes in their writing can sometimes lead to penetration by knowledgeable outside sources. One particular problem with many Windows single-user systems is that they require that the scripting language of any CGI scripts be located in the CGI directory. This directory is accessible clients anywhere on the Web. If the scripting language accepts directives or entire programs via the command line, any Web user can submit a computer program for execution on the targeted server! To escape this problem, ensure that all scripting programs accessible as CGIs do not accept command-line instructions. For even more security, obtain a server that allows script languages and the CGI scripts that use them to be located in different places, and allow only CGI programs to exist in CGI-accessible directories.

Another potential problem, from a data-integrity viewpoint, is that information can be sent to a CGI program on your server from any client on the Web, regardless of where the HTML documents referring that information were dispersed from. Figure 1 depicts the situation in which a document on Computer B (doc.html) sends information to a CGI program to be executed on Computer C, should the user of Computer A (the client) activate this link. As the figure depicts, the CGI program and the document responsible for referencing it do not need to be accessed from the same Web server. Hence, any Web user could create an HTML document that sends data to your CGI. Measures can be taken when writing CGI programs to ensure that the data they receive originated from authorized hosts (see environment variable HTTP_REFERER in Table 1).

More information dealing with WWW security, including a list of known loopholes in various Web-server applications, can be found at <http://www-genome.wi.mit.edu/WWW/faqs/www-security-faq.html>.

CONCLUSIONS

This paper has surveyed a number of Web-server software features (CGI and SSI execution, image maps, password protection, and domain restriction) that can assist psychology researchers in the collection of data and give them better control over delivering their experiments via the WWW. Experimenters are urged to consider operating their own server so that they can have immediate access to configuring all facets of the server software to further their experimental goals.

REFERENCES

- KIELEY, J. M. (1996). CGI scripts: Gateways to World-Wide Web power. *Behavior Research Methods, Instruments, & Computers*, **28**, 165-169.
- KRANTZ, J. H. (1995). Linked Gopher and World-Wide Web services for the American Psychological Society and Hanover College Psychology Department. *Behavior Research Methods, Instruments, & Computers*, **27**, 193-197.
- SCHMIDT, W. C. (1997). World-Wide Web survey research: Benefits, potential problems, and solutions. *Behavior Research Methods, Instruments, & Computers*, **29**, 274-279.
- WELCH, N., & KRANTZ, J. H. (1996). The World-Wide Web as a medium for psychoacoustical demonstrations and experiments: Experience and results. *Behavior Research Methods, Instruments, & Computers*, **28**, 192-196.

NOTE

1. Even if a researcher does not have these resources, a user account with about 6 MB of disk space on almost any internet-capable UNIX platform can suffice. Although this is a technically more involved method of setting up a server, it is inexpensive. See the resource page associated with this article (<http://or.psychology.dal.ca/~wcs/WebServerLinks.html>) for an introduction toward getting such a server up and running.

(Manuscript received September 23, 1996;
revision accepted for publication January 9, 1997.)