Using Gopher with the World-Wide-Web

Paul Lindner
lindner@boombox.micro.umn.edu
GopherCon 95

Hypertext Markup Language (HTML) is now a common document format viewable with a World Wide Web (WWW) Browser. Gopher hierarchies are fast to navigate and let you associate abstracts, alternate views, and other meta information with any type of document. This paper looks at combining the strengths of Gopher and WWW to create a better, more integrated information system. We'll look at how to publish HTML documents from Gopher servers, point Gopher links at documents on WWW servers, and customize the appearance of your Gopher server using HTML.

Introduction

In the past year we've seen an explosive growth in the number of World Wide Web (WWW) software clients that use the HyperText Markup Language (HTML).

Most of these WWW client programs have rudimentary Gopher support. However, most of these clients do not support the Gopher+ protocol. To remedy this the University of Minnesota Gopher Team made many modifications to the Unix Gopher server to support direct HTML and HyperText Transport Protocol (HTTP) access. Many of these modifications were refined over time with the help of the Internet community.[1]

The modifications fall into three broad areas:

- Storage of HTML Documents.
- Internal Changes to support Uniform Resource Locators.
- Automatic HTML generation for Gopher Directories

The World Wide Web

A review of the basic syntax and terms used in the World Wide Web (WWW) will be helpful in understanding the changes to the Gopher software. The three definitions that you must know about are URLs, HTML, and HTTP.

2.1 Uniform Resource Locators

The "Uniform Resource Locator" (URL) was first used by WWW software to describe the servers it could connect to in a uniform way. It has since become popular outside WWW software as a simple, standard way to locate resources. URLs are now internet standards. The Internet Engineering Task Force (IETF) has codified the URL standard in RFC 1630.

A Gopher item can be represented in URL form as

```
gopher://<host>:<port>/<type><path>
```

If the port is the gopher default of 70, the URL can optionally be shortened to:

```
gopher://<host>/<type><path>
```

The "root level" of a gopher server has always been defined as a menu of type 1 with no path required, and it can be represented as:

```
gopher://<host>:<port>/ (or without the :<port> if the port is 70)
```

---

[1] We'd especially like to thank Alan Coopersmith for spearheading a number of these modifications and providing some of the text of this document.

The popular Unix gopher server software from the University of Minnesota uses path strings with the type as the first character, which results in a URL that looks like the type character has been doubled.

In order to make URL's more transportable, spaces and other special characters in paths are "escaped" into the form "%XX", where XX is the hexadecimal ASCII/ISO-8859-1 character code. For more details, see the URL definitions on the Web at

http://www.w3.org/hypertext/WWW/Addressing/Addressing.html

or in Internet RFC 1630.

HyperText Markup Language

HyperText Markup Language (HTML) is a data format used to create hypertext documents that are portable from one platform to another. HTML documents are SGML documents with general semantics that are appropriate for representing information from a wide variety of domains.

Here's an example HTML file:

FIGURE 1. **Sample HTML File**

```
<HTML>
<HEAD>
<LINK rev=made href="mailto:gopher@boombox.micro.umn.edu">
</HEAD>
<BODY>
 <H1>My Home Page</H1>
 <P> Click <A HREF=http://www.w3.org/>Here</A> for more information.
 </BODY> </HTML>
```

HTML document can be written by hand, or generated automatically using a variety of utility programs.

HyperText Transfer Protocol

The HyperText Transfer Protocol (HTTP) is the method whereby most WWW clients move information from servers. HTTP uses many of the same concepts as the Gopher Protocol. Both of them use TCP-IP streams to transmit data, both have a command-response method of operation.

There are currently two flavors of HTTP being used actively, HTTP/0.9 and HTTP/1.0. HTTP/0.9 is slowly being phased out. They are mostly upward compatible however.

Adding HTML documents to your Server.

Simple HTML Files

Adding HTML documents is as simple as copying them to your server's data directory. You just need to make sure that the document ends with the file extension *.html*.

For example, if you had an HTML document in the file *moo.html* you could give access to this file by copying it to the Gopher Server's data directory.

Gopher clients will see this as a HTML document. Gopher clients cannot view these files directly, they must spawn a separate viewer. However, World Wide Web clients can directly view these files.

Using Multiple Views

The Unix Gopher+ server supports the concept of multiple representations of a document. We call this concept 'multiple views' of the same document. Thus we could have a picture available in a number of different graphics formats.

We can use this feature for HTML documents as well. Since most Gopher clients cannot interpret HTML we can make a text version of our HTML documents available by adding another file to our server.

To make this text version available we create a file without the HTML extension. In this file we put the simple text version. For instance, the HTML in figure 1 above could be represented as:

```
          MY HOME PAGE


    Click Here for more information.
    Comments? Send Email to gopher@boombox.micro.umn.edu
```

Note that the text version will lose the ability to connect to the hyperlinks in your HTML document.

We did this conversion by hand. What if we had a lot of large documents that we needed to convert? We don't want to do this by hand, that's much too tedious. Instead we can use the *lynx* WWW browser software to automatically convert our HTML files to text. You can get lynx by accessing the following URL:

<URL:ftp://ftp2.cc.ukans.edu/pub/lynx>

Here's an example of using lynx to convert an HTML file to text:

```
lynx -dump moo.html >moo
```

If you expect your HTML file to change often you can make the 'moo' file a shell script. The following figure provides an example of using a shell script to automatically convert HTML to text:

FIGURE 2. moo shell script

```
#!/bin/sh
/usr/local/bin/lynx -dump ${xxxx}.html
```

When the client user selects the moo file as text the shell script ‘moo’ is executed. This automatically converts the HTML file to text.

Note that you can do this in reverse. You could write your own shell script that automatically converted text files into HTML.

Linking to Remote HTML documents and other URL’s

Now that we’ve put HTML documents on our server we might want to link to remote servers. This is easy enough for Gopher servers, we can use the standard link mechanism.

HTTP servers on the other hand, are a different beast. There is currently no officially defined way to create a Gopher or Gopher+ link pointing at a HTTP server. However, by what seems to be a fortunate accident, the HTTP 1.0 protocol and the Gopher protocol are very similar. In fact, most Gopher clients can be faked into thinking they’re speaking gopher0 when they connect to an HTTP 1.0 server. The primary difference is that a Gopher request consists of just the path, while HTTP requests are of the form “GET / path”. You can use this similarity to create gopher links that look like the following figure:

FIGURE 3. Sample Link to a HTTP Server

```
Name=My Home Page
Type=h
Path=GET /myhomepage.html
Host=www.foo.org
Port=80
```

which fool clients into retrieving the file normally known by the URL

```
http://www.foo.net/file.txt
```

One problem with trying to fool clients like this is that HTTP and Gopher have both been upgraded, and the resultant HTTP/1.0 and Gopher+ protocols are not similar enough for this trick to work. Fortunately, most HTTP servers still support the old protocol, so this method still has some life left in it. (And in fact, there are those who believe this method should become the officially defined way to link gopher to http.)

Older versions of the gopherd 2.0x server can use the “GET /” trick described earlier to make links http servers from their menus. With newer versions however, the following link format is preferred:

FIGURE 4. Sample URL link to a HTTP Server

```
Type=h
Name=Turnip Lovers of America (TLA) Home Page
URL=http://www.turnip.com/~tla/index.html
```

The current software will generate the "GET /" Gopher0 link from this, and it will also send the unmodified URL in Gopher+ menus to Gopher+ clients.

The *URL=* link line can also be used with gopher, ftp, telnet, and tn3270 URL's. The software will take what information it can from the URL to fill in empty spots in the gopher link information. You can explicitly specify any part of the link tuple to override the URL information. The exact conversions are in the gopher source code, but the current URL handling code translates as follows:

FIGURE 5. Default URL Conversion

http://<host>[:<port>]/<path>

```
Type=h
Host=<host>
Port=<port> [or 80 if unspecified]
Path=GET /<path>
```

gopher://<host>[:<port>]/[<type><path>]

```
Type=<type>[1 if type & path are missing]
Host=<host> Port=<port>[70 if not given]
Path=<path>[blank (for "root menu") if not given]
```

telnet://[<user>@]<host>[:port]

```
 Type=8
 Host=<host>
Port=<port>[23 if not given]
Path=<user>
```

tn3270://[<user>@]<host>[:port]

```
Type=T
Host=<host>
Port=<port>[23 if not given]
Path=<user>
```

ftp://<host>/<path>

```
Type=[1 if path ends in '/', 0 otherwise]
Host=+
Port=+
Path=ftp:<host>@<path>
```

Remember, you can override any of these guesses, like this:

FIGURE 6. Overriding Guesses

```
 Type=g
 Name=Miss Turnip Festival 1994
URL=ftp://ftp.turnip.com/pub/miss-turnip.gif
```

Customizing Gopher Directories for HTML

In addition to simple HTML documents, we can use the Unix Gopher Server to display Gopher Directories as HTML documents. The conversion is automatically done by the server, no special scripts are required.

The HTML pages generated in this fashion support many nice features:

They look nicer much nicer.   Abstracts and other Gopher meta-information are viewable.   http: URLs specified in link files are passed correctly.

Accessing Custom HTML pages.

Most WWW clients can connect to gopher servers and display gopher menus. The URL's used when retrieving a Gopher menu from the Unix gopherd look like:

```
gopher://<host>:<port>/11/<path>
```

Since these menus are just text/html views of a gopher+ object, the above method can be used to access these menus by merely changing the "11" in the URL to a "hh". (The first h tells the client what to expect, the second h is part of the request sent to the server.) For instance, some server administrators use links like this:

```
Type=h
Name=WWW Hypertext version of this gopher server
Path=h/
Host=+
Port=+ #
```

which allows you to switch to the HTML versions of menus when you select that item.

The 2.1.4 release of the University of Minnesota Unix Server also supports more direct access via a URL that starts with *http://* It does this by listening on both the Gopher and HTTP ports (by default 70 and 80)

To configure this in your server you can add the following line to your gopherd.conf file:

```
HTTPport: 80
```

This will tell the server that you want to listen for HTTP requests on port 80 (the standard HTTP port.)

You could then put a link on your server like this:

```
Name=This server viewed as HTML
URL=http://thishost/
```

Default Representation of Gopher Directories

The default representation of a Gopher directory in HTML is very similar to the normal Gopher directory. The documents are listed in the same linear fashion, one per line. This, by itself, is nothing to jump up and down about. There are some extra features that you might not notice right away:

1. The Server will put any abstract information directly below the text.
2. For the top level Gopher directory the server puts up a banner that includes the name of the site and organization.
3. The 2.1.4 release puts an icon to the left of each object. These icons can be customized.
4. Gopher type ‘i’ info items are added as inline text. Gopher and HTML users can see these text items.

The following figure is an example Gopher directory viewed as HTML.

FIGURE 7. **Simple HTML Gopher Directory**

Distributed Computing Services

University of Minnesota

Information About Gopher
Computer Information
This item contains vendor information from Apple, IBM and more!

Customizing with Abstracts

You can edit files that end with.*abstract* to add text below a directory entry in the HTML page. Abstracts are automatically inserted underneath the directory entry in the HTML page.

For instance, if we have a file called *moo* we can put the information about moo in a file called *moo.abstract*. This automatically maps to the Gopher+ ABSTRACT block.

You can also add abstracts (and text in the HTML page) by adding *Abstract*= lines to your links files. This even works for remote servers. Here's an example:

FIGURE 8. **Abstracts in a Links file**

To get multiple line abstracts you have to put a backslash at the end of the line.