Internet Publishing Handbook - Copyright © 1995 by Mike Franks (Source: http://mikeandnaomi.com/book/ch03.htm)

CHAPTER 3: Gopher and Gopher+

Depending on whom you ask, Gopher and its newer version, Gopher+, are in the midst of either their growing pains or their death throes. Gopher has already been surpassed by WWW in total number of servers and amount of traffic on the Internet, but with more than 7,000 servers worldwide and an extremely active development community, Gopher is an Internet publishing method that is hard to ignore. Here is a summary of Gopher's pros and cons:

ADVANTAGES

- Easy set-up and low maintenance.
- Little or no document preparation necessary.
- Light load on network because of small file sizes.
- Facilitates automatic indexing of server menus.
- Servers available for many types of computers, including mainframes.
- Client can run on old, slow computers.
- Gopher+ can serve up multiple versions of the same file.
- Client software available in English, Chinese, Japanese, Spanish, French, and Finnish.
- Strong development community.

DISADVANTAGES

- Can't mix graphics and text on same screen.
- Gopher client can't view HTML.
- Commercial users of University of Minnesota server software must license it.
- Gopher is divided between Gopher0 and Gopher+, and not all clients (browsers) handle Gopher+.

But even if you're enticed by WWW, don't pass Gopher by too quickly. Graphics are nice, but for a large percentage of Internet users, graphics are a luxury that their modem speed can't support. For these users Gopher is an excellent tool. Considering that preparation of material for Gopher is extremely simple and the Gopher software is easy to install and maintain, like FTP, Gopher will probably always have a place on the Internet. Gopher is an excellent choice when graphics are not an issue and a wide variety of people with little or no training need to add their files to the server. If you just want to get the information out there, Gopher is likely your best choice.

Gopher: Pros and Cons

Gopher has the advantage of being a simple protocol that is easy to set up and maintain. Adding documents and files to a Gopher server is usually a matter of just copying them into the Gopher directory structure. Unlike WWW servers, Gopher servers don't need special formatting or linking. Just drop the file in a Gopher directory and it will show up in the Gopher menu.

The average size of Gopher menus is small, usually 2K to 6K, whereas WWW home pages are usually 5K to 10K or more, and **inline images** (graphics that appear within a Web document), can increase that size to 100K or more. The smaller size of Gopher menus translates to quicker response to the user and less load on the Internet.

Gopher also does not require the user to have an extremely fast computer. Because the principal element of Gopher is a menu, it can be conveyed in text as easily as in graphics. So graphics monitors aren't even necessary.

A Gopher server is also flexible--it can serve up almost any type of file; **Gopher**+ allows you to offer different versions of the same file (for example, WordPerfect, PostScript, and plain text versions of the same document, or Spanish, French, and Swedish versions).

The biggest drawback to Gopher is that it just isn't as sexy (or impressive in appearance) as WWW can be. Although it can make picture files available for downloading, it can't mix the pictures and text together in a glossy brochure-like presentation as WWW can.

One drawback to Gopher as a client program is that Gopher client programs can't browse WWW servers, although WWW client programs can view Gopher servers and retrieve data. Nevertheless, Gopher remains an extremely powerful and useful Internet publishing tool, with thousands of Gopher servers worldwide.

How Gopher Works

For technical details on the operation of Gopher, see "The Internet Gopher Protocol" and "Gopher+--Upward Compatible Enhancements to the Internet Gopher Protocol," both available at <<u><gopher://boombox.micro.umn.edu:70/11/gopher/gopher_protocol></u>">.

Here's a simple description of what goes on between a Gopher server and client:

Person selects an item on a Gopher menu.

Person's Gopher client reads the Gopher server address and port number in the link the person selected.

Gopher client contacts Gopher server on that port.

Gopher server accepts connection.

Gopher client asks for directory or file.

Server either fulfills the request or reports (via an error message) that it's not there.

If you want something else, you repeat the process. The Gopher server finds it an efficient process because it doesn't have to keep track of ongoing requests. Answer a question and wait for the next question--that's all it does. This can happen hundreds to thousands of times per day, depending on the popularity of the Gopher server. There is an agreed-upon list of the "types" of items that the system can convey, and the list is open-ended. It is administered by the University of Minnesota and is listed in the Gopher FAQ. You need to study the exact language or syntax used in the Gopher protocol only if you're writing Gopher server or client software.

Gopher and Gopher+ Features

Unfortunately, it's not enough to decide you want to run a Gopher server. You then have to choose Gopher0 or Gopher+. (The original Gopher protocol is referred to now as *Gopher0*.) Whether Gopher+ is an option depends on the type of computer you're going to run your server on. Gopher+ is the newer definition, or standard, toward which the Gopher community is moving. Gopher+ adds some nice features to Gopher, such as interactive online forms, multiple versions of the same file (called *views* in Gopher+ documents), and additional meta-information (or information about the files, such as administrator, date updated, and so on).

Gopher simplifies downloading from FTP servers because you immediately view what you download and then download more. This is particularly helpful because most FTP sites have readme and index files that you need to read.

Shell scripts are specially written programs that allow you to add other functions to your Gopher server. Scripts have been written to connect Gopher servers to database programs, for example.

The list that follows details features of both Gopher0 and Gopher+, but be aware that some Gopher servers may have additional features. Others may not have all the features listed here.

Gopher0 FEATURES

- Provides users with a menu of items available on Gopher server.
- Delivers directory indexes.
- Delivers files of specified types.
- Delivers telnet or tn3270 sessions (remote login) to another computer.
- Delivers links to items on other Gopher servers.
- Delivers links to files on anonymous FTP servers.
- Delivers results of index searches.
- Runs scripts on server.
- Answers requests in one step and then disconnects, which allows the Gopher server to run faster and handle more requests.
- Uses easily understood directory structure to store files on server.
- Permits system to define other item types at a later date.

- Configurable for querying remote WAIS servers and funneling the results to Gopher clients.
- Configurable for searching local WAIS indexes and indexing your own information.
- Queries remote FTP sites and funnels the results to Gopher clients.

Gopher+ FEATURES

- Provides administrator contact information for each item as a way to track the sources of information.
- Provides the date on which the document was last modified.
- Provides abstracts of items, especially useful for long documents.
- Provides online forms (called **ASK blocks**). This lets you ask your users to fill out an online form.
- Provides alternate views or file formats for Gopher items. This simplifies the process of providing documents in different word-processing formats.
- Permits system to define other item attributes and easily add them at a later time.

Gopher Resources

One way to learn about Gopher is to download the server software and set one up. It's easier than you think. But for those of us who like to test the waters first, see Table 3-1 for some sites and resources that will tell you a great deal about Gopher matters.

Note that some of the mailing lists are duplicates of Usenet newsgroups. And both are often archived at sites that provide indexed searching. Learning how to search these is one of the first skills you'll want to acquire. As you will see, Gopher servers can be run on many different types of computers. Some mailing lists and Usenet newsgroups have sprung up around different versions of Gopher servers. They may have a narrow focus, but they can be extremely rich in expertise and experience.

Generally, your plan to learn more about Gopher and Gopher+ should be to

Read the FAQs.

Look at existing servers and note features that you like.

Follow related Usenet newsgroups and list servers.

Install and run the Gopher software.

Learn how to use the bookmark feature in your Gopher or WWW browser. Then use it to "capture" (keep a record of) links to any site that might be useful to you later. It's possible to learn a lot from existing Gopher servers, both in style and in technique. Look at how they are laid out, and watch for techniques that make it easier or harder for you as a user to find information. If you have questions about how something was done, send e-mail to the Gopher administrator and ask. And if you like what you see, say that too. Don't be upset if you don't get a response (they are probably as busy as you are), but it's very much within the culture of the Internet to ask how something is being done. No one waits for an introduction.

The University of Minnesota's Mother Gopher site is the first listed in Table 3-1 because it is the official Gopher archive. Among other things, the Gopher team has set up examples of how all Gopher data types show up. These demonstrate each capability of Gopher.

Types of Items Gopher Handles

Gopher item types are important because they describe the kinds of information and services Gopher servers can deliver. You may think you just want to put text files and images on your server. But the Gopher server has to tell the Gopher client exactly what type of information will be sent out if the user's Gopher client program is to transmit and display the text file or image correctly. Most Gopher browsers (the terms **browser** and **client program** are used interchangeably) use a different symbol for each type of information they show. The users don't actually see the type codes listed in Table 3-2, but as a Gopher administrator, you will be dealing with them often. In many cases part of putting something on your Gopher server means specifying its type. It is important to understand the different types so that you can use them appropriately. Right now the key thing to know is that Gopher has a wide variety of types; you can come back and learn the details later.

See Table 3-2 for the current list of the many types of data and services Gopher servers can provide. The University of Minnesota's Gopher team keeps track of additions to the list. Some types aren't files at all. They represent some other kind of action that the Gopher server can provide, such as a telnet session and a tn3270 session type. They are both methods of logging in remotely to another computer. When a Gopher browser program sees type 8, which signifies a telnet session, it takes the host's name and port number provided and starts its own telnet session to that host computer. Type 7 signifies that a search will be initiated. For an explanation of **MIME** types, see Table 3-3. You can create your own data type too, although if you want anyone else to recognize it, you should register it and make sure their browser programs know how to handle it.

The list of Gopher item types has grown since Gopher was originally created. New types such as MIME, HTML, and GIF have been added because users expressed the need for them. This flexibility is extremely important, because it means that Gopher has the capacity to grow and adapt and allow you to connect your users to many, if not all, existing Internet services.

Given the wide variety of formats in which you can publish using Gopher, don't forget the lowest common denominator, plain text. Gopher was designed for the low-end machine with a slow connection to the Internet. Also, Gopher is so widespread that you can never tell what kind of computer your viewers will be using. Although PostScript, Portable Document Format (PDF) from Adobe, and various word-processing formats make documents look much nicer than plain text, only plain text will be viewable if the user is on a mainframe Gopher client.

Gopher Servers

Gopher servers have been written or ported (short for *transported*) to many different types of machines. The University of Minnesota's Gopher Development Team started with a version for machines running the UNIX operating system, which includes Sun Microsystems, Silicon Graphics, IBM RS6000, and NeXT computers. Then the team wrote a version to run on Macintoshes, Gopher Surfer, which now has full-text searching. Other groups then wrote versions for their favorite machines, including DEC Vaxen, IBM mainframes, PCs running Windows NT and OS/2, and even lowly DOS machines.

Some servers run only the basic Gopher protocol, but many have moved on to Gopher+. Other servers offer more advanced functions. GN, from John Franks (no relation to the author), a professor at Northwestern University, is a combination Gopher and HTTP (WWW) server.

As with any software you need to address at least these five concerns when choosing a Gopher server:

Function--Look at the functions supported, the extra capabilities the server provides, as well as its record of compatibility with add-on indexing programs.

Platform--Check the platform (operating system and hardware) required, such as UNIX, Windows, and Macintosh, and its relative stability. Be aware of the difficulties in supporting platforms you're not familiar with.

Load--Investigate the load this server/platform can be reasonably expected to handle. Be sure it is high enough for two to three times the load you expect.

Support--Verify the level of support and activity this software enjoys, including newsgroups and mailing lists, as well as more formal commercial support mechanisms. An active newsgroup of technical administrators can be better than many companies' tech support desks.

Price--Determine the licensing policy. If it's freeware, check that it's not limited to educational organizations. Many free programs have a different policy for commercial uses.

The section that follows discusses the capabilities of many existing Gopher server programs.

UNIX

UNIX is the most powerful and flexible operating system available for Gopher servers. It is also extremely widespread. For this reason most new development is for UNIX first. Maintaining a UNIX system does require quite a bit of technical knowledge and experience.

Gopherd. Gopherd is a UNIX-based Gopher+ server from the University of Minnesota and is the leader in the development arena. This server always gets the new features first. Contact the university for licensing information.

<<u><gopher://boombox.micro.umn.edu:70/11/gopher/Unix></u>>

GN. GN is a free UNIX-based multiprotocol server from John Franks at Northwestern. This server is unique in that it serves both the Gopher and WWW worlds, sending the same documents out via either protocol, depending on the type of request.

The GN server can act as both a Gopher (but not Gopher+) and WWW (HTTP) multiprotocol server and has been recommended as an intermediate step between Gopher and WWW. The support newsgroup gn-maint-l is very active and enthusiastically supports GN. An FAQ file on GN is available at <<u>http://www.unicom.com/1/FAQ></u>> whereas the software can be found at <<u>http://hopf.math.nwu.edu:70/></u>>. John Franks has stopped development of GN and moved onto a Web (HTTP only) server called WN. See Chapter 4 for a list of Web servers grouped by computer platform.

VMS

VMS, short for Virtual Memory System, is Digital Equipment Corporation's multi-user, multitasking operating system. It runs on DEC's VAX series of computers, from the smallest minicomputer to the biggest mainframe.

VMSGopher Server. VMSGopher Server 0.6b is a free Gopher server for the VMS operating system. The "tweaked" code is

not always available at Boombox, so check other sites as well. Check out the Gopher FAQ and VMSGopher Server Archives (see Table 3-1). Several sites have beta-test source code for VMS, which is often ahead of the version stored at the

University of Minnesota. <<u>gopher://psulias.psu.edu></u>> or <<u>sopher://niord.shsu.edu></u>> or <<u>sopher://gopher.wfeb.edu></u>>.>

Precompiled (ready to run) executables are available at

<<gopher://trln.lib.unc.edu>>.

The Minnesota archive is available at <<u><gopher://boombox.micro.umn.edu:70/11/gopher/VMS>></u>.

Macintosh

Macintoshes are a user friendly place to set up a Gopher server. You can manage quite a bit of customization by using AppleScript (a command language that comes with System 7) and MacPerl (a free Macintosh port of the Perl language). An alternative to mounting a Gopher server on a Macintosh or PowerMac is to run A/UX (Apple's UNIX) instead of the normal Macintosh operating system. Then you would choose a Gopher server from the UNIX section.

Gopher Surfer. Gopher Surfer is a Macintosh-based Gopher+ server from the University of Minnesota that requires System 7.0. It uses AppleSearch to provide full-text searching. Contact the university for licensing information.

<<u><gopher://boombox.micro.umn.edu:70/11/gopher/Mac_server></u>>

Windows NT

Microsoft Windows NT 3.5 is now a serious choice for Gopher administrators, primarily because of the ease of installation and administration. Windows NT also offers the ability (on some servers) to interact with Visual Basic and data in regular Windows applications such as Microsoft Excel and Access.

EMWAC GopherS for Windows NT. GopherS, pronounced *Gopher-ess*, is a Windows NT-based Gopher0 server (not Gopher+) that runs as a Windows NT service so you don't have to remain logged in. It handles multiple simultaneous connections using multiple threads and can handle WAIS database searching if you use its WAISTOOL tool kit. It has a UNIX compatibility mode, which will be helpful for those familiar with UNIX Gopher servers. FTP from <*emwac.ed.ac.uk*>in the directory pub/gophers. <<u>http://emwac.ed.ac.uk/HTML/internet_toolchest/top.HTML</u>>

DOS and Windows

DOS runs on a minimal PC so it is ideal for setting up extremely inexpensive Gopher servers. Windows has the advantage of being widely used (even if it's not always stable), so it is an easy platform to begin with.

KA9Q NOS. KA9Q NOS is a DOS-based Gopher and e-mail (POP2/POP3/SMTP) server written by Phil Karn. It requires at least a 286, but a 386SX with 2MB of RAM is recommended. <<u><gopher://boombox.micro.umn.edu:70/11/gopher/PC_server/ka9q></u>>

GO4HAM. GO4HAM is a Windows Gopher server (using WINSOCK.DLL) written by Gunter Hille of the University of Hamburg. It includes full-text search capability and gateway to the Borland Paradox database program. Available from <<u>ftp://ftp.informatik.uni-</u> <u>hamburg.de/pub/net/Gopher/pc/go4ham></u>> or from <<gopher://boombox.micro.umn.edu:70/11/gopher/PC server/hamburg>>

OS/2

IBM's OS/2 version Warp 3.0 has the advantage of being relatively stable while running on PCs with less memory than Windows NT requires.

GoServe. GoServe is a Gopher server designed to run on IBM's OS/2 2.x operating system. It provides for an audit trail of requests and actions. GoServe is free. <<<u><gopher://boombox.micro.umn.edu:70/11/gopher/os2></u>>

Other Platforms

MVSGopher Server. MVSGopher Server runs on the MVS operating system, which usually means large IBM mainframes. <<u><gopher://boombox.micro.umn.edu:70/11/gopher/mvs</u>>

Rice CMS Gopher Server. Rice CMS Gopher Server runs on the CMS operating system. Check the resource table (Table 3-1) for the list server information. <<<u><gopher://boombox.micro.umn.edu:70/11/gopher/Rice_CMS></u>>

VieGOPHER. VieGOPHER runs on the CMS operating system and was written by Gerhard Gonter of Vienna. <<u><gopher://boombox.micro.umn.edu:70/11/VieGOPHER</u>>>

Other Servers

The best way to find out about new Gopher servers is to go looking on <<u>gopher://boombox.micro.umn.edu></u>>, check the Gopher FAQ periodically, and watch the Gopher newsgroups (see Table 3-1).

Installing Gopher Server Software

Gopher and Gopher+ are protocols. The Gopher protocol is a defined way that the server and the client communicate. The protocol does not specify how information is to be stored on the server, nor does it in any way define how the Gopher server should go about doing its job.

The programmers who wrote each of the Gopher servers had a lot of leeway to do things in a way that made the most sense for the machine and operating system they were using. A Macintoshbased Gopher server will look very different (to the administrator) from one designed for a mainframe. Basically, each version of the Gopher server software is different, and the installation instructions are different as well.

For these reasons I won't give you step-by-step instructions. We'll go through the basic steps of installing and setting up Gopher server software, using the UNIX Gopher server as a model. The point here is to understand what options are generally available and the kinds of decisions you'll have to make, no matter which type of Gopher server you choose.

If you have problems, check the mailing list or Usenet newsgroup (and their archives) related to your server software to see whether someone offers a solution. Providing step-by-step installation instructions for each version of Gopher server software is beyond the scope of this book. Instead, I assume that you or the person who will be responsible for your server will be able to use the software documentation to install it, and I will concentrate on providing you with an explanation of the concepts and techniques that you'll begin to use once the software is installed.

Downloading, Uncompressing, and Compiling the Software

You have to get your software from somewhere, and usually that means you're getting it by Gopher or anonymous FTP from one of the sites listed earlier in this chapter. The most common error is to forget to switch to binary mode before FTP-ing the software down to your machine.

On UNIX the files come in the form filename.tar.Z, which means they've been both tarred and compressed. (**Tar** is a UNIX command for combining zillions of tiny files and directories in one big file.) Make sure you have plenty of disk space, then type Uncompress filename.tar.Z, which will replace your compressed version with a much larger file without the .Z at the end. Then make sure you are sitting in a directory one level above the directory into which you want to put your files, and type tar -xfv filename.tar, which will create the subdirectories and unpack all the files that go into them.

Macintosh files usually come in BinHex format (.hqx) to travel safely across the Internet, and sometimes they are also in a self-extracting archive (.sea). Often these two are combined so that the file would look like filename.sea.hqx, but unloading it is easy with the shareware BinHex program (use Archie to find it on the Internet).

UNIX and VMS versions usually have to be compiled before they can be run. Before you compile, determine which additional gateway features you want your server to have. WAIS is the program most commonly added to Gopher, and it's essential for some of the indexed searching functions you may want. Compilation problems are one of the most talked-about issues in the Gopher newsgroups. The best thing you can do is read the documentation carefully and take careful notes of each step in the compile process. If you do have to go asking for help in the Gopher newsgroups, you'll be able to accurately summarize your situation (don't leave out the details of your operating system). And when the next release of your Gopher server software comes out, you'll be able to avoid the mistakes you made the first time around.

What's Involved in Gopher Server Configuration

Gopher server configuration refers to editing certain configuration files to add information about your site and decide how the server should behave in various situations. The information specific to your site is easy. It consists of such things as the server's name, administrator's name and e-mail address, time zone, default language, location, latitude/longitude, and other identifying factors.

The hard part of configuration is understanding what each configuration option means and deciding how important it is in the operation of the server. Some options affect performance, and others can affect your system's security and may require consultation with your system administrator.

The configuration information will be in different locations and may have different names, depending on the version of Gopher server software you use. (In the UNIX version the files to be edited are gopherd.conf, makefile.config, and conf.h.) There also are some command line options when starting the Gopherd program.

Configuring Site-Specific Information

The site-specific information for the Minnesota UNIX Gopher includes such items as organization, site, administrator, latitude/ longitude, time zone, host alias, and language.

Organization is the group that owns the **site**, such as UCLA or Addison-Wesley. The site is the name you want to appear when your server is contacted, such as Social Sciences Computing for my Gopher server. **Administrator** and **admin-email** refer to the name and e-mail address of the person who administers your Gopher server. This information is required under the Gopher+ protocol because it will be listed for every menu item on that Gopher server that does not specifically have its own administrator information. It's common to use an alias such as Gophermaster and have the mail forwarded from there to a real person. *Note:* With this server you can have different administrators for different subdirectories. To do this, see auxconf in the gopherd.conf file.

The 3-D version of Gopher, which is in development, will use

latitude and **longitude** to provide a graphic display of the server's geographic location. Check an atlas or these two geographic information servers to find this information for your location: <<u>http://sipb.mit.edu:8001/geo></u> or <<u>telnet://martini.eecs.umich.edu:3000></u>. You can also display the **time zone**, or the number of time zones your server is from Greenwich Mean Time. Minnesota is GMT-6, Los Angeles is GMT-8, Rio de Janeiro is GMT-3, Taipei is GMT+8, Sydney is GMT+10.

Remember that the machine you put your server on today may not be the machine it's on next year. For that reason it's best to set up a host alias, instead of the specific name of the machine on which it resides. For example, we (Social Sciences Computing at UCLA) originally set up our Gopher server on one machine, but instead of registering that name, we used an alias. Within six months we had to move our Gopher server to a machine with a different name, but the alias stayed the same.

Language identifies the default (or most commonly used) language in the text files on your Gopher server. Like the administrator attribute, this meta-information is passed along by Gopher+ servers whenever item information is requested. A different language can be specified for individual files.

Configuring Gopher Performance

Gopher can be configured to behave in different ways in particular situations. Some options affect the speed with which a Gopher server will respond. The following options are available on the Minnesota UNIX Gopher server but may not have an equivalent in other servers. The **cache time** variable sets the amount of time (in seconds) before the server checks whether the Gopher files have changed. Because it's faster for the Gopher server to grab a cache file of a directory list than to actually run a directory listing, this setting affects the server's response time, particularly for large directories. The number of seconds should be low when you expect rapid change in a directory and high when change is likely to be rare.

You can configure your Gopher server to ignore or not include any files that end with a particular extension or that match a particular pattern. This can be a convenience for the Gopher administrator. ViewExt lets you specify certain file name extensions for different views of a similar base file. For example, message.txt.SP could designate a text file in Spanish, whereas message.txt.DE means text in German (Deutsch). When Gopher+ browsers select this item, they see a list of the different Spanish and German views available for that document. Instead of languages, the different views might be different word-processing formats, such as WordPerfect 5.1, Word for Windows, PostScript, and plain text.

The decoder option allows you to run files with certain extensions through certain programs before delivery. One use is storing files in a compressed form with a specific extension, such as .zip. Then the Gopher server will run the decompression file specified with the decoder option when delivering any .zip files. VeronicaIndex determines whether your Gopher server allows Veronica indexing. Veronica is an extremely useful Internet-wide Gopher index collection system. See the Veronica section later in this chapter for limiting indexing to certain directories.

Configuring Access and Security

One of the first things you need to think about after you have the Gopher server running is whether you need to restrict access. If anyone, for any reason, should not see what's on your server, you should consider whether you want to put that information in a public place like the Internet. Don't trust that the server software in its default configuration will be secure on your system. Pay close attention to exactly what options for limiting access your server software provides, and be sure you understand the method in use. Here are three basic techniques for restricting access to your server:

- Restrict or allow access by domain name.
- Hide your server.
- Require a password for connection.

With these techniques you can restrict part or all of your Gopher server. You can have certain information available only to those coming from certain **domains** and have other information sent to different domains. You can require a password for certain items. Remember that each technique has its limitations.

Restricting or Allowing Access by Domain Name. The first method of restricting access is to allow or restrict access based on the Internet domain or subdomain of the Gopher user's machine. You can also restrict by subdirectory as well as the entire Gopher server. For example, in our Gopher server, <<u>gopher://gopher.sscnet.ucla.edu></u>, sscnet stands for Social Sciences Computer Network, which is in the UCLA network, which is in the edu (education) domain, as opposed to the com (commercial), gov (government), mil (military), or .uk (United Kingdom), .fr (France), .au (Australia), .br (Brazil), and .sg (Singapore) domains. Say I have something on our Gopher server that I want to make available only to people from UCLA. I can restrict access to only those users whose computers are in the ucla.edu domain. Further, if I had something else of interest only to those in the social sciences, I could restrict access to only those people whose machines are in the sscnet.ucla.edu subnet.

This method works when you have a small number of local subdomains that can be easily listed and anything else shouldn't be allowed to see your server. This is appropriate when your company or campus is licensed for certain types of things, but you don't want to publish them for the world.

The restricted access option specifies IP addresses or subnet domains that are or are not allowed access to specific directories or the whole server, whereas you can also limit access by type (browse, read, or search only) for certain machines or subnets. "Bummermsg" is the message that goes out to anyone refused access to your Gopher server because of the restrictions you set up. You can make it friendly, authoritative, or mean like a snake. It's your server, your choice.

Hiding Your Server. Hiding your server may seem silly but actually can be quite effective. Because Gopher servers are expected to run (or be listening) on port 70, hiding your Gopher server means running it on a nonstandard port. Each UNIX machine has thousands of ports. They aren't physical ports, just numbers assigned for different purposes. Port numbers 1023 and below are reserved for the system administrator, but others can use ports with higher numbers, so you'll often see personal Gopher servers listed at port 7000 instead of Gopher's usual port 70. Sites that need different features provided by different Gopher servers often run both but on different ports of the same UNIX machine.

Although the degree of safety obtained by running on a nonstandard port number is not great, it does keep casual browsers out, simply because they would have to try all your ports to find your Gopher server. Of course, once someone sees a link (or URL) to your server, nothing prevents that person from copying it and linking to it. But if you don't publicize this number, it is extremely unlikely that someone will search all possible ports on your machine just to find a hypothetical Gopher server.

Requiring a Password for Connection. Requiring a password (sometimes called a *ticket*) for connection to the Gopher server is basically a low-security method of validating users. Different techniques are discussed more fully in Chapter 7. But for now you should know that this capability is built into at least the University of Minnesota UNIX Gopher server.

Startup: Stand Alone Versus Inetd

(This section applies only to UNIX servers.) Obviously, you want your Gopher server to come up automatically whenever your server is rebooted. This is handled differently on different platforms, but UNIX has two options, stand alone and inetd (Internet Daemon). They determine which part of the operating system has actual control of the Gopher server process.

Inetd is a special process on UNIX that listens for a call on various ports and, when it finds one, starts up the appropriate program, in this case the Gopher server program Gopherd. This would seem sensible, because the Gopher server runs only run when a request comes through, but unfortunately the time delay to start up the Gopher server program, as well as other factors, make this the less desirable alternative.

Standalone mode means that your Gopher server is running all the time, sitting and waiting for someone to talk to it on port 70 or whichever port you've assigned it to watch. The important thing here is to make sure that the Gopher server is not offering too many privileges. You don't want your Gopher server to be subverted by a hacker who gains enough access to damage your system. That's why the UNIX Gopher documentation strongly suggests that you run your Gopher server from a limited privilege account, with only enough privileges to access the directories it needs.

Setting Up the Gopher Data Directory

No matter what version of the Gopher server software you use, you will need an area in which to store files and documents. This is called the **Gopher data directory**, and it's where you'll do the bulk of your work. Each directory acts as sort of a submenu; the files in that directory show up as menu items. You might, for example, have a subdirectory for each department in your organization. These departments might then have subdirectories for different content areas. See Figure 3-1 <fig3-1.htm> for a sample directory structure for the UCLA Social Sciences Computing Gopher. (*Note:* The list of departments is shortened.)

You are not obliged to have many subdirectories, but subdirectories often help to keep things organized and keep your Gopher menus from getting too long. And using different subdirectories for different departments makes it easy to set up group permissions (if your operating system allows it) so that one department can update its own material but not another department's.

<u>Figure 3-2 <fig3-2.htm></u> shows how the first-level directory from the example in <u>Figure 3-1</u> <<u>fig3-1.htm></u>would look in Gopher. Note that the directory and the one file name, welcome, appear in alphabetical order.

Short names for directories and files are convenient for Gopher managers, but users like longer names that include some description. Also, the order here isn't right. The welcome message should be the first item on the menu, not the last, as it appears alphabetically. The UNIX Gopher+ server lets you make such cosmetic changes via the .names file. *Note:* Different Gopher servers use different techniques to make these changes. This section isn't meant as a tutorial but rather as an example of the kind of control almost any Gopher server allows you.

When using the University of Minnesota's UNIX Gopher server software, putting a period at the front of a file name makes that file invisible to Gopher. It also means the Gopher server will check on whether it contains valid instruction statements. These instructions can do three things:

- Give longer names or titles to items (directory and file names).
- Specify the order in which items appear.
- Link to other locations for items.

The example in <u>Figure 3-3 <fig3-3.htm></u>uses the first two of these and shows a .names file setup to give longer names and change the order. <u>Figure 3-4 <fig3-4.htm></u> demonstrates the results of using the .names file to make these cosmetic changes.

Note how much more descriptive that Gopher menu is now. And the welcome file has a longer title and shows up at the head of the list. The Numb=1 command for the welcome file in Figure 3-3 < fig3-3.htm > made that happen. The rest of the items are alphabetized--by their new descriptive titles, not by their file names.

For the final step in this example let's add a link back to UCLA's main Gopher server, so that our users can find their way out to other parts of the campus. Also, the Chicano Studies Research Center has its own Gopher server, and because that's one of the departments we support, let's link to it as well. These links can be added to the .names file or to another file that starts with a period. Let's call our other file .links and keep it separate. Figure 3-5 <fig3-5.htm> shows the .links file, and Figure 3-6 <fig3-6.htm> shows the results of adding the two links.

<u>Figure 3-5 <fig3-5.htm></u> shows how to add information that describes how and where to locate the other Gopher servers. And the Chicano Studies Gopher server is a Gopher+ server, which allows us to add the administrator's name and contact information. This shows only when the user requests information on this link.

<u>Figure 3-6 < fig3-6.htm</u>> shows the two new link items in alphabetical order as items #2 and #5 on the Gopher menu, the result of the Gopher server program's doing its job of coordinating the various files and items to make a useful Gopher menu out of them. One advantage is that you can group your links in different files without having to worry about combining them.

The next step in setting up this Gopher server would be to add files and links to each directory (anthro, econ, geog, and sscnet). The procedure is the same, however, and each directory can have its own collection of .names and .links files. If you need to add more directories at the top or in any subdirectory along the way, you're free to do so. Just check it with a Gopher browser to make sure it shows up the way you want it to.

Useful Techniques for Adding Information and Files

The whole point of the original design of Gopher (and WWW as well, for that matter) was to make it easy for people to publish their information. Gopher succeeds at this better than WWW because there is less emphasis on the format of the material. All you have to do is put your files in a Gopher directory and they become available over the server. It is not necessary to format the document in hypertext.

Eventually, you will find that you don't have time to handle all the files and links that need to be added to your server. Because no one knows as much as you do about the operation of your Gopher server, it is important to have easy, convenient, foolproof methods for others in your company or organization to add information to your Gopher server.

What follows are some techniques for adding information to your Gopher server. Some are simple and some are complex.

Group Permissions

If you are using UNIX or any other operating system that allows rights to be assigned to a group of users, take advantage of that feature to allow individuals to update their own areas of expertise in your Gopher server. Set up a group and then give that group access to the files and directories in its area of the Gopher data directory.

The Macintosh Gopher Surfer server administrator can use the User and Group Privileges features of Apple Personal Filesharing to permit participants in a Macintosh Appletalk network to put their files in your Gopher by just dragging them over with a mouse. Be sure that users know what to do if they need to modify the access rights to the files they add. The limitations of this method are that the users need to know their way around your operating system and they have to be familiar with any specific Gopher server set-up instructions your server may have.

Adding Documents with FTP

On UNIX systems you can give users the rights to FTP files into their subdirectories of your Gopher data directory. It is relatively easy to teach users to do this, and they don't need to learn UNIX commands. And users who become familiar with the .names file format can design and upload that file and control the appearance of the file names and order of appearance.

Linking to Users' Directories

You can create links within the Gopher data directory that point out files in other directories, specifically a subdirectory of the home directory of the person you want to administer that section of the Gopher data. This works on UNIX and may work on any other operating system that allows aliases or links to make a file appear to be in two places at once.

Connecting Gopher to a Database

Although most Gopher servers store their data in files, that isn't necessarily the only way to go. Some Gopher servers are built on top of databases and then use a gateway program between the two. This arrangement allows you to use Gopher to access information in a sophisticated and powerful database package, where it might be used for many other purposes as well. For example, you could keep your company's product information, prices, and inventory in a database that you also make available via Gopher. Instead of scattering files throughout the Gopher data directory, these Gopher servers store at least some data in databases.

This has its advantages, especially if you can link Gopher to an existing mainframe SQL (Structured Query Language) relational database that is already being used for other purposes. This linkup provides broader access to the same data with much less user training. Users find Gopher is much easier to learn than SQL query statements.

The University of Minnesota Gopher Development Team wanted to give Gopher users access to information about campus events that was stored on a mainframe computer. The data was in an SQL database (in this case Sybase), so the team developed the GopherSQL gateway to translate Gopher requests into SQL query statements. The responses from the mainframe database return through the GopherSQL gateway to the Gopher user. This process is described in Paul Lindner's paper presented at the 1994 Gopher Conference, "A Gopher Interface to Relational Databases," which you can find at <<u>gopher://boombox.micro.umn.edu:70/11/gopher/Unix/gopher-gateways/</u><<u><gopher://boombox.micro.umn.edu:70/11/gopher/Unix/gopher-gateways/</u>

Getting Others to Collect Links

Encourage your staff and users to suggest links to you by saving them in their own bookmark files and e-mailing them to you. That gives you a group of people scouring the Internet for related resources, instead of just one. And your staff will feel it is participating in the publishing process.

Adding Links via Forms

Build a Gopher+ form on which users or staff members can enter link information. Encouraging others to gather links for you will bring in more than you'd find on your own. The limitations are that users need to have a browser that is compatible with Gopher+ (forms are a Gopher+ feature) and that anyone can add documents to your Gopher server if you have not installed some authorization or password control options.

Using Veronica to Collect Links

Use the Veronica Gopher Indexing program to search out links to related or useful information. Some Veronica servers allow you to download the entire .link file for the entire search (instead of one item at a time) and add those links to your Gopher server.

Adding Documents via E-Mail (GMAIL)

GMAIL is a Perl script for UNIX Gopher servers that allows your data maintainers to send you their additions and deletions via e-mail (see Table 3-4). This script allows you to create a list of e-mail addresses and match each to a directory in which users can deposit messages. The subject line of the message becomes the Gopher menu item for that file, and the rest of the mail header is stripped off for security and appearance. If you put *delete* in front of the same subject line and send it, GMAIL will delete the message. The script even sends back confirmation messages. The limitation of GMAIL is that each e-mail ID can deposit in only one directory, although each directory can have multiple contributors.

Adding Documents via E-Mail (Appending to a File)

On UNIX systems you can set up an alias such that any messages received are added to the end of a specified file instead of being sent to a person. The University of Minnesota's Gopher server displays these files (they are also called *mail spool files*) as a menu of items, using the subject line of each message as its menu item. This is a good way to add the messages from a mailing list to your Gopher server, if you make that alias a subscriber to the mailing list. But be aware that this file can get quite large if it is not monitored. One solution would be to run a cron job (a program on UNIX that lets you schedule certain events at specified dates and times) so that the file is renamed once a week with the week name. You could also instruct the system to delete the oldest week file to save space. One limitation of constantly appending to a file is that the latest items always appear at the end of the Gopher menu.

GopherLunch

GopherLunch is a Gopher data submission system from Arizona State University (see Table 3-4). It runs on UNIX Gopher servers and allows relatively easy submission of documents via ASK block forms, FTP, or e-mail messages.

Gopher Tools

Whenever you get a bunch of programmers interested in something, you'll soon see that they've developed additional programs and tools that make their job easier. Tools like Jughead were developed to index Gopher servers and allow Boolean searches (logical operators, such as search for this AND that OR something else). Other tools make it easier to link to certain types of items or test your links automatically. These tools can make your life as a Gopher administrator much easier and can allow you to improve your server. Table 3-4 lists and describes some of the more useful tools for Gopher administrators.

Gopher Menu Layout

The layout of a Gopher server is not as simple a project as you might think. In fact, it can be the cause of quite heated discussions. Different camps have good reasons for their opinions. Basically, what drives the controversy is which layout works best to communicate to the user. Typical users of a poorly designed Gopher server might

*Get lost and give up in disgust.

*Falsely assume you don't have what they want when they can't find it immediately.

*Look for something by one name or in one category, when you've grouped or named it differently.

*Have to go so deeply into your Gopher menu that they can never get back or tell anyone else how to get there (of course knowledgeable users would save a bookmark to that link once they had found it).

*Spend too much time following links and menus to get to a popular item that should have been near the top of the menu.

*Find an index search of your server's contents but then have no way to understand the terminology you used in your documents, and all their guesses fail to hit.

Here are some guidelines/rules/suggestions for Gopher menu design:

1.Be clear and descriptive in your menu titles.

2. Avoid vague words like other and miscellaneous.

3.Keep your main Gopher menus to one screen, two at most.

4.Test your menus on novices.

5. Make a searchable index of everything on your server and make it highly visible.

Password Protection

Sooner or later Gopher managers find that they need some sort of password verification to allow only certain users access to particular data files or directories. Providing password protection is a problem endemic to the Internet publishing industry, and some say its solution is the key to making the Internet commercially successful (as opposed to just wildly successful at encouraging cooperation and sharing of information). Needless to say, much research is concentrating on Internet authorization and secure charge-account systems. See Chapter 7, Internet Commerce, for more detailed descriptions of some of these efforts.

However, you should be aware that some Gopher servers offer a "lightweight" security system that exchanges some form of password or token to "prove" the identity of the user. It is called lightweight security because the password is usually sent across the Internet in the clear or with only modest encryption.

Lightweight security systems can be enticing, because they appear to give you at least a certain level of security. But don't confuse the appearance of security with security itself. Although you might be acutely aware of the limitations of your security scheme, your data librarians and others who add items to your server may not always be so cautious.

What Logs Can Tell You

Almost all Gopher servers have some provision for keeping logs, and some utility programs provide statistics so you can analyze those logs. Here's a sample of the information usually available from the log files:

IP address of all clients

Dates and times of connection

Item(s) retrieved (this does not include links to items on other servers, because they would attach to the other server, not yours)

Host names of all clients

These logs make it possible to determine the number of times an item is retrieved and when and its relative popularity. You cannot yet get a reading on the type of clients (Gopher, Gopher+, HTTP, and so on) connecting to your Gopher server. That information would tell you how many of the Gopher+ features you use are beyond the abilities of the Gopher client programs that access your site.

Knowing what files are most and least popular can be useful for measuring utility as well as for redesigning your layout. Perhaps the information you know your users would find most helpful is so far away from the top level that they never find it. Or perhaps the name of the file is not clear enough. You can experiment by listing the file under a new name while keeping the old one. See if the new one gets retrieved more often.

However, you need to be aware of the potential for invasion of privacy inherent in these logs. The logs record the host name (and IP address) of each user's computer, along with the items that person retrieved. Gopher (and Web) administrators often automatically strip off the lowest level of IP address in their reports, precisely to remove this link to individuals. I'm told that at least one Gopher site has had to modify its Gopher server software to remove the possibility of linking this information, because it has a grant that requires absolute anonymity for those retrieving information.

Of course no log actually tells you whether the information downloaded was useful or was even what the user expected. To gather that kind of information you could provide an online form for your users to fill out, asking them specific questions about certain features or selected files on your Gopher server.

Setting Up Forms Using ASK Blocks

Among the most useful enhancements in Gopher+ is the ability to interact with your users. Before Gopher+ the only way for Gopher users to send you a message was by e-mail. ASK blocks in Gopher+ allow the user to fill out a form on your Gopher server online. The information is sent back to your Gopher server to be appended to a file or processed by a shell script or program. Uses of ASK blocks are endless, including asking for comments, filling out an order form, and checking off answers in a multiple-choice questionnaire.

ASK block forms can be built from the following component commands:

*Ask asks the user a question and allows a one-line response.

*AskL asks the user a question and allows a multiple-line response.

*AskP asks the user a question and hides the response (as is commonly done with passwords) so that no one can see the response over the user's shoulder.

*ChooseF asks the user for the name of a local file on her machine. The user can then send this up to the server. This would be a simple way to accept online résumés.

*AskF asks the user to create a new file name on his machine so that the Gopher server can send and store something there. This would allow downloading of data files.

*Select shows the user a set of options. The user can then pick one or more (similar to Macintosh check boxes).

*Choose shows the user a set of options from which the user may select only one (similar to Macintosh radio buttons).

Check out these Gopher+ test sites at the University of Minnesota and the University of Indiana for examples of the latest ASK types. (*Note:* WWW browsers don't accept all Gopher+ features, so use a Gopher+ browser for these sites. For consistency we've given URLs here, but if your Gopher browser doesn't handle them, just connect to the sites listed (*mudhoney.micro.umn.edu*) and (*FTP.bio.indiana.edu*) and follow the menus to the Gopher+ examples.)

<<u><gopher://mudhoney.micro.umn.edu/11/gplustest></u>>

<<u><gopher://FTP.bio.indiana.edu/11/Gopher+/Gopher+test/ask%09+application/Gopher+-</u> <u>menu%20En_US></u>>

ASK blocks are fairly simple. Basically, you create two files with similar names. One lays out the form with questions, types of variables, and so on. The other is the script that handles the information the form provides. If you just want the information appended to a file, the script does that. For example, in a sign-in log you'd want to collect the names in a steadily growing list. But it can do much more complicated things as well. For more information look for the latest version of the document called ASK FAQ version 0.5

<<u><gopher://grace.skidmore.edu/0R50706-69321-/help/Gopher/plus/ask-tips>></u>.

One big drawback to using Gopher+ ASK blocks is that Web browsers don't support them and it's not certain when or if they will. Because Web browsers are a popular method for surfing the Internet, this is a disadvantage.

Security Holes and How to Prevent Them

To think about security you have to understand where your main points of weakness are and what you might be able to do about them. One main concern is that the Gopher program itself not have too many rights. It's important to prevent a user from somehow subverting the Gopher server to get it to perform actions that you didn't intend.

Another area of security is in limiting the directories that are accessible to the Gopher server program. You don't want your Gopher server delivering up your machine's password file, for example. If

the Gopher server can see only the files in your Gopher data directories, no one will be able to trick it into handing over something it shouldn't. And make sure that only those you trust have the right to add items to the Gopher data directories. If, for example, you're laying your Gopher server over an FTP server, make sure that no one can make an anonymous upload. Otherwise, someone could upload a virus into the middle of your Gopher server.

Finally, scripts and programs that are run in response to Gopher requests are always a point of concern. If it's possible to "shell out" or execute other commands via one of these programs, you have to make sure that nothing dangerous can be run via your script.

Different versions of the Gopher server software offer different solutions to these problems, and some may raise additional concerns. Check the relevant newsgroup and list server archives for discussions of these issues, and read through the documentation for the server. Usually, as security problems are found, patches or even new releases are quickly announced. For the Minnesota UNIX Gopher server start by reading "Guide to Safe Gophering" written by Paul Lindner for the 1994 Gopher conference <<u>egopher://boombox.micro.umn.edu:70/00/Gopher/Gopher Conference</u> <u>94/Papers/SafeGopher></u>>.

Remember that you may have security problems simply by being connected to the Internet. Consult a good Internet security text, such as *Firewalls and Internet Security* by William R. Cheswick and Steven M. Bellovin (Addison-Wesley, 1994).

Running Scripts or Programs from a Gopher Menu

You occasionally may want to provide a Gopher menu option that runs a program or shell script on your server and returns the answer to the user. For example, you could have a menu item that asks,

Who is logged onto this machine?

When someone chooses this item, it executes the who program to show the list of users currently logged on and sends the results to the client. This can be used with other UNIX commands and programs as well. This permits you to truly provide access to changing data or information, because your server will reply to the user with the most recent information. Some examples of how a script or program might be used include

1. Checking for the latest weather information

2.Displaying the last 10 comments made by users

3. Checking for what has changed on the Gopher server in the last few days

All kinds of scripts or programs can be run behind the scenes by

the Gopher server--they need not be interactive, but they should make some sort of text response to the Gopher user. This is the mechanism by which gateways to other systems are added to Gopher servers. For example, a WAIS gateway takes a query by a user on a Gopher server, sends it out to a WAIS server, and returns the answer via Gopher.

Different operating systems have different scripting languages; Perl is available for Macintosh, DOS, and UNIX.

Registering Your Gopher Server

Registering your Gopher server is essential if you want others to be able to find it easily and if you want it to be included in index servers like Veronica. The University of Minnesota maintains a master list of Gophers <<u>gopher://gopher.tc.umn.edu/></u>>, which is then referenced by other Gopher lists. You get your Gopher server on this list by sending the university an e-mail message with the following information: server's name, host name, port number, administrative contact, an optional selector string, and an abstract describing the content of your server. According to the Gopher FAQ, if your Gopher server is in Europe, send your e-mail to gopher@ebone.net; otherwise send e-mail to gopher@boombox.micro.umn.edu.

Indexing

Although you will design your Gopher menus for ease of access and the convenience of your users, that's often not enough. As the material you offer grows in size, it increases the size and depth of your Gopher menus. This means people have to spend more time searching through your menus to find the material they want or assume (sometimes incorrectly) that you don't have it.

Minimizing the time users spend and improving the accuracy of their searches are in everyone's interest: yours, because users taking wrong paths tie up your server, and theirs, because they obviously don't want to spend more time searching than they have to.

Indexing the contents of your Gopher server (also known as your section of Gopherspace) is helpful to everyone. It can also compensate for any uncertainties in your Gopher's organization. Gopher indexing usually means that all the menu items and directories in your Gopher site are indexed and available for searching. This allows people to quickly see whether your server has any titles related to their area of interest.

You can get your server indexed in several ways. You can do it, and you can arrange for Veronica to do it. I say *and* because you may well want to index your server both ways.

If you do the indexing, somewhere on your main Gopher menu you'll want to offer users the choice of searching the entire contents of your Gopher site or just going through the Gopher menus. A searcher using Veronica might not know about your Gopher server. Users connect to a central Veronica server to do their search. If the Veronica database shows that your server has something in one of the Gopher menus that matches, the searcher is given links to those items on your server. But if your server is not indexed by Veronica, it doesn't show up in Veronica's database.

The remainder of the chapter briefly summarizes some indexing schemes that are available to Gopher administrators.

Veronica

Veronica is a central indexing service that attempts to index all of Gopherspace each month. It is an extremely useful-- indispensable--method of searching for items of interest in Gopher servers around the world. That's why you want your Gopher server to show up in the Veronica database. Think of it as free advertising for your Gopher server.

Getting your server indexed by Veronica is easy. (For the Veronica FAQ see <<u><gopher://Gopher.scs.unr.edu:70/00/veronica/veronica-faq</u>>.) Assuming your site is not under restricted access, Veronica will know to index your server if you register it with the Mother Gopher at the University of Minnesota or if your server is linked to a Gopher server that is registered with the Mother Gopher.

The biggest problem is deciding which areas of your Gopher server you don't want indexed. Of course, if you don't want any part of your server indexed, that is easy to arrange as well by using a Veronica control file to leave instructions for Veronica. (For detailed instructions and a sample see <<u><gopher://gopher.scs.unr.edu:70/00/veronica/About/Index-control/veronica-ctl</u>>>.) This is a plain text file that is left on your Gopher server for the Veronica harvester to find when it comes by to index your Gopher server. Specific instructions to the Veronica server include

- VeronicaIndex: no--This command stops Veronica from indexing your server.
- *Disallow: Gopher menu*--Use this command (often more than once) to tell Veronica not to index certain submenus.
- *Veronica-datafile: URL*--This command tells Veronica that it can retrieve a prebuilt index. This saves the search-and-retrieval load Veronica would place on your Gopher server in retrieving all your Gopher menus for indexing. These index files can be created by a program such as Jughead, described later.

Note: If you are running the UNIX Gopher+ server from the University of Minnesota, gopherd.conf gives you the option of completely excluding your Gopher server from Veronica harvesting. But the Veronica control file method works no matter which type of Gopher server you are running.

You might think, "Why not let Veronica go ahead and index my entire Gopher site? It couldn't hurt." But you'll be doing yourself and the Internet a favor if you think this through carefully. Are any menus in your Gopher server of local interest only? They might be course lists or assignments for classes or information of interest only to employees of a particular company. Do you have menu items that change so often that indexing them is not useful? Veronica updates only once a month, so if you have items that will be added and replaced several times within a month (like Usenet newsfeeds), Veronica indexing won't be helpful.

The Veronica control file cannot specify individual files, only menus (directories). Therefore it might be worth rearranging your menu structure if you find that some Gopher menus contain items that should be indexed and others that should not.

Jughead

Jughead is an alternative to Veronica that lets you generate your own indexes of your Gopher menus and items, but unfortunately it runs only on UNIX systems. Jughead's designer meant it as a tool for Gopher administrators, but among other things it builds an index of a selected Gopherspace--titles and menu items only, not full text. Like Veronica, it then can act as an index server by listening for search requests at a certain port and responding with the appropriate Gopher item or document. It allows **Boolean searches** (AND and OR) of the indexed material.

One advantage of Jughead is that it was written by a Gopher administrator specifically for Gopher servers, as opposed to some of the other indexing programs, which are more general. One disadvantage is that Jughead runs only on UNIX platforms.

Glimpse

Glimpse is a powerful full-text indexing and query system that allows approximate matching, Boolean searching, and limited **regular expression** (UNIX-style) searching. It allows for three different sizes of indexes, and the larger the index file, the faster it searches. Glimpse does not directly support Gopher servers, but you may be able to make it work as a back-end indexer for Gopher, which is how WAIS works. One advantage of Glimpse <<u><http://glimpse.cs.arizona.edu:1994/></u>> is that development work is continuing, whereas freeWAIS appears to be stalled.

WAIS

WAIS indexing is a powerful way to index the full text of your Gopher server and much more. See Chapter 5 on WAIS for more information.

GN (Gopher/WWW Server)

GN is a combination Gopher and WWW server that has its own indexing scheme built in. <<u><http://hopf.math.nwu.edu:70/></u>>

Summary

Gopher is an inexpensive and simple method for publishing information on the Internet that does not require users to have a fast or powerful computer. It uses a text-based menu-oriented hierarchy that is easy for users to follow. Gopher was an amazing step forward in making files and information available across the Internet to nontechnical people. Gopher's ability to link to items on other computers as easily as to those on the same computer freed users from worries about login IDs, passwords, and protocols when moving from system to system. This, more than anything, opened up the Internet as a true information resource.

Indexing programs like Veronica and Jughead increase the utility of Gopher tremendously, particularly because the results of each search become its own Gopher menu. Unsuspecting users might think the Gopher menu of search results they are seeing (see Figure 1-10 <fig1-10.htm> for an example of a Veronica search) is stored some place on the Internet, but actually it is created at the time of the search. This means that if the user does the same search two weeks later, the menu that appears might be different, because other resources have been added to the Internet and indexed by Veronica in the interim. The dynamic, ever-changing nature of the Internet is one of its most noteworthy characteristics.

Gopher servers and clients (browsers) come in two forms, Gopher+ and plain Gopher, or Gopher0. The enhancements to Gopher+ include such online forms (ASK blocks), the ability to provide different views of the same document, and the addition of meta-information (information about the item, such as date, author, language, and so on). The Gopher protocol was designed with long-term flexibility in mind--Gopher can deal with all types of data.

Gopher server software is available for a wide variety of computer platforms. UNIX once was the most popular, but the use of Macintosh and Windows bases for Gopher servers is growing. Some people run UNIX on PCs and Macintoshes to get faster performance out of their Gopher (and Web) servers. Configuration of Gopher servers varies by the server software used but often includes some sort of access and security options. Usually, they allow you to restrict (or allow) all or part of your Gopher data directories to machines from certain domains of the Internet. This is only useful when those domains can be identified and are precise enough for your needs. Hiding your Gopher server by running it on a nonstandard port is another way to discourage access.

The virtue of publishing via Gopher servers is that information can be made available simply by dropping the appropriate file into the Gopher data directory. The Gopher server automatically adds it to the Gopher menu provided to users. Often, however, particularly with UNIX systems, nontechnical users will want simpler methods than FTP to move their files and information into the appropriate Gopher data directories. There are many alternatives, including GMAIL, which allows e-mail accounts to be used as trusted contributors to specific directories of your Gopher server.

The design of your Gopher server's menu layout is an important factor in making it easy for users to find what they need. Make the menu titles as descriptive as possible, and make the subject categories clear. Indexing your Gopher server with Jughead (or other indexers) is an excellent way to allow users to quickly search the contents of your server to find out if you have what they want. Registering your Gopher server with the Mother Gopher server at University of Minnesota guarantees that it will appear in the master list of all Gophers in the world. Be careful to use the Veronica control file if you do not want all areas of your Gopher server to be indexed.

Security is always an issue when dealing with the Internet, but properly configured, a Gopher server doesn't add to that risk. Be sure to read the security-related discussions on the Internet and in your Gopher server software documentation.

Gopher is said by some to be dying, but it provides an efficient method for making text, graphics, and other files available across the Internet (although not in one document like WWW). Because Gopher browsers can run on even the simplest computers available (a two-floppy PC, for example) and with the slowest of modems, it will continue to serve a purpose long into the future. WWW browsers can also browse Gopher servers, but unfortunately most cannot deal with Gopher+ features. This is unfortunate, and I hope that will change. Gopher is an excellent choice when your main goals are simply to present text and other files for downloading.

FIGURE 3-1 Sample Gopher Data Directory

/gopher-data I-----anthro I-----econ I-----I-----announcements I-----I-----data I------I-----faculty I------geog I------I-----climatology I-----sscnet

Top-level menu items}Appears under econ menuAppears under geog menu

FIGURE 3-2 Sample Data Directory as Seen by Gopher User

Internet Gopher Information Client v2.0.16
Social Sciences Gopher
-->1. anthro/
2. econ/
3. geog/
4. sscnet/

5. welcome [5Mar95, 1kb]

FIGURE 3-3 Cosmetic Touches--.names File Adds Titles and Reorders Items

=====

```
#
Name=UCLA Anthropology Department
Path=./anthro
#
Name=UCLA Geography Department
Path=./geog
#
Name=UCLA Economics Department
Path=./econ
#
Name=Social Sciences Computing
Path=./sscnet
#
Name=Welcome to the UCLA Social Sciences Computing (SSCnet) Gopher
Path=./welcome
Numb=1
#
```

Longer, more descriptive name.Path to file or directory being renamed.Path to file or directory being renamed. This item becomes first in Gopher menu.

FIGURE 3-4 Sample Data Directory After "Cosmetic Touches"

Internet Gopher Information Client v2.0.16

Social Sciences Gopher

-->1. Welcome to the UCLA Social Sciences Computing (SSCnet) Gopher

- 2. Social Sciences Computing/
- 3. UCLA Anthropology Department/
- 4. UCLA Economics Department/
- 5. UCLA Geography Department/

FIGURE 3-5 Adding Links to Sample Gopher Menu

```
#
Name=Info UCLA - UCLA's Gopher Server
Type=1
Port=70
Path=
Host=mic.ucla.edu
#
Type=1+
Name=UCLA Chicano Studies Research Center
Path=1/Directory/Research/UCLA CSRC
Host=latino.sscnet.ucla.edu
Port=70
Admin=Richard
                Chabran,
                            Chicano Studies
                                                  Research Center,
                                                                         UCLA
<chabran@latino.sscnet.ucla.edu>
```

Items with these titles will appear on Gopher menu.

FIGURE 3-6 Menu After Adding Links to InfoUCLA and Chicano Studies

Internet Gopher Information Client v2.0.16

Social Sciences Gopher

-->1. Welcome to the UCLA Social Sciences Computing (SSCnet) Gopher

2. Info UCLA - UCLA's Gopher Server/

- 3. Social Sciences Computing/
- 4. UCLA Anthropology Department/
- 5. UCLA Chicano Studies Research Center/
- 6. UCLA Economics Department/
- 7. UCLA Geography Department/

FIGURE 1-10 Results of Veronica Search for *Gambia*

Veronica: Search ALL GopherSpace by specifying word(s) in Titles: gambia -->1. *** Results provided by Veronica server at University of Koeln: 2. Radio Gambia newsman at WBFO for training program 3. ALL: NEW TRAVEL INFO -- Gambia 4. GAMBIA SITUATION (L ONLY) By KIRK TROY/ABIDJAN 5. The Gambia 6. gambia 7. Gambia 8. GAMBIA and Intl. Marine Science 9. the gambia communications.01 10. Rosevear, D.R. (1937) "Forest Conditions of the Gambia." Empire 11. Background Notes - GAMBIA 12. Gambia.GIF <Picture> 13. GAMBIA / COUP / SUNDAY (L-ONLY) BY KIRK TROY/ABIDJAN 14. Key Officers for GAMBIA 15. IFAD loan to Gambia 16. Fin. & Sub-sect. Framewk. Gambia 17. NEW TRAVEL INFO -- Gambia 18. the gambia people.01

Press ? for Help, q to Quit, u to go up a menu (Note that this is only one of eight screens of search results.) Page: 1/8