Chapter 1: WWW Daemon user guide

The http daemon, httpd , is a general server program which runs a w3 protocol, " HTTP ".

1.1 More Information

Distribution	How to get the code.
Compilation	How to compile the daemon for your system.
Installation	How to install a server under unix internet daemon
Options	Command line options at run time
Rule File	The format of a rule file. By default, /etc/httpd.conf
Debugging	If it doesn't seem to work
Known bugs	and improvements desired
History	change list of improvements made and bug fixes.

1.2 Note

The IANA-allocated port number for HTTP servers is 80.

Background: The W3 project , Other servers .

Tim BL

1.3 The basic W3 server: Internals

This describes the generic hypertext daemon (server) program. The daemon is part of the WWW project. See also:

- User guide .
- Bugs and Features
- Other servers

The hypertext daemon, like the ftp daemon, is a program which responds to an incomming tcp connection and provides a service to the caller.

1.3.1 Sources

A compilation option (SELECT) controls whether more than one connection can be handled at a time. This is a function of whether the TCP/IP implementation beneath the application has a working "select()" routine. If it is not true, this implementation services one connection, then drops it before accepting another one. In neither case does the daemon concurrently serve two clients, nor does it fork off a process to do that.

The basic server loop is in the file HTDaemon.c . A separate module (for example HTRetrieve.c) contains the code to handle one request. Various specific versions of this may be written for different flavours of server. Also used are various modules of WWW common code. Tim BL

1.4 Status of basic WWW server

A basic fast information server for files.

Author	TBL
Status:	Version 0.4 available by anonymous FTP, with no index search but file access, name mapping and security filter.

Plans:	A version of the WWW daemon which will allow general unix users to set up an index search daemon. As index search tools are not generally available, we may use the NeXT digital Librarian or WAIS as an basis
Platforms	Unix, VMS, VM/CMS (VM/XA).
Next Milestone:	Serious improvements to installation ease, and simplicity of publication. Directory browse. Run shell scripts to implement virtual documents and searches.
More information:	Overview ,User guide, Bug list , features .
Wider scope:	Other WWW software

1.5 Installing the basic WWW server

The usual way to install a daemon is to either run it from the bootstrap command file (for example /etc/rc) so that it runs continuously, or to set up the internet daemon (inetd) to run it when a call comes in.

See a csh script which does everything below for unix BSD systems.

Remember to make a rule file unless you explicitly supress it with the -R option.

1.5.1 Using the Internet daemon

There is a csh script to do this for a BSD unix system. You might want to edit it to run on other systems.

Otherwise, just follow the instructions for installing a server under the internet daemon .

1.5.2 Log file

If a log file is required, make sure that the user name under which the daemon is run has the right to write the file Tim BL

1.5.3 Priviliged ports

The TCP/IP port numbers below 1024 are special in that normal users are not allowed to run servers on them. This is a security feaure, in that if you connect to a service on one of these ports you are fairly sure that you have the real thing, and not a fake which some hacker has put up for you.

The normal port number for W3 servers is port 80, which is such a port. (This number is assigned by the Internet Assigned Numbers Authority, IANA).

When you run a server as a test from a non-priviliged account, you will normally test it on other ports, such as 2784 or 5000 typically.

Under unix

The inet daemon (running as root) can listen for incomming conections on port 80 and pass them down to a process with a safer uid for the server itself. Of course, you have to be root to set up the inet daemon.

Under VMS

Under UCX, The process running as a server needs BYPASS privilege to listen to ports below 1024. This might mean you have to install the server. With other TCP/IP packages, privilege of some sort is similarly required.

Tim BL

1.5.4 Adding a service on the NeXT

The NeXT uses the the "netinfo" database instead of the /etc/services file. This is managed with the /NextAdmin/NetInforManager application. Here's how to add the service "www":

- Start the NetInfomanager by double-clicking on its icon.
- If you are operating in a cluster, open either your local domain (/hostname) or if you have authority, the whole cluster domain (/). If you're not in a cluster, just use the domain you are presented with.
- Select "services" from the browser tree.
- Select "ftp" from the list of services
- Select "dupliacte" from the edit menu.
- Select "copy of ftp" and double-click on its icon to get theproperty editor.
- Click on "name" and then on the value "copy of ftp". Change this to "www" by typing "www" in the window at the botton, and hitting return.
- Click on "port", and then on the value "21". Change it to "80".
- Use "Directory:Save" menu (Command/s) to save the result. You will have to give a root password or netinfo manager password.

Tim BL

1.5.5 The Rule File

The rule file defines how the server will translate a request into an actual filename. It allows one to provide an extra level of name mapping above that given by links in the file system. It allows, for example, out of date names to mapped onto their more recent counterparts.

The rule file also allows access to be restricted. This is essential, to prevent, for example, unauthorized access to your password file.

By default, the rule file /etc/httpd.conf is loaded, unless specified otherwise with the -R or -r options .

See example rule files .

Format

Each line consists of an operation code and one or two parameters, referred to as the template and the result. Anything on a line after and including a hash sign (#) is ignored, as are empty lines.

The server uses the top rule first, then each successive rule unless told otehrwise. The operation codes are as follows

map template result If the address matches the template, use the result string from now on for future rules.

pass template If the address maches the template, use it as it is, porocessing no further rules.

fail template If the address matches the template, prohibit access, processing no futher rules.

The template string may contain at most one wildcard asterisk ("*"). The result string may have one wildcard only if the template has one.

When matching,

- Rules are scanned from the top of the file to the bottom.
- If a request matches a "map" template exactly, the result string is used from then on.
- If the request maches a "map" template with wildcard, then the text of the request which matches the wildcard is inserted in place of the wildcard in the result string to form the translated request. If the result string has no wildcard, it is used as it is.
- When a substitution takes place, the rule scan continues with the next rule using the new string in place of the request.

Tim BL

Chapter 2: Rule file examples

A basic rule file for the http daemon might look like this:

```
map    /          /welcome.html
map    /*         /u/john/public/*
pass   /u/john/public/*
fail   *
```

The first line maps the root document onto a specific document about the server.

The second line maps all document names onto filenames in a particular directory.

The third line allows access to all documenst in that directory.

The fouth line disallows access to all other documents. (There won't be in any in this case because of the mapping, but its wise to put in for later).

2.1 Second example

```
map    /            /tnotes/welcome.html
map    /tnotes/*    /u/john/public/*
map    /seminars/*  /u/jane/seminars/*
pass   /u/john/public/*
pass   /u/jane/seminars/*.html
fail   *
```

The first line maps the root document onto a specific document about the server. (See etiquette)

The second line maps all document names starting with /tnote/ onto filenames in a particular where john maintains the technical notes. If someone else takes over the technical notes, we can change this. Here we are starting to distinguish between document names and file names. This can be carried much further if necessary, but one level of mapping is enough to allow for changes of administration of different areas.

The third line separately maps the seminar information into Jane's directory.

The third and forth line enable access to anything in John's "public" directory, and any .html file in Jane's "seminar" directory tree. Note here that the * maps to any sequence INCLUDING SLASHES so all files in any subdirectory of /u/jane/seminars will be enabled so long as they end in .html.

The bottom line will pick up for example any attempt to use the server to access non-html files in Jane's seminar directory.

2.2 Installing a daemon under inetd

The usual way to install a daemon is to either run it from the bootstrap command file (for example /etc/rc) so that it runs continuously, or to set up the internet daemon (inetd) to run it when a call comes in.

There is a csh script to do all this for BSD systems, which should be run from the directory in which you have compiled the daemon.

2.2.1

Copy the daemon program or shell script (httpd in this example) into a suitable directory such as /usr/etc/httpd. Protect it from anyone writing to it except root.

Put "http" in the /etc/services file, or use the name of a specific service of your own if you want to use have a special port number.

(Exceptions: on a NeXT, see using the NetInfomanager . On any machine running NIS (yellow pages), see specicial instructions).

For example, beginverbatim www 80/tcp # WorldWideWeb server

endverbatim Put a line in the internet daemon configuration file, /etc/inetd.conf. For example,

```
www stream tcp nowait nobody /usr/etc/httpd httpd -l /private/adm/httpd-log.txt
```

(That was all one line.) Here "www" is used as a link between the services file and inetd.conf: it could have been any identifier. "nobody" is the user name under which you want the daemon to run, which determines what privileges it has for example to read data. The inted.conf format varies from system to system so copy the format of other lines in your version.

Note: there seem to be, on the NeXT at least, a limit of 4 arguments passed across by inetd!When you have updated inted.conf, find out which process is running inetd, and send it a "HUP" signal. On BSD unix (For system V, use ps-el for ps aux) this looks like: beginverbatim

> ps aux — grep inetd — grep -v grep root 85 0.0 0.9 1.24M 304K ? S 0:01 /usr/etc/inetd > kill -HUP 85 >

endverbatim

2.2.2 Also remember for httpd:

You must make or copy a rule file /etc/httpd.conf, unless you have used the -r option to change its location, or -R to disable it.

The -r option tells the daemon which rule file to use. See rule files .

If a log file is required, make sure that the user name under which the daemon is run (the 5th field in the line above) has the right to write the file.

Test the server with the line mode browser by giving its address explicitly: beginverbatim www http://myhost.dom.ain:80/junkdocumentname

endverbatim If it doesn't work, you have probably missed something. See notes on debugging . Tim BL

2.2.3 Using NIS (Yellow pages)

If your machine is running Sun's "Network Information Service", originally know as 'yellow pages", read this.

You must:

- First make an addition to the /etc/services file just as for a normal unix system.
- Then, change directory to /var/yp and type "make".

This will load the /etc/services file info the yellow pages information system.

Some peopl ehave found that they needed to reboot he system afterward for the change to take effect. Tim BL

2.3 Daemon Command Line

The command line options for the simple www server are as follows.

-a address Specify the port number. The address should be in the form "*:80" with 80 replaced with the desired port number, which can be different for tests for instance. Note that the string must be quoted because of the asterisk. If this option is not given, the daemon assumes that it has been run by inetd, and uses stdin and stdout as its communication channel . Note that port numbers under 1024 are privileged .

-l file	Log all calls to the given file. The file is appended to if it already exists.
-v	Verbose mode. Copious trace messages are written to the standard output stream. Mainly for debugging.
-r file	Load a rule file . The rules are added after any rules already loaded. Inhibits the loading of the default rule file.
-R	Inhibit the loading of the default rule file. Warning: running without a rule file normally poses a security problem.

2.4 Compiling httpd

The daemon files come in a tar file, which you should uncompress and unwrap. There are also binaries available for some architecures: if you have one, already, skip to installation.

In the WWW/Daemon/Implementation directory created are several subdirectories, for each machine architecture. Each has a Makefile. If your machine type (eg sun4, NeXT) is not there, copy one that is and its Makefile.

Edit the Makefile to set up particular paths you use on your system.

Type make httpd

Go on to install the deamon under the inetd or in /etc/rc to run continuously.

Tim BL

2.5 Running under shell

You don't have to run the daemon under the inted if it doesn't work. You can run it from a shell session.

If the daemon is httpd, then run it from your terminal, with a different port number like 8000. You use the -a option .

```
httpd -v a \{\char'134\}*:8000
```

Note: You must be root (under VMS, have some privilege) to run with a port number below 1024. You can't use a port number which has been used by a daemon process recently, so you may have to switch port number if you Ĉ and restart the daemon.When it is running like this, you can read the trace messages and use a debugger on it if necessary. (See also: telnetting to the server)

If the server is a shell script "myserver", try just running it with the terminal and typing GET /documentname into its stdin.

Tim BL

2.6 Document address invalid

You have accessed a W3 server and you get back a message "Document address invalid or access not authorized".

The server does not (originally for security reasons) distringuish between a document which does not exist, and one to which you are not allowed access. However, most server are public servers which allow access to anyone, so if you are following a bona fide link, this could mean

- You have been passed a bad document address. If you are following a link, check with the author of the document which contained the link.
- The document has been moved. Check with the server administrator. You should be able to find out who runs the server by going to the welcome page (type "g /" with the line mode browser) and seeing a link to information about the maintainers.

If you are the server administrator, and you can't understand why the daemon refuses to deliver the file,

- Check the rule file .
- Try running the daemon with trace from a terminal session

Tim BL

2.6.1 Debugging the daemon

Suppose you think you have installed a W3 server but it doesn't work. That is, you have followed the installation instructions and the test at the end fails. Here we assume you have used port 80. If you have a situation not handled by this problem-solving guide, please mail me.

Type

```
www http://myhost.domain:80/
```

What happens?

- "Cannot connect to information server" message .
- An empty document is displayed
- A document containing the words "Document address invalid or access not authorised" is displayed.
- A document is displayed, but not what you wanted the server to give in response to that document name (/)

Tim BL

2.6.2 Can't connect to server

There is more information you can get. use the "verbose" option on the browser to find out what went wrong:

```
www -v http://myhost.domain:80/
```

What do you get? A load of trace messages. There are several cases.

- The browser can't look up the name of the host. If it can, it will display "Parsed address as" message. If not, try fixing your name server or /etc/hosts file, or quoting the IP number of the host in decimal notation (like 128.141.77.45) instead.
- The browser can get to the host but gets "Connection refused" status back .
- Your browser gets an error number but prints "error message not translated". This is because when it was compiled on your platform it didn't know what form the error message table took. Try the same thing form a unix platform for example.
- You get some network error like "network unreachable". Depending on whether the IP network is your responsibility or not, and your attitude to life, either fix it, try again in an hour's time, or complain to someone.

Tim BL

2.6.3 "Connection Refused"

The browser tries to connect to the daemon but gets this status in the trace.

This means that noone was listening on that port number. Check the port numbers match btween server and client. Make sure you specify the port number explicitly in the document address for www.

If you are running the daemon without the inet daemon, (with the -a option) then try running it from the terminal with -v as well. The trace for the server should say "socket, bind and listen all ok". If it does, and you still get "connection refused", then you must be talking to the wrong host (or, conceivably, different ethernet adapters on the same host)

If you are running with the inet daemon, then check both the services file (/etc/service) or database (yellow pages, netinfo) if your system uses it, and the /etc/inetd.conf file. Check the service name matches between these two.

Did you remember to kill -HUP the inet daemon when you changed the inted.conf file?

Try running the deamon from a shell window to see what happens better. Tim BL

2.6.4 You get an Empty document

The document sent back is empty, but there is no error message.

The inet daemon has started a process to run your server but it immediately failed. Possibilities include:

- The daemon may not be in the file specified, or may not be executable by the specified user (or, if user id not specified in your variety of inetd.conf, root)
- The rule file does not exist.
- You have written your own daemomn and it crashes.
- You are using ours and it crashes (mail us!)

Try running the daemon from a terminal window to see what happens.

2.6.5 Bad output from the daemon

These are some ideas:

- Try running the server from the terminal .
- Check the HTML source the daemon produces with

```
www -source http://myost.domain:80/
```

- Try telnetting to the daemon and simulating the client:

```
> telnet myhost.domain 80
Connected to myhost.domain on port 80
Escape is \^[
GET /documentname
```

Tim BL

2.6.6 Telnetting to a server

Most implementations of telnet allow you to specify a port number. Under unix this is often just a second parameter, under VMS a /PORT option.

The HTTP protocol is a telnet protocol, so you can simulate it just by typing things in. This will help you to see exactly what a sending back, and it will check you that it really is the server not the browser which has a problem.

Here is an example. (You type "telnet..." and "GET ...").

```
> telnet myhost.domain 80
Connected to myhost.domain on port 80
Escape is \^[
GET /documentname
<PLAINTEXT>
Document name "/documentname" invalid.
```

Chapter 3: W3 Server Software

A W3 server, like the ftp daemon , is a program which responds to an incoming tcp connection and provides a service to the caller.There are many varieties of W3 server software to serve different forms of data.

3.1 The basic daemon

The basic W3 daemon program serves files already in hypertext or plain text. This daemon then is used as a basis for many other types of server and gateway. Documentation includes:

- User guide .
- Internals – a description of the code.

Much of the above may also apply in whole or part to other servers mentioned below. Whatever server you are running, you will probably be interested in:

- Tools for information providers

3.2 Making a new server

This daemon is often used as a basis for a more specific server for a given application. A server which allows a world of data to be seen as part of the W3 universe is known as a gateway. (Most servers could therefore be regarded as gateways, but the term implies some conversion or mapping between dissimilar worlds) . For short tutorials with examples, see:

- Writing a server in C
- Writing a server as a script

It is a good idea to pick the basic daemon or one of the servers in the list as a starting point when making a new server.

3.3 Other servers and Gateways

These are servers which provide data extracted from other systems. they are built using code from the basic daemon, or scripts. See

- List of Gateways available.

Tim BL

3.4 Bugs and Improvements needed

Improvements to be made in the HTTP daemon program are as follows. (Se also Features)

- The trace options are needed
- Call shell scripts to perform searches on directory trees or documents.
- Browse directories.
- The server still uses the <PLAINTEXT> deprecated tag.
- The HTRetrieve() routine ought to be able to pick up the user node and userid, etc...

Tim BL

3.5 Daemon features: Update history

History list for the WWW daemon . (See also bugs)

3.6 unrealeased

- Bug fix: If a PASS or FAIL line in the configuration file acted on a single document id (ie no wildcard) then it crashed the daemon. (HTRules.c, 17-Jun-92, TBL).

3.6.1 Sept 1991 v0.3

- Bug fix: Plain text files were returned to be parsed as SGML, causing them to come out as garbage. (Mike Sendall)

3.6.2 August 1991 v 0.2

- -R option now suppresses default rule file.
- Rule file format changed completely. Now allows authorisation of specific paths only.

3.6.3 June 1991 version 0.1

- -r and -R options for rules
- Default address is now for Inet daemon working. (29 June)
- -l option to log to a file.
- -a option for address other than default

Chapter 4: Gateway Servers

See also: W3 server software

These are servers which provide data extracted from other systems. they are built using code from the basic daemon, or scripts.

FIND gateway	for CERN/VM XFIND which calls a REXX exec to get the information from the XFIND system running on the CERNVM mainframe.
Hytelnet gateway	A gateway to Peter Scott's list of telnet sites
VMS Help gateway	This allows any VMS help files to be made available to WWW clients. Runs on VAX/VMS.
WAISGate	A gateway to information available using the W.A.I.S. protocol.
DCLServer	A server for VMS systems which allows you to write a gateway to your own favorite information system using DCL.
System33	A (big) csh script server providing data including Xerox System33 documents, man pages in plain text, phone numbers, etc. etc...!
Oracle	A generic server to oracle. Could be used as a basis for gateways to specific Oracle databases.
Geography	Gateway to the Geography server at U Michigan
TechInfo	TechInfo is the CWIS from MIT. A gateway exists thanks to Linda Murphy/Upenn.

Tim BL

4.1 Geography gateway

Wed, 18 Nov 1992

Jim Davis <davis@dri.cornell.edu> Here is a quickly hacked up Gateway from WWW to the University of Michigan Geography server. It expects one argument, a WWW doc id. It ignores the "pathname", extracts the search words, then passes those to the server. It does NOT parse the data returned by the server (that is an improvment yet to be done) but you can understand the output.

To use this, you would need to have an HTTP server running someplace where you can attach this gateway. I can provide the very simple HTTP server I use here, but this subject is already documented in the WWW online documentation.

Source code in perl

4.2 The WWW TechInfo gateway

This is a gateway built using the basic server code, plus one source file in C. Thanks to Linda Murphy of Univerity of Pennsylvania for the etchinfo code.

- The gateway data as running at CERN
- The source file

Tim BL

4.3 The W.A.I.S. - WWW gateway

This is an example of a WWW server and a WAIS client. It uses the common WWW daemon sources and also has one extra file, WAISGate.c The command line syntax is the same as for the normal W3 daemon, except that a rule file is not used. (No -r or -R options).

See a summary of some data available through the gateway .

4.3.1 WSRC files

The gateway keeps a cache of WAIS "source" files. These are files

describing WAIS servers. They are normally picked up automatically by searching a "directory of servers" index. Once the gateway has picked up a desciption of a server, it uses the description to describe the server to those who follow links to it.

These source files are converted into hypertext, and are kept in the directory /usr/local/lib/WAIS under the server name, port, and database name.

To add an extra source file to this cache by hand, feed it as standard input to the ParseWSRC program.

4.4 Gateway to VMS Help: Internals

These are technical and installation notes about the gateway to VMS Help . Please send bug reports and suggestions to Jean-Francois Groff (jfg@cernvax.cern.ch).

4.4.1 Sources

The program consists of the generic daemon HTDaemon.c , and a special function, stored in VMSHelp-Gate.c , to retrieve VMS Help data and convert it to HTML.

4.4.2 Installation

The files you need are as follows. You should customise them, putting in your own directory names.:

launchgate.com Runs the server as a detached process. Put a call to this from your sys$startup procedure, wherever that is. This detaches a job to use www_server.com ans input, and a log file as output.

www_server.com The server command file, a wrapper for the actual server executable. In this file, set the temporary directory for the storage of a cache of .HLP files. This file runs the executable.

test.com Here is just an example of a file to build and test the server.

descrip.mms This is an MMS file to build the executable. If you don't have MMS, you may be able to figure out from loking at it which commands you should use. You can find a machine running MMS and generate the equivalent .com files. See comments at the top of this file on how to run it.

The source files and executable .EXE are currently (October 92) available on HEP decnet in vxcrna::disk$d1:[jfg.www...]. Note also you can pick up the master sources from dxcern:: automatically by running

MMS /MACRO=(U=DXCERN::).

If you are not in HEP decnet, you should find the sources in the WWWDaemon_v.vv.tar.Z file in the distribution. See the README file.

JFG

4.5 VMS Help server

This server can provide WWW users with any information stored in VMS Help format. It currently runs on ONLINE as an experimental service to access standard VMS Help topics. We plan to install this server on VXCERN and other machines to access data stored in the VMS Help system by various CERN Experiment Groups.

Additional information available: :->

Try me !	An example server running at CERN
Features	Improvements made in chronological order.
Bugs	Known problems and improvements being considered.
Technical notes	Technical notes, installation procedure and source files.
Other servers	Other exampels of W3 servers.

4.5.1 VMS Help server Bugs

This is a list of known bugs and desired improvements. Don't let it shrink too fast : send your bug reports and suggestions to Jean-Francois Groff (jfg@cernvax.cern.ch).

- The keyword search works fine on any number of levels down, but then the generic daemon doesn't know how deep the server went, so anchor names lack the intermediate levels. Solution : generate anchor names relative to the input path (before '?').
- DANGER : Attempts to access VMS topics with a weird name like ":=" will crash the server because VMS will try to create a .HLP file with an invalid file specification due to these special characters. Solution : Make a good escaping system (that works with VMS and Un*x styles as well). Crude and bulletproof solution : Ignore any offending topic name !
- Reference to another help library through @ will only search SYS$HELP for the corresponding .HLB file.
- We need an overview page that lists all help libraries available.

JFG

4.5.2 VMS Help server Features

This lists the main features of the VMS Help gateway, with improvements in reverse chronological order. Help make it grow fast : send your bug reports and suggestions to Jean-Francois Groff (jfg@cernvax.cern.ch).

Experimental gateway 0.4 – 2 Oct 91

- Accepts user queries by number or by name. In the latter case, can go down several levels, for instance, from the main help page : "cc /lib" will go to topic CC, subtopic /LIBRARY.
- On invocation with only //node:port/HELP, displays the contents of the standard VMS Help library SYS$HELP:HELPLIB.HLB (function lis_to_html).
- Address format : //node:port/HELP/[@library/][topic[/subtopic]*]

JFG

Chapter 5: Mail Robot

The mail robot is a program which will accept incoming mail and allow remote users to:

- Subscribe to mailing lists (and unsubscribe)
- Retrieve information given a W3 addresss (URL)

Originally from UC Berkeley, an enhanced robot is distributed as part of the world-wide web global information initiative. Futhur information available is:

Help	The help file for users of the robot service
Installation	Installation instructions for unix system managers
Bugs	Lists of improvements requested or needed.
Change history	A list of features introduced and bugs fixed.

5.1 Installation

Here are the steps necessary to install the Mail Robot product on your unix system.

5.1.1 Customisation

Set up the variables in listserv.h and CommonMakefile to suit your site.

POSTMASTER	The address from which messages appear to come. Why not listserv? Perhaps to prevent mail loops.
SECUREWWW	The executable W3 line mode browser (v1.3 or later, so as to have the -listrefs option). This is a separate product. For security, www should be writable only by root.
SERVERDIR	The directory in which you want to put your mailing lists and help about them.

5.1.2 Compile the programs

Everything compiled on AEM's MicroVax II running ULTRIX 3.0 then TBL's NeXT without any problem at all. Your results may vary.

5.1.3 Create your SERVDIR

wherever you specified in listserv.h. Install a HELP file, perhaps using the example-files/HELP in this directory as a template.

5.1.4 Set up an alias "listserv"

Make an alias in your /etc/aliases (or /etc/sendmail/aliases, whatever you have) that points to this program, for example:

```
listserv: "|/usr/local/mail/listserv"
robot: "|/usr/local/mail/listserv"
```

5.1.5 For each mailing list

Create a name.info file giving a bit of information about that mailing list. see the *.info files in the example-files subdirectory.

Create a name file in the same directory, consisting of email addresses one to a line of subscribers to a group. If it is for a brand-new group, create an empty file. Remember that this file must be writable by the mail daemon. The name of the file is just the name of the group.

Depending on how you have your mailing lists set up, you may need to add an alias to the /etc/aliases file for each of the mailing lists. For example:

```
real-recipes: :include:/usr/local/mail/maillists/recipes
```

So sending mail to real-recipes actually goes to each of the subscribers listed in /usr/local/mail/maillists/recipes

5.1.6 Install listserv

Install in the appropriate directory. Edit the CommonMakefile and then

```
make install
```

5.1.7 Run newaliases

This gets sendmail to read the changes in /etc/aliases.

```
newaliases
```

5.1.8 Try it out

Send mail to listserv with body

```
HELP
```

```
<HEADER>

<NEXTID N="1">
</HEADER>
<BODY>
<H2>Using the W3 mailing robot</H2><A
NAME=0 HREF="Overview.html">This robot</A> maintains the W3 mailing
lists, and allows W3 documents to
be retrieved on request.<P>
You can subscribe or unsubscribe
to any of the various WWW mailing
lists by sending email to the robot
"listserv@info.cern.ch" -- see the
commands listed below.<P>
If you have any problems, requests
or questions for a human being, mail
"www-request@info.cern.ch". Lists
are:
<DL>
```

```
<DT>www-announce
<DD> Anyone interested in
WWW, who would like information about
new releases or new online data available.
Please refrain from posting administrivia
to this large list !
<DT>www-talk
<DD> Developers of WWW code,
or those interested in discussions
of technical details
</DL>
You can also find information on
WWW (as well as many other things!)
by telnetting to info.cern.ch (no
username, no password).<P>
If you want to pick up the WWW software,
then use anonymous FTP to info.cern.ch
and look in directory /pub/www. Subdirectories
are src for the latest source packages,
bin for executables for various machines,
doc for "paper copies" of articles
on WWW in PostScript and ASCII forms,
and also the "WWW book" which is
a compendium of the hypertext documentation
of the project. To read the latest
documentation, use WWW !
<H3>Commands</H3>The commands understood by the listserv
program are:
<DL>
<DT>HELP
<DD> lists this file.  This is also
sent whenever a message to listserv
is received from which no valid command
could be parsed.
<DT>HELP groupname
<DD> lists a brief description
of the group requested.
<DT>ADD listname
<DD> Add yourself to the
list
<DT>DELETE listname
<DD> take yourself off
the list
<DT>ADD address listname
<DD> Add yourself
with a given mail address to the
given list. The address must not
contain spaces!
<DT>DELETE address listname
<DD> Remove the
given name from the given list. For
all ADD/DELETE commands, mail is
```

```
sent to the address given to confirm
the add or delete operation.
<DT>SEND document-address
<DD> returns a document
with the requested W3 address.
<DT>STOP
<DD> Stop processing requests: ignore
the rest of the message. Needed if
you send a signature on the end of
your message (or if some gateway
adds one). If in doubt, use it.
</DL>
A command must be the first word
on each line in the message.  Lines
which do not start with a command
word are ignored.  If no commands
were found in the entire message,
this help file will be returned to
you. A single message may contain
multiple commands; a separate response
will be sent for each.
<H4>Examples</H4>
<PRE>
add www-announce

add me@host.uni.edu www-announce

delete me@host.uni.edu www-talk

send http://info.cern.ch/hypertext/DataSources/bySubject/Overview.html

</PRE>
<H3>Subscription</H3>If you are not sending mail from
your preferred mail address, then
you can use the second form of the
command to give your mail address.
If you are not on the internet, please
convert your address into arpa stye.
(For example, UK users please use
international ordering joe@host.ac.uk)
Just speficy the mailbox, without
any spaces.<P>
If you omit the 'address' the command
will assume the mailbox that is in
the From: line of the message.  Note
that SUBSCRIBE is a synonym for ADD;
UNSUBSCRIBE for DELETE.<P>
Please note that is IS possible to
add or delete someone else's subscription
to a mailing list.  This facility
is provided so that subscribers may
alter their own subscriptions from
```

```
a new or different computer account.
There is therefore some potential
for abuse; we have chosen to limit
this by mailing a confirmation notification
of any addition or deletion to the
address added or deleted including
a copy of the message which requested
the operation.  At least you can
find out who's doing it to you.<P>
Note that although you would mail
submissions to a mailing list by
addressing mail to e.g., www-talk@info.cern.ch,
in a subscription request you specify
the name of the list simply (without
the @hostname part) as in the first
example above.
<H3>Retrieving documents</H3>The SEND command (or the WWW command
which is equivalent) returns the
document with the given W3 address,
subject to certain restrictions.
Hypertext documents are formatted
to 72 character width, with links
numbered. A separate list at the
end gives the document-addresses
of the related documents.<P>
If the document is hypertext, it
links will be marked by numbers in
brackets, and a list of document
addresses by number will be appended
to the message. In this way, you
can navigate through the web, albeit
only at mail speed.<P>
If you don't know where to start,
try asking for one of
<PRE>
 http://info.cern.ch./hypertext/DataSources/bySubject/Overview.html
 http://info.cern.ch./hypertext/DataSources/bySubject/Physics/HEP.html
 http://info.cern.ch./hypertext/WWW/TheProject.html

</PRE>for lists of futher pointers.
<H3>CAUTIONARY NOTE</H3>As the robot gives potential mail
access to a *vast* amount of information,
we must emphasise that the service
should not be abused. Examples of
appropriate use would be:
<UL>
<LI>Accessing any information about W3
itself;
<LI>Accessing any CERN and/or physics-related
or network development related information;
</UL>Examples of INappropriate use would
be:
```

```
<UL>
<LI>Attempting to retrieve binaries or
.tar files or anything more than
directory listsings or short ASCCII
files from FTP archive sites;
<LI>Reading internet newsgroups which
your site doesn't take;
<LI>Repeated automatic use;
</UL>There is currently a 1000 line limit
on any returned file. We don't want
to overload other people's mail relays
or our server. We reserve the right
to withdraw the service at any time.
We are currently monitoring all use
of the server, so your reading will
not initially enjoy privacy. End
of cautionary note.<P>
Enjoy!
<ADDRESS>The W3 team at CERN  (www-bug@info.cern.ch)</A>
</ADDRESS></BODY>

<NEXTID 70>
<DL>
<DT>
</DL>

<H2>Mail Robot</H2>This is a "listserv" type program which maintains mailing lists, and
allows W3 documents to be retrieved by electronic mail.
<DL>
<DT>Author:
<DD> Various, modified by TBL.
<DT>Status:
<DD> Source available  by anonymous FTP. (Oct 92)
<DT>Current version:
<DD>1.0
<DT>Platforms:
<DD> Unix only.
<DT>More information:
<DD><A NAME=58 HREF=Overview.html> Overview</A>, <A NAME=69 HREF=Bugs.html>Bugs</A>, <A NAME=68 HREF=Features.html>change history</A>.</A>
</DL>

<NEXTID 2>
<H2>Bugs</H2>This is a list of bugs in or improvements desired in the<A NAME=0 HREF=Overview.html> Mail Robot.</A>
See also the list of <A NAME=1 HREF=Features.html>bug fixes</A> .
<UL>
<LI>The INDEX command ought to be implemented, but for some reason always
returns an empty list.  Occasionally it seems to work.

<NEXTID 1>
<H2>Change History</H2>Changes to the <A NAME=0 HREF=Overview.html>Mail Robot</A>, in reverse chronological order:
<H3>October 1992</H3>TBL added information retrieval possibility using WWW. Release as
an unsupported W3 product to those who ask for it.
```

```
<H3>1991</H3>TBL rewrote str.c (used to overwrite its arguments).
<H3>AEM</H3>A. E. Mossberg, aem@mthvax.cs.miami.edu made a couple minor changes,
to make it slightly less UCSD-specific. He also added a README, and
example files in the subdirectory example-files.
<H3>Origin</H3>Note this is NOT the bitnet LISTSERV program. The term "mail robot"
is yused to attempt to prevent confusion between these two products,
which have different functionality although they do basically the
same sort of thing.<P>
This was the UCSD listserv program, which AEM retrieved from ucsd.edu
by anonymous ftp, TBL retrieved from ftp.eff.org  As retrieved, from
file://ftp.eff.org/pub/listserv2.shar, it consisted of the following
files:
\begin{verbatim}    README
    Makefile
    commands.c
    listserv.h
    main.c
    str.c
    subscribe.c



<HEADER>

<NEXTID N="10">
</HEADER>
<BODY>
<H1>W3 Tools</H1>These tools aid managements of W3
servers, generation of hypertext,
etc.
<DL>
<DT><A
NAME=1 HREF="../Daemon/Basic.html">W3 basic daemon</A>
<DD> Part of the W3 project
code.
<DT><A
NAME=5 HREF="../Daemon/Gateways.html">Gateway servers</A>
<DD>which you can take
and adapt.
<DT><A
NAME=8 HREF="../Frame/fminit2.0/www\_and\_frame.html">Framemaker interface</A>
<DD> There are some
tar files on the anonymous FTP archive
on file://info.cern.ch/www/src which
allow FRAMEmaker to be used as a
W3 tool. Dan Conolly, Convex. Incldues
MIF HTML translation.
<DT>Making HTML into TeX
<DD> We did this
with the "WWW Book" to print it.
See the <A
NAME=2 HREF="TeX/Makefile">Makefile</A> for example, and
```

```
the scripts <A
NAME=3 HREF="TeX/html2latex.sed">html2latex.sed</A> and <A
NAME=4 HREF="TeX/sub1.sed">sub1.sed</A>
. We wrote a special introduction,
but otherwise all the text was hypertext
from the W3 project.
<DT><A
NAME=6 HREF="HTMLGeneration/Overview.html">Generating HTML</A>
<DD> These are scripts
for generating SGML hypertext from
things like directory listings, etc.
Also, for checking and correcting
dubious HTML.
<DT>Server log analysis
<DD> Analysing server
logs requires first of all changing
the numeric internet node numebrs
into domain names. <A
NAME=7 HREF="LogAnalysis/httpd-analyse.c">httpd-analyse.c</A>
is a program to do that. Feed the
results through awk and grep of your
choice!
<DT><A
NAME=z9 HREF="Python/Overview.html">Web-roaming  robot etc</A>
<DD>Guido van Rossum's
knobot code in "Python" language.
<DT>Telnet server
<DD>Setting up a service
machine for anonymous users to log
in to a www client.
</DL>

<ADDRESS><A
NAME=0 HREF="http://info.cern.ch./hypertext/TBL\_Disclaimer.html">Tim BL</A></BODY>
<HEADER>

<NEXTID N="9">
</HEADER>
<BODY>
<H2>HTMLGeneration</H2>
<PRE>
</PRE>Here are some example files you can
use for generating HTML from lists
of files and other things.
<DL>
<DT><A
NAME=5 HREF="fix-html.pl">fix-html.pl</A>
<DD>written by Dan Connolly,
is a perl script to legitimize existing
HTML files into SGML-abiding HTML
(as per the new <A
NAME=6 HREF="../../MarkUp/Connolly/Current/html.dtd">DTD</A> that Dan created).
```

```
<DT><A
NAME=z8 HREF="txt2html.sed">text2html.sed</A>
<DD>A sed script to turn
plain text into plain-looking valid
HTML markup so that it will be rendered
just as it was.
<DT><A
NAME=4 HREF="ls2html.awk.txt">ls2html.awk</A>
<DD>is an awk script which
will just take a list of names and
generate a menu.
<DT><A
NAME=3 HREF="dir2html.txt">dir2html</A>
<DD>is a shell script which
generates a menu of pointers to files
with particular suffixes in a set
of directories. It also includes
a README file at the head of the
hypertext list if one exists.
<DT>htn2html.c
<DD> See the <A
NAME=z7 HREF="../../HytelnetGate/Overview.html">Hytelnet gateway</A>
for the program to convert hytelnet
data into HTML.
</DL>
You can make any variations on these
you like of course.</A></BODY>

<NEXTID 4>
<H2>Updating the Newsgroup lists</H2>To update some of the news pages automatically you must be logged
on to the news server or have the news directories mounted.<P>
 Carl mentioned that you must be a member of the UNIX group news (otherwise
you won't have permission to read the news directories) but that doesn't
seem to be necessary for these functions.
<H3><A NAME=0 HREF=UpgateGroups>UpdateGroups</A></H3>This script updates the list of newsgroups. For the <A NAME=1 HREF=../../../DataSources/News/Groups/Overview.html>overview list</A>
, it saves everything before the "Others" heading, and adds on a list
of pointers to newsgroup stems not already mentioned in the saved
hypertext.<P>
For each stem, it saves any command before the glossary list of groups,
and then regenerates that list of groups.
<H3><A NAME=2 HREF=NewsPage\_Update>NewsPage\_Update</A> (old)</H3>The script NewsPage\_Update creates complete lists of active groups
for the following groups: alt, bionet, bit, biz, cern, ch, comp, eunet,
gnu, news, rec, sci, soc, talk, vmsnet. It does this by writing the
header in explicitly for each group, and then generating a list of
of subgroups using FindGroups<P>
For comp and news, a full list is placed in fullcomp.html and fullnews.html.
 The files comp.html and news.html are formatted by hand already,
and so are not touched by the script.<P>
NewsPage\_Update works by writing some HTML text into a file for each
group to be updated, called [newsgroup\_name].html.new, then calling
the script FindNewsGroups.  This checks the file /usr/local/lib/news/newsgroups
for the groups within the current group which are active.  Finally
```

the new file is renamed to remove the .new.<P>
The list of stems to search, and their titles and any other comment
is hardcoded into the NewsPage_Update script, and the list is DUPLICATED
in Others_Update.
<H3><A NAME=3 HREF=Others_Update>Others_Update</A></H3>The Others_Update script finds stems which are not included in the
Overview.html file, but which are active. This list of which groups
not to include is hardcoded into the script. For each group, it calls
GrpCreate. This adds the name to OtherGroups/Overview. It then
runs FindNewsGroups for each group.
<H4>NOTE</H4>Once the script has completed all the .new groups must be renamed
manually to remove the .new extension.
<H3>GrpCreate</H3>This reads a newsgroup stem name from stdin.<P>
It then creates the top of a file for the list of groups with that
stem. This will be called \$\{nn\}.html.new. where \$\{nn\} is the stem
name. Unfortunately there is no way to get a description of the stem
to include in this file. However, if the .html fie already exists,
it will use everything up to an excluding the first DL tag from the
.html file for the .html.new file. Therefore, everything above the
DL tag may be hand edited.<P>
GrpCreate adds a pointer from OtherGroups/Overview.html.new to the
.html file.<P>
The .html file is renamed .html.old, and teh .html.new becomes .html,
.\{\char'134\}" Macros for HTML

.\{\char'134\}" Jim Davis 6 Nov 92

.ps 12

.in 5

.de B

..

.de R

..

.de H1

.ti -5

.ps 18

\{\char'134\}fB\{\char'134\}\{\char'134\}\$1\{\char'134\}fR

.ps 12

.br

..

```
.de H2
.ti -3
.ps 14
\{\char'134\}fB\{\char'134\}\{\char'134\}\$1\{\char'134\}fR
.ps 12
.br
..
.de H3
\{\char'134\}\{\char'134\}\$1
.br
..
.de H4
\{\char'134\}\{\char'134\}\$1
..
.de H5
\{\char'134\}\{\char'134\}\$1
..
.de H6
\{\char'134\}\{\char'134\}\$1
..
.de H7
\{\char'134\}\{\char'134\}\$1
..
.de H8
\{\char'134\}\{\char'134\}\$1
..
```

```
.de H9

\{\char'134\}\{\char'134\}\$1

..

.de DL

.in +5

..

.de DE

.in -5

..

.de DT

.ti -3

* \{\char'134\}\{\char'134\}\$1

..

.de DD

.br

..
```

```
\begin{verbatim}
Date: Wed, 4 Nov 1992 16:48:34 -0500
From: Jim Davis <davis@dri.cornell.edu>
To: wei@xcf.berkeley.edu, www-talk@nxoc01.cern.ch
Subject: improved printing of WWW files
```

If you can't quite manage to live without hardcopy, you may wish sometimes to print WWW files. I have written a couple of scripts to do this. They are particularly useful with Pei Wei's excellent Viola WWW browser.

A tar archive is available for anonymous FTP:

dri.cornell.edu/pub/davis/print-www.tar

It contains:

```
README
print-www
print-www.l
html-to-latex
html2latex.sed (modified version of original CERN version)
```

The hardest part was writing the perl script to obtain documents via http protocol - turns out you cant just run pipes through telnet.

The conversion from HTML to LaTex is not really robust yet - this is doubly hard since there is no guarentee that the HTML is legal. But at least it works for my test cases. No doubt it will be improved in time.

best wishes

5.2 Making a server

Here is a run-through of what is needed to make a www server , with examples from a suggested server for the HEPDATA base of Mike Whalley . See also etiquette .

Basically, to make the data available, you make a server which is a modified version of your program. When a user follows a link to HEPDATA (or runs a command to jump straight there), the client program opens a connection to a server program on a VM machine (say, but could be VMS or unix). The server in turn runs your program.

Let me just describe the essence of the changes needed so that you can get an idea of how much effort would be involved.

The first thing you do is to make up an arbitrary naming method for anything which HEPDATA can display. In this I include the welcome page, any menu, any article, any help text. Typically one invents a hierarchical naming scheme, like

```
/HEPDATA The first "welcome" menu
/HEPDATA/HELP The top-level help

/HEPDATA/HELP/REAC The help on the reaction database.

/HEPDATA/REAC The reaction database itself

/HEPDATA/REAC?P+PBAR list of reactions involving p and pbar (?)

/HEPDATA/DATA/RD125V687 Some article (say).
```

You do this because, whereas an interactive user follows a path through the program, the W3 user calls the program once for each thing. There is no "state" information. This allows one to make a hypertext link to any part of the scheme and jump back in again later. For example, one might want to quote an article, or the reaction database, or a particular list of reactions.

Now all you do is modify the program so that, given a name above, it will

return the required document. This means basically turning it from a sequence the user goes through into a set of conditionals to isolate each of the individual cases above. Apart from that, the data retrieval code is unchanged apart from the output formatting. Many of the options in fact mean mapping the name onto a fixed

file's name its the searches which have to activate real code.

The hypertext trick you need to use in the menus. Where an option is normally output to the screen, you have to tell the client what to ask for is the user selects that option. For example, in the main menu /HEPDATA you have an option which gives the help. You would represnt this "anchor" as

```
<A NAME=4 HREF=/HEPDATA/HELP> Help </A>
```

"Help" is all that is displayed, with some indication that it is an option. If the user choses (clicks a mouse on, choses by number depending on which client he has) then the client asks the server for /HEPDATA/HELP. ("A" is for "anchor", "HREF" is for "hypertext reference")

For the index searches, it's as simple. When the server sends the text called /HEPDATA/REAC it also sends a special tag . This tells the client to enable a FIND command, or find panel etc (depending on the client). You don't have to do any human interface work. The client automatically comes back with a search coded up in the form /HEPDATA/REAC?P+PBAR etc. Your server in turn returns a menu (say) with pointers to the data which has been found.

You can also put some formatting tags (like headings) which will make the data look really nice on a window system.

Tim BL

5.3 A Shell Server for HTTP

The HTTP protocol is very simple. The following is an example of a server program written in sh: beginverbatim #! /bin/sh read get docid echo "<TITLE>$docid</TITLE>" echo Here is the data

endverbatim The docid may have a trailing carriage return to be stripped off on some systems. You can modify that script to produce the data you actually want. The HTML syntax for marked-up text is fairly simple, but if you want just to send plain text, then just send the .PLAINTEXT.tag first: beginverbatim #! /bin/sh read get docid sed -f txt2html.sed $docid

endverbatim or in csh beginverbatim #! /bin/csh request = ('echo $<') if ($#request <2) exit endverbatim sed -f txt2html.sed $request[2] beginverbatim

endverbatim When you have written your script, set the execute bit and then configure the inet daemon to run it . A few more examples:

- A sh script to generate a menu for files in a directory
- An awk script to generate menu from a list of files .
- A perl script for all kinds of stuff on the ASIS server
- The shell script of the Hytelnet gateway

If you know the perl language, then that is a powerful (if otherwise incomprehensible) language with which to hack together a server.

See also a case study of mapping a database onto the web .

All contributions to these examples welcome!

Tim BL