Chapter 1

# STARTING POINTS

Ten years ago, no one could anticipate the extent to which the World Wide Web would become an essential aspect of library services. Small numbers of librarians took up the Web with a sense of experimentation, adventure, and play. Without conscious forethought, they invented what eventually resulted in today's library websites.

As has always been common in the computing world, but not in the library world, the technology continually changed faster than the organizations using it, and the position of library Web manager came into being with no consistent definitions of the responsibilities involved or training required. What does a Web manager do? What does that person need to know? How can a Web manager contribute to the success of a site, and how is that success defined?

This report explores these questions. It discusses technical requirements for a server, including decisions many libraries face about hardware, software, and site hosting options. It describes the types of tools used to make both HTML files and dynamic pages, and some of the goals of Web design regarding usability and accessibility.

This report emphasizes the importance of standards in the creation and design of websites. Many Web design tools support an increasingly outdated authoring model in which sites *look* right for almost all users. Web managers who understand and apply the standards discussed here can make sites that *look* right for almost all users, but that also make sense and *work* right for all users. More and more libraries include that difference between supporting *almost all* and *all* users as an important part of what defines a successful website.

This report does not contain product recommendations or reviews. It describes issues that arise in managing library websites. Web managers with an understanding of these issues are better prepared to address questions they will face about creating and running their libraries' websites.

## What is a Web manager?

In this second decade of the Web's existence, many libraries can point with justifiable pride to the ways they have added Web resources to their service offerings. The technology of the Web is compelling, and the benefits of a library's presence on the Web are numerous. But personnel choices about who should guide this presence may be less than clear. Technology changes more quickly than job descriptions; libraries are left to determine what training, job skills, and experience are needed by the person running their Web servers.

As librarians integrate the Web more deeply into overall library services, the role of the Web manager often remains grafted onto the responsibilities of some other position. As a result, the definition of a Web manager is nebulous.

In simplest terms, a Web manager is responsible for establishing, configuring, maintaining, and upgrading a website. This role is often assigned to someone who also fulfills the duties of any or all of the following positions:

- Systems administrator, maintaining server computers
- Systems librarian, maintaining library online systems and in-library computer equipment

- Web designer, making large-scale decisions about a website's form and structure
- Web author, generating the website content

As Web services become mission-critical for libraries, the role of the Web manager deserves greater attention and understanding. The Web manager can make systems decisions about server platforms to support, determine Web servers and client software on workstations, and create designs that affect the usability of the site.

The terms *Web manager* and *webmaster* are interchangeable, but an e-mail address for webmaster@domainname.org is a de facto standard. Always make it the alias for the Web manager's e-mail address.

## Using relevant standards

In many cases, Web managers are best prepared to make these decisions by understanding the relevant standards affecting a particular aspect of Web services. This report refers repeatedly to standards for HTTP, HTML, stylesheets, and accessibility. Most of these standards are maintained by the World Wide Web Consortium (W3C), and W3C standards are typically written in language that encourages Web managers to read them, rather than dense technical jargon.

**HTTP:** Hypertext transfer protocol

**HTML:** Hypertext Markup Language

Far too many popular books and tutorials on Web design make no reference to or use of these W3C standards. This lack contributes to the current state of the Web. The majority of Web pages consist of HTML markup that violates both the letter and the spirit of the HTML law, including invalid syntax and attempts to control document appearance in poorly designed ways that were officially discouraged by W3C as early as 1996, such as the nearly ubiquitous <CENTER> and <FONT> tags.

In addition, on many sites poor coordination between Web managers and Web authors leads to combinations of HTML and JavaScript being employed for functions that more properly should be handled through the standard mechanisms of the HTTP communications standard.

**HTTP 1.1 standard,** ftp://ftp.isi.edu/in-notes.rdc2616.txt

In many cases, these poor authoring practices are encouraged by or entirely attributable to editing tools purchased in good faith that were touted to create websites appropriately. These editing tools create bad markup because they attempt to support a vanishing generation of obsolete browsers that required tortuous misapplication of simple HTML elements for satisfactory page design.

The most common examples of bad markup include page designs that consist of intricately nested tables used not for tabular information but simply to position page elements. Although the number of browsers in use that require this type of deliberate mistagging is now tiny, editor programs have been slow to support newer methods to achieve the same page designs with cleaner markup.

Web managers can break this pattern of poorly structured documents provided for poorly designed browsers if they apply standards with clarity and coordinate their efforts with the efforts of others working on a website. Web designers may need reassurance that pages can be written with valid syntax and with care for accessibility concerns—yet not be unattractive and boring. Library systems staff may need reassurance that state-of-the-art pages can use detailed stylesheets to affect their presentation without breaking older browsers still in use.

## Choosing a server location

Web managers rarely start sites from scratch. Typically, some decisions have been made about where the site will be hosted, the host computer's operating system, and Web server software (or the preferred language for scripts and dynamic pages).

These decisions are often reflect larger institutional choices about what hardware and software to support, or whether departments have permission to run their own Web server or locate their sites on a central server. Success in starting a new site is often a matter of peaceful coexistence with these decisions.

In some cases, though, a library Web manager does have input into basic decisions about the server and will want to consider the advantages and disadvantages of each option.

### *Hosted by institutional server*

Most organizations expect their Web servers to be available 24 hours a day, to have high-speed network connections, to maintain adequate backups to restore the server in event of a crash, and possibly to use an uninterruptible power supply (UPS). Although these goals can be met by a computer running in the corner of someone's office, many libraries belong to a larger organization with an information technology (IT) department already offering these services.

A city, campus, or company computer room will already have high-speed network connections, backup and UPS systems, and perhaps even overnight staffing to respond to problems 24 hours a day. These benefits of locating the library site on a central server may be worth any drawbacks.

Those drawbacks should be known and understood before making a choice. Without firm agreements in place, the operators of the central server may make unilateral decisions regarding the server-side technology available for use, access to logs for usage analysis, or even the stability of URLs on the server. Before locating a site on a central server, the library's Web manager should know the answers to these questions:

- What server-side technologies are available for form-handling and scripting? What is the procedure for requesting new technologies be installed?
- Will the library have access to all usage and error logs for hits on its site? Some system administrators may offer only a usage summary.
- If the central site has to move files and directories, what guarantees are made about retaining URLs? Most changes can be made without requiring sites to change the URLs, or at least their homepage URL. If URLs do have to change, how long will the old URLs continue to point to the new ones?

If the central IT office cannot make adequate promises in these areas, a library may be more likely to consider running its own server.

### *Hosted by an ISP*

Libraries with no central IT department to rely on, and that may not feel ready to run a server locally, may choose to locate their site at an Internet service provider (ISP) or commercial Web hosting company. As with a central organizational computer room, this alternative puts the responsibility for IT chores such as server maintenance, backup, and upgrades in the hands of full-time IT professionals, but it further removes the library from decisions about what software to support, log access, and URL stability.

If a library website is hosted by either an institutional server or a commercial ISP, a Web manager may want to inquire about running the site as a virtual host. Without virtual hosting, a website maintained on a commercial ISP's server may have a URL such as www.gargantua.com/~smithvillelibrary.

With virtual hosting, the same site on the same commercial site could have a URL such as www.smithvillelibrary.org, which offers several benefits. This type of URL gives the library's website a greater sense of identity: it has a URL that clearly belongs to the library and establishes its location on the Web. This type of URL also leads to greater stability. The library can eventually move to another ISP or choose to run its own server without changing any URLs.

Virtual hosts are differently named web sites located on the same host computer. Names such as www.smithvillelibrary.org and www.jonesburglibrary.org may correspond to the same physical server, but browsers receive different content depending on the host name they use in their request.

### *Locally run*

Many libraries have their own IT staff and may already run servers for local area networks or online catalogs. Compared with these responsibilities, maintaining a Web server is not particularly challenging. Other libraries may be satisfied with the reliability offered by simply loading Web server software on a hand-me-down desktop computer. In either case, running a local Web server is not a great technical challenge, and it gives the library the maximum degree of control over its site.

In many cases, the arguments to be made against running a local server are not related to the library's technological capabilities. The concerns may reflect policy: for example, a large organization may wish to ensure consistent quality, reliability, and appearance across all its departments' websites. The library may want to outsource the jobs of performing backups, handling hardware and software upgrades, and guaranteeing uptime and security. The library's own Internet connection may be either too slow or too unreliable to maintain a server locally. Although every other configuration choice for Web servers has options, no alternative exists to having a fast network connection to the server.

Any of these hosting options can give the library a well-run, reliable website with good performance. Virtual hosting can allow a library to choose any option and still establish its online identity with a meaningful URL that is likely to be stable over time. The deciding factors may be whether remote hosting options provide sufficient control over important aspects of website management, and whether the library's IT staff can provide the level of support considered necessary.

## Choosing server hardware

Libraries that choose to run their own Web server, and even many that have their sites hosted elsewhere, need to choose the server's hardware, operating

system, and Web server software. As with the choice of server location, some or all these choices may be set by organizational fiat to establish consistency or simplify support needs. Some organizations only allow Windows servers, and others may specify Unix servers running Sun Microsystem's Solaris operating system, for example. Some may require using Microsoft's Internet Information Server (IIS) as the Web server software, and others may ban the use of IIS entirely.

Sites without policies determining software choices may have contracts with hardware vendors that make buying one type of server hardware easier than another. This arrangement often leads to a choice of server operating system as well as hardware.

For example, some organizations receive steep discounts on Apple Macintosh servers, so they run many servers with the Mac operating system. Others routinely receive server hardware from Sun or Hewlett-Packard, so they run the versions of Unix those companies provide. Others buy server equipment built on Intel processors or compatible PC platforms and typically run a server version of Windows or Linux.

The computing world has a history of spirited debate among proponents of each operating system, but the truth is that current versions of any of these platforms are easily capable of running basic websites.

Because any reasonably up-to-date combination of server hardware and operating system can run a website, trying to identify the one absolute best combination is usually less productive than identifying the combination that best matches available expertise. For example, sites with experienced Windows administrators are most likely to receive optimal performance from a server that runs under Windows.

In addition to these basic choices of server hardware and operating system, setting up a Web server includes a choice of Web server software. Sites planning to offer more than just static HTML pages also need to decide on scripting languages and often on Structured Query Language (SQL) database software. As with choices of hardware and operating system, the major options in these areas are all capable of supporting good websites.

If options for hardware, operating system, server software, scripting, and databases have no indefensibly bad choices, how should a Web manager make these choices? Two main configurations for Web servers exist, although each is made up of several components that can be substituted for many alternatives.

---

*In the course of maintaining the Libweb directory of library sites (http://sunsite.berkeley.edu/Libweb), the author surveys the server software used by libraries around the world. As of October 2002, 41% of library websites run on IIS, 39% on Apache, 10% on servers from Iplanet (formerly Netscape servers), and 10% on less common servers.*

*By contrast, the Netcraft survey of servers on the entire Web shows Apache running 60% of the world's servers and IIS running 29%.*

---

**Netcraft,** www.netcraft.com

**IIS,** www.microsoft.com/iis

**The "Nobody ever was fired for buying Microsoft" model:** This configuration uses a server version of Microsoft Windows NT, Windows 2000, or Windows .NET. It uses Microsoft's IIS as the Web server, Microsoft Active Server Pages (ASP), and Visual Basic for server-side scripting, and either Microsoft Access for lightweight database needs or Microsoft SQL Server for

more substantial work with databases. This model makes sense in an organization that has invested heavily in support for Microsoft products and has technical staff trained in supporting Microsoft products.

**The LAMP model:** This configuration uses the Linux operating system, the Apache Web server, the MySQL database program, and the PHP scripting language. Proponents of other scripting languages maintain that the "P" in LAMP stands for Perl or Python, two more open-source scripting languages. Many servers use combinations of these languages.

**Apache,** www.apache.org

All the components of a LAMP server are freely available open-source software. (Not all free software is open-source and not all open-source software is free.) Open-source software makes its internal programming code available for any interested party to view, use, and develop. One advantage of this availability is that many software developers can, and do, examine the code in detail, adding beneficial improvements and also discovering potential bugs and security holes.

By contrast, proprietary closed-source software can only have its source examined by the comparatively few programmers employed by one company. In large part because of this restricted access, proprietary Web servers have a track record of having more security holes than open-source Web servers.

Common security holes include ways a Web user might force the server to run any command of the user's choice or to view files not intended for public availability. Almost by definition, this use involves actions that the software's developers did not anticipate, making open review by other developers effective for finding these holes.

In either of these models, the scripting language (ASP or PHP) can be replaced by or used in conjunction with alternative languages such as Cold Fusion, or CGI scripts written in any language, with Perl predominating. PHP also can run with IIS, and, with the addition of a freely available software module, Apache can run ASP. Likewise, organizations with other SQL database programs on hand can usually use them as easily as SQL Server or MySQL.

**A mixed model:** One server configuration that has not yet seen widespread use is Apache 2.0 running on Windows. Versions of Apache 1.3 have been available for Windows servers for several years, but some Unix-specific techniques for managing memory use lead to performance problems under Windows that make Version 1.3 unusable for many sites. By contrast, Version 2.0 performs on Windows servers as well as or better than IIS. At sites with a preference or requirement for Windows, a WAMP (Windows, Apache, MySQL, and PHP) server gives Web managers robust open-source tools for all the remaining software components under their control.

### *Server security*

Security is another factor in deciding what server to run. A server with security holes is subject to several types of attack: denial of service (DoS) attacks make the server nonresponsive or bring it down entirely; defacements occur when attackers are able to alter the server's Web content; other attacks allow users to view system information or files that should not be available through the Web server or to have the server execute arbitrary commands.

Hackers can use these security holes to view lists of accounts on the server and attempt to access those accounts, or to run programs on one server that attack additional servers. Running a Web server requires the Web manager to take all reasonable steps to keep the server secure.

Many security holes have been discovered in Microsoft's IIS, but comparatively few have been discovered in Apache. To an extent, this difference is part of the advantage of open-source software; many developers viewing and working with a program's source code increase the likelihood that a potential weakness of any sort will be spotted before the program goes into use.

IIS's greater number of security holes is also due to its design, which Microsoft tightly ties to the server versions of its operating systems. Apache remains separated from a server's operating system, reducing the risk that a hacked Web server will give the hackers control over other system functions.

Security concerns have prompted some companies to migrate their servers from IIS to Apache. In fall 2001, the Gartner research group made a controversial recommendation that companies stop using IIS entirely. Apache 2 can run on Windows servers with performance comparable to IIS, making this switch easier than in the past.

But don't assume every Apache server is secure. Security holes have been found in Apache itself. Any server can become insecure if it runs scripts that are themselves insecure. Examples of script-related security holes are scripts that allow Web users to view files that are not otherwise accessible via the Web; script authors must ensure no one can use these files to display sensitive system information.

Other scripts pass user input to system commands. This input must be inspected first to verify it has no special characters that could cause user-specified commands to run on the server and also that the input is not likely to cause the system to hang (typically by being many times longer than expected).

These script-related problems are not an exhaustive list of possible security issues but illustrate the hard reality that any Web server may have security holes and may come under attack. Web managers must continually review system security. This review should:

- Regularly monitor security sources for announcements of new attacks. These sources include the U.S. Department of Energy's Computer Incident Advisory Capability (CIAC) and the Computer Emergency Response Team. Prominent attacks also are often reported on computer industry news sources such as CD|Net and Ziff-Davis Net.
- Periodically review the websites of server programs for announcements of security patches, and apply new security patches with all due speed.

  Several of the most infamous attacks on IIS servers used holes for which Microsoft had provided patches months before, but many Web managers had not installed them. In other cases, vendors of compromised programs commonly provide patches within one or two days of the attack. Web managers should monitor the sites both for Web servers themselves and for any other scripting languages used on the website.
- Continually check scripts and dynamic pages on the server to guard against holes that provide system access.
- Confirm that the server is physically secure. Servers located in unlocked offices are easy prey for attackers. In the space of a few seconds, even an inexperienced intruder can use physical access to the server to disable it. Experienced hackers are able to run commands that will give them control over the server at a later time.

The U.S. Department of Energy's Computer Incident Advisory Capability (CIAC) lists 28 security bulletins related to IIS at www.ciac.org/ciac/bulletinsByType/vndr_ms_bulletins.html, including the Code Red and Nimda worms that brought down thousands of servers. By contrast, CIAC lists six bulletins for Apache, at www.ciac.org/ciac/bulletinsByType/vndr_apache_bulletins.html. Of the six, two are advisory notes not related to Apache software.

On Unix and Linux systems, a common target is the /etc/passwd file. This target lists all the user accounts on the server with encrypted versions of each password. Attackers with access to this file can systematically test every word in a dictionary to see if its encrypted version matches a user password; if it does, attackers gain access to that account.

**U.S. Department of Energy's Computer Incident Advisory Capability (CIAC),** www.ciac.org
**Computer Emergency Response Team,** www.cert.org
**CD|Net,** www.news.com
**Ziff-Davis Net,** www.zdnet.com

Patches are program updates designed to fix specific problems, rather than to provide a general upgrade.