Running WWW on top of Gopher

Mike Potter
Network Engineering
C-5, MS B255
Los Alamos National Laboratory
Los Alamos, New Mexico, USA 87545
mep@lanl.gov

Presented at WWW '94, CERN, May 1994

Running WWW on top of Gopher

paper for WWW'94, CERN, May 1994 and GopherCON '94, Minneapolis, Minn, April 1994

Mike Potter
Network Engineering
C−5, MS B255
Los Alamos National Laboratory
Los Alamos, New Mexico, USA 87545
mep@lanl.gov

At Los Alamos National Laboratory (LANL), we wanted to set up a World−Wide−Web (WWW) server. A few months earlier we had already set up a Gopher server. Rather than maintaining two separate systems, with a plain ASCII copy of the information in one system along with an HTML version in the other, we decided to write a WWW server that would make use of the existing Gopher information structure.

Of course, you don't need to do anything special in order to view Gopher information via the Web, you can simply use the URL of the form

```
gopher://nodename/type/path
```

to point to your existing Gopher server. However, we wanted the ability to enhance the existing information using HTML, without having to duplicate all of it. The following goals lead to the development of the *gopherhttpd* server:

- Seamless integration of Gopher and WWW from the perspective of both the users and the information provider. If a new document is added to the Gopher system, it should automatically appear in the WWW system.
- Provide a WWW server that installs and operates much like a Gopher server to allow existing system and network managers to install a WWW server with little additional training.
- Provide the ability to completely override ASCII documents in Gopher with HTML documents in WWW when desired.
- Allow easy annotation of the existing Gopherspace.
- Provide nicely formatted HTML pages by default.

This document describes the *gopherhttpd* server that achieves all of the above goals. The installation and operation of this server will be described, and examples from the LANL server will be shown.

Overview of *gopherhttpd*

The easiest way to understand the power of *gopherhttpd* is to take a look at an example.

Here is what the LANL Gopher server looks like.

Here is what the LANL WWW Home Page looks like.

The LANL WWW Home Page was generated "on the fly" based upon information from the LANL Gopher server. Note that all of the Gopher menu items appear in the WWW Home Page. Each menu item is "annotated" with a short description. In addition, the entire menu is preceded by a short description (like an "About" or "README" file would normally be used for in Gopher).

Note that the WWW Home Page takes full advantage of HTML, including the ability to present a logo.

A Closer Look

Let's take a closer look at each element in the LANL WWW Home Page and see how it is generated from the data on the Gopher server.

<Optional logo>

This text is used to describe the current page. It is optional. The contents of this section are taken from the file `README.html` in the current Gopher directory. This file can contain any HTML code that is desired.

<optional icon>First Menu Item
    Description of the First Menu Item. This section is optional.
<optional icon>Second Menu Item
    Description of Second Menu Item

_____
Author Name or email address

Each of the Menu Items in the template shown above are taken from the Gopher menu. The *gopherhttpd* server reads both the `.cap` files and the `.Links` files and produces a menu just like the Gopher menu. Each menu item corresponds to a specific file or directory in Gopherspace. The description of this file or directory is read from the file `filename.about.html` where `filename` is the name of the Gopher file or directory. The optional icon is taken from the file `.cap/filename.gif` if it exists.

The only thing that was done to the Gopher server to make all of this work was to add the line

```
ignore:          .html
```

to the `gopherd` configuration file. Thus, any file ending in `.html` will be ignore by the Gopher server, and handled instead by *gopherhttpd*.

The Author name or email address shown at the bottom of the screen is taken from a file in the current Gopher directory called `AUTHOR.html`. If this file is not present, and is not found in any parent directories, default information contained in the *gopherhttpd* configuration file is placed here instead. Since this information can contain any HTML commands, you can provides links to your phone book entries.

That's it! This is the basic template for every HTML page created by *gopherhttpd* from your existing Gopher menus.

Further Customization

There are some tricks that you can use to further customize the look of your WWW server.

Normally, *gopherhttpd* will automatically create the URL for each item in the menu to point to either another menu, or to the information file itself. If the file `filename.html` exists, then a link to this file will be generated instead. This allows you to completely override a file in Gopher with an HTML file by putting both into the same Gopher directory. Since Gopher ignores HTML files, Gopher users will only see the original data, and WWW users will only see the HTML data.

You can prevent Gopher items from appearing in the WWW page by adding a file to the Gopher directory called `filename.ignore` Of course, you then also need to tell your Gopher server to ignore files that end in `.ignore`. To avoid this, *gopherhttpd* also recognizes the file `.cap/filename.ignore`. Now Gopher users will see the original `filename`, but *gopherhttpd* will omit it from the WWW menu.

Extensions to the Gopher .cap files

Instead of creating description files using the `filename.about.html` method described above, you can achieve this same effect through some extensions to the Gopher `.cap/filename` and `.Links` files. Gopher clients ignore lines in these files that do not start with recognized keywords. Thus, we have added some new keywords that *gopherhttpd* will recognize, but Gopher will ignore. Here is a list of the new keywords:

`Desc`
: Allows you to specify the description to be placed under the menu item name. Only a single line description is allowed, but that line can be as long as you want, and Mosaic will wrap the long line as needed into a full paragraph.

`WWW`
: What if a system is running both a Gopher and a WWW server? Normally, *gopherhttpd* points to the Gopher system. You can override this using the `WWW=url` option in the `.cap` file. The `url` specified on this line will be used as the menu item. Also, a line in the description field will automatically be

generated that says

> You can also access their Gopher server.

which points to the Gopher system. If you put an asterisk (*) at the end of the URL, this Gopher message will be supressed. A dollar−sign ($) in the URL is expanded from the `Host` field. If the protocol string is missing from the front of the URL, `http://` is added by default. Thus, the string

```
WWW=$/welcome.html
```

will generate a menu item with a link of `http://hostname/welcome.html` where `hostname` is taken from the `Host=hostname` line in the `.cap` file.

`Pre`
: HTML text assigned to Pre will be output immediately before the highlighted Name of the menu item.

`Post`
: HTML text assigned to Post will be output immediately after the highlighted Name of the menu item.

`Before`
: HTML text assigned to Before will be also output before the menu item, but also before the `<dt>` that flags the start of the menu item. You can think of this text as appearing after the Description text of the previous item. A common use of this item is with `Before=<hr>` to put a horizontal rule before the menu item.

NOTE: When using extended syntax in the `.cap/filename` file, be sure and put the new items after any existing Gopher items. Many gopher clients abort their parsing of the `.cap` file when they reach an unknown keyword. Thus, all standard Gopher keywords should come first, followed by the extended *gopherhttpd* keywords.

Putting .cap information into the About file

As mentioned in a previous section, you can use a file called `filename.about.html`, where `filename` is the name of the Gopher file or directory, to specify the description that appears under the menu item. This is just the default use for the About file −− you can do much more. If a line in the About file doesn't start with `keyword=`, then it is assumed to be a line in the menu description. A single line in this file is equivalent to putting a `Desc=description` line in the .cap file.

However, in the About file (the `filename.about.html` file), you can specify multiple description lines. In the HTML output, a `<dd>` is inserted at the beginning of each line to force a line break in the description.

You can also put any valid `.cap` information into the About file. Any information specified in the About file will override information in the `.cap` file. This allows

you to further modify and customize your WWW page since you can change the Gopher information in the *gopherhttpd* About file for a given menu entry.

Some real examples

Let's look at a real–life examples where we make use of some of these advanced features.

Separating menu items into sections

In order to organize a Gopher menu into sections, sometimes you will see gopher sites do this:

```
                  Internet Gopher Information Client 2.0 pl10

                      Root gopher server: gopher.lanl.gov

 -->  1.  News Flash 7-Mar-1994: What's New in the LANL Gopher....
      2.  ----------------------LANL Information---------------------.
      3.  News and Events/
      4.  Phone Book/
      5.  Job Openings/
      6.  Library Catalogs and Information/
      7.  Computing at LANL/
      8.  Information Architecture Project/
      9.  Software Archive/
      10. Information by Division/
      11. Information by Subject/
      12. ------------------------The Internet----------------------.
      13. About the Internet/
      14. How to get Gopher/Mosaic Software/
      15. The Internet via Gopher/Mosaic/
      16. Finding People, Places, and Information/
      17. Selected Software Archives (FTP)/
      18. Network News (USENET)/

Press ? for Help, q to Quit                                  Page: 1/2
```

What we want the WWW home page to look like is something like this:

News Flash 7–Mar–1994: What's New in the LANL Gopher....

LANL Information

- News and Events
- Phone Book
- Job Openings
- Library Catalogs and Information
- Computing at LANL
- Information Architecture Project
- Software Archive
- Information by Division