

httpd4Mac v13b

Thank you for taking an interest in 'httpd4Mac' (for Macintosh) now at version BETA 1.3 This program used to be called just 'httpd', but that caused some confusion, so I have renamed it 'httpd4Mac'. httpd4Mac is a simple program that allows you to use your Mac as a WWW server.

This home.html file is an HTML version of the file called 'About httpd4Mac...' that used to come with this program.

I used to operate an experimental home page at <http://130.246.18.52/> but due to a change in circumstances I cannot maintain this anymore, however Peter Hardman of Manchester University has kindly offered to host the pages at <http://sodium.ch.man.ac.uk/pages/httpd4Mac/home.html> and I very grateful to him, this site may occasionally contain latest development version, but cannot offer technical support.

v13b is basically a release of the development version v123d9 which has been running on the server above successfully for some time now, plus a couple of minor bug fixes and one improvement.

The following areas are covered in this file:-

Index

- [Warranty](#)
- [Overview](#)
- [Why httpd4Mac ?](#)
- [Machine Requirements](#)
- [Running the program](#)
- [Memory setup](#)
- [Prefs file directives](#)
- [About MIME](#)
- [Multiple servers](#)
- [Files that can be served](#)
- [Writing WWW pages](#)
- [Likely improvements for future releases](#)
- [Possible improvements for a future release](#)
- [Unlikely improvements for a future release](#)
- [Unexpected 'features'](#)
- [A bit about the code development](#)
- [List of bug fixes/release history](#)
- [Incompatibilities](#)
- [New Versions](#)
- [Comments/Problems](#)

WARRANTY

This program is supplied free with absolutely no warranty and may do unthinkable damage to your machine or data, for which I will accept no liability. Having said that, I didn't write it to do that, but you must satisfy yourself it will not do any damage before using it.

You can do whatever you want with it except:-

- Distribute it without this file, home.html, which describes it.
- Charge money for it (fair duplication/floppy costs excluded)

[Back to index](#)

ABOUT httpd4Mac

'httpd4Mac' is an fba (Faceless Background Application) that implements a minimal http server.

It currently supports the GET and HEAD methods, with full or simple requests. It ignores 'Accept' fields from the client. NB The http spec does not allow such a thing as a simple HEAD request. The only simple method is GET. But seeing as the CERN server accepts it (and the guy who invented WWW wrote that program) I thought I should also.

WHAT DOES THAT MEAN ?

That means it should work as a fairly straightforward WWW server. You should be able to write pages with images, and have separate image, sound and movie files. What it does not (yet) support are clickable images or forms and other interactive pages. I may write the code to support this later but it is not planned at the moment. I have tried using MacWeb and X-Mosaic and X-Netscape to retrieve pages and they appear to work OK.

A Faceless-Background-App (FBA or Background-Only-App (BOA) as the name suggests runs in the background. When you launch it you will see it go grey, but it will not appear to have run. No menu will appear, nor will any windows. It will not appear in the application menu either. This is perfectly normal, but unless you set the prefs file to notify you about major events, you will not know if it is running or not.

At some stage I may code a foreground version which will have a menu and text window. This will not be available for some time though.

[Back to index](#)

WHY HTTPD4MAC ?

There is one other http server I know of for the Mac which is not free. The aim of 'httpd4Mac' is to provide a free http server, which implements enough to serve out pages on your mac, without it putting an enormous load on the machine. It is designed to be lightweight and fast.

It may not be much good if you expect to have loads of pages and images, or if you want to do more than just serve out static pages of HTML. However I do know of a few people that have written quite an extensive set of pages and of users that have implemented servers from old Mac Pluses and other lowly machines.

It is not a port of either CERN's or NCSA's httpd for UNIX. It does not use the common code library. The only reason I gave it a similar name was because I couldn't be bothered to think of a different one.

[Back to index](#)

MACHINE

You will need a Mac running at least Sys 7, with MacTCP version 2.xx an internet connection, some memory and preferably a hard disk. I think that should do it. It will run on 68000 machines.

[Back to index](#)

RUNNING IT

Just launch it. You may need to read the section about [memory allocation](#) as well. It will look for its config/prefs file firstly in its own folder, then in the system/prefs folder. If it does not find it, it creates it own from default info and quits. This default prefs file can be edited to set up httpd4Mac how you want. The app will still look 'greyed out' as if it is still running. This is normal. Just double click it again when you wish to re-launch.

You will find this self-created prefs file in the preferences folder (within System folder) You can place this prefs files in the same folder as httpd4Mac or leave it in the prefs folder. I prefer keeping it with the app. Re-launch when you have edited the prefs file.

If httpd4Mac finds a prefs file, it checks the version as it is reading it in. If the notify option is on within the prefs file and the version is out of date, httpd4Mac will notify you. If you have an out of date prefs file, this means you may not be using the latest features of httpd4Mac.

You can let httpd4Mac create an up to date version for you by hiding your current prefs file (eg move out of httpd4Mac's folder or system folder). Re-launch the app and you will get the latest version.

Use a text editor to edit the prefs file. There are comments within it to help you. It would probably help if you knew a bit about servers, http etc, but the sample prefs will work so if you know nothing except how to write WWW pages then you could just re-launch. Apologies for the fact that some of the configuration options don't work yet. Occasionally I refer to the prefs file as a config file, but I'll try to avoid that.

[Back to index](#)

MEMORY PARTITION

The partition is set at 500K. In fact httpd4Mac may run in a partition of ~200K + size of largest file expected to be served out by httpd4Mac. ie If you have a 100K image file that can be accessed, then the partition should be at least 300K. It may run with considerably less though. The memory requirements can depend on many variables.

If you do not set enough memory you may get problems.

The best way to find out the ideal partition size is to start httpd4Mac with the debug option on with verbose messages, then force it to quit immediately. If you do not have a process management utility, then just restart your Mac. When you examine the debug log file you will see some lines showing a memory analysis. As long as there is enough free memory to hold the largest file you expect to serve, plus a bit spare (say 10%/5K which ever is larger, this will probably do. However you will get better results the more memory you allocate.

To allocate more memory use the Get Info function from the Finder File menu.

[Back to index](#)

MORE ABOUT THE PREFS FILE DIRECTIVES

dnr

This option tells httpd4Mac to do a reverse DNS look up on each connection it receives when logging. If you are not logging this makes no difference. If this option is off the server may run a bit

faster. If an error occurs when looking up the machine name, then only the IP number will be recorded. Log info is actually queued while the client request is being serviced, the actual writing to a log file being deferred until later. Should httpd4Mac be forced to shut down before all logging is complete, it is rushed through without looking up names, so you may find just a list of numbers occasionally at the end of the log file.

mime_def

Allows you to associate suffices and/or Mac file Types with a MIME Type, read about MIME below.

create_access_log

Attempts to create a new log file each time httpd4Mac is launched. If it fails logging is disabled. If an old file exists it is renamed with a .bak extension. (If a .bak file exists it is simply deleted) If this renaming process fails then again logging is disabled. If this option is off, but a file is found with the correct name (eg httpd_access_log) then log info is simply appended to this file.

cache_check and cache_life

httpd4Mac caches every file it reads in into memory so that if the file is re-requested it is delivered that much faster without having to access the disk and slow down other apps. When the cache is full, it makes space for new items by looking to see how many times items have been downloaded and the last time items were downloaded and throws 'unpopular' items out on this basis. In addition httpd4Mac checks the cache regularly to see how old items are. Once they are over a certain age they are thrown out. This is set by the 'cache_life' option. Although this is good for a server that has a few files that are rarely altered, but often downloaded, it is not so good for development of your pages and in addition means that when you do occasionally update your pages, the old version is still served until the cached item 'times out'. A further refinement (cache_check 'on') allows you to tell httpd4Mac when serving from the cache to check the cache dates against the file dates, if they do not match (indicating a possible file change has occurred) then the server trashes the cached item and re-reads the file before serving. However this will slow the speed of the server slightly, the choice is yours.

[Back to index](#)

ABOUT MIME

MIME stands for Multipurpose Internet Mail Extensions. While most operating Systems now revolve around the idea of a file as a basic unit of storage on a disk, none has really defined a standard way of identifying a particular file type or format. Various OS have their own methods though. Apple incorporates a file type into the file header, meaning you can have any file name and change it, but the contents are still regarded by the OS as having not changed. UNIX users rely on file suffices (eg .html) and sad DOS users rely on 3-letter suffices (eg .htm) Mac users can also use suffices if they want.

The idea of MIME (I think) was to fill that gap and provide a OS-Independent method of identifying information types, like GIF image files for instance. RFCs 1521 - 1524 describe MIME, it was really meant not to be incorporated into an OS as such, but used when transferring files (attachments) from machine to machine by E-Mail.

As the World Wide Web uses many different file types and should be almost platform independent I guess it made sense for the designers to choose MIME to define what was being sent. In fact the http spec allows for a MIME dialogue to take place between client and server, with the client saying 'I know about xxx format' and the server saying 'The file you are getting is xxx format'

The idea of the mime_def directive is for you to tell httpd4Mac what mac file suffices and/or what mac filetypes you are associating with what type of MIME format. Some examples are given in the sample prefs file.

When httpd4Mac receives a request it works its way through the list of mime_def directives, until it finds a match. Unless you know exactly what type of Mac file that, say your scanner software produces, you

may want to just stick with suffices. Mac text files always have a file type of 'TEXT' though. httpd4Mac always slips in a default def of AnyFileType & AnySuffix = text/plain, for those types that do not match at the end of the list.

eg You can assume that the list always ends (internally) with

```
mime_def      '*' & * = text/plain 7
```

You can put a wildcard character (*) in the mime_def line to match against anything. It is perfectly acceptable to have all of your MIME defs lines with a wildcard for the Mac filetype, and only use suffices to distinguish between different file formats.

httpd4Mac converts all URLs it receives into lower case and then attempts to match to a MIME def. Make sure your mime_def suffices are in lower case !

NB There are some MIME types which have 'application' in them. You cannot use most of these with httpd4Mac to deliver a Mac application to a client. This is because a Mac application uses both the resource and data fork of the file, but httpd4Mac serves only the data fork. What you have to do is convert the app to a bin-hex text file and define a MIME type that maps .hqx to application/mac-binhex40. In fact any file that you to serve and have both the resource and data forks recovered at the browser end will have to be converted to stuffit archives or bin-hex. Fortunately most document type files use only the data fork, so you should not have a problem. See the section on [Files that can be served](#)

[Back to index](#)

MULTIPLE INSTANCES OF HTTPD4MAC

Just copy http4Mac for each 'server' you wish to run. Place the copies in separate folders and create separate prefs files, also placing these in the folders. Set them up to run on different TCP ports and then launch them all. Make sure you set them up to serve on different TCP ports. Although it will not detect multiple servers on the same port and will appear to run fine, you may get some unexpected results.

[Back to index](#)

WWW FILES

httpd4Mac can potentially serve any file in the same folder as the app or in sub-directories (sub-folders). This is the same if you launch it by means of an alias (Say by putting an alias in the startup items folder), the files served will be only those in the same folder as the app. If the URL '/' is requested (ie 'http://host.name/') httpd4Mac looks for the file 'home.html'.

'/' delimiters in the URL indicate sub-dirs in the usual manner (even though the specs say the use of the same delimiter as is used for UNIX directories is co-incidence)

You can happily use the '.' and '..' format for relative URLs, well, with the clients I tried anyway. You cannot (yet) put aliases in the app folder to serve out files from elsewhere. All httpd4Mac understands at the moment is normal folders and files. It does not distinguish between text, sound, or image files, as regards data, but will send different MIME headers according to the MIME definitions. In other words it just sends the file requested without any preprocessing.

Make sure of course that sound/image files you serve out match what the client expects, the standard format seems to be Sun ULaw sounds format and GIF image format, although Netscape users can accept inline JPEGs.

You can use httpd4Mac to serve out Mac-specific files for Mac users only (eg applications). The trick is to convert them to their bin-hex equivalent first (using Bin-Hex4.0) or to a stuffit archive. This will give you a '.hqx' or '.sit' file. If you try to serve a self-exploding '.sea' file, this will not work because the exploding code is in the resource fork, which httpd4Mac does not send. (The archive data will get there, but the archive won't be self exploding because basically an .sea file is an application) Using the mime_def directive create a mime type 'application/mac-binhex40' that will map to all '.hqx' files and 'application/x-stuffit' to map to '.sit'.

On the client side you have to set up the browser so that it understands this new MIME type and suffix and that it launches a program called 'Stuffit Expander' when it receives such a file. This would probably be done under helper applications setup in your browser. Stuffit Expander can be obtained from your fav ftp site. Stuffit expander is (I think) a free program that understand bin-hex and sit formats.

[Back to index](#)

HOW DO I WRITE WWW PAGES

If you don't know I'm not going to explain it here. There is a web site offering [Help on HTML](#) which I used to learn and this also has references to other sites. You can also try to find something from the [W3](#) consortium.

WWW pages are actually just text file written in a language called HTML, and include pointers to the image and sounds files etc. These files can be local to the machine or served out from a remote machine. Some client programs like MacWeb, come with local files written in HTML explaining how to set it up. These are often set up as the home page as default. You'll get an idea about HTML simply by looking at those files with a text editor.

[Back to index](#)

LIKELY IMPROVEMENTS FOR A LATER RELEASE

- Error logging
- Prevent serving out of logs and prefs file etc.
- Clickable image support
- Support for Finder aliases (+ maybe remote volumes)
- Better caching

[Back to index](#)

POSSIBLE IMPROVEMENTS FOR A LATER RELEASE

- Speed improvements
- Understand client accept fields and act on them, maybe
- Do prefs file syntax check
- A version that runs in foreground and prints log output to a window.
- A configuration program with proper User Interface to allow easy configuration. May allow remote changing of configuration by Program Linking.
- An option to give directory listings in HTML

[Back to index](#)

WHAT IT MAY NEVER DO.... (Depends if I can be bothered)

- Support Applescript or anything that might be needed for interactive pages ie Search Functions. Clickable images are coming though !
- Support additional methods.(eg POST for fill out forms)

This is mainly because I do not understand the CGI interface or how things like Applescript/MacPERL would drive/be driven by httpd4Mac, but also because of time. If someone can point me to a good source for explaining this, then it may be a possibility. I know there is a mailing list for MacPERL, but I have not had time to enquire further yet.

[Back to index](#)

UNEXPECTED FEATURES

1. At present there is nothing to stop you requesting the debug info or log files (or even httpd4Mac itself !). Such action might have some very weird if not dangerous results, I would not recommend it.
2. httpd4Mac takes a little time to get going. Usually access is not possible until about 10 seconds or so after start-up. The reason is because httpd4Mac relinquishes control ASAP and does its initialisation slowly in the background. The reason being if the app is in your startup items folder, the boot process does not appear to be slowed down by it.
3. httpd4Mac uses a crude caching mechanism to store files that have previously been requested. Basically an item is cached until the copy is a certain age, no check is made to see if the file has changed. This is fine if you do not change the pages pages often, but if you do you may wish to set this lifetime to a lower value. A better caching system will be in future versions. The cache life is controlled in the prefs file.

[Back to index](#)

ABOUT THE CODE DEVELOPMENT

This is my first TCP programme. It was developed on Think C v7. I made use of some utilities, which you also find useful if you have any problems.

MacsBug

Apple's low level debugger

ZapTCP

Apple extension, helps when TCP apps crash unexpectedly

MacTCPwatcher

Utility from Peter Lewis for testing MacTCP functions

ZoneRanger

Utility from Joshua Golub, used for examing memory

ProcessFinder

Utility from Edward Harp, used to kill the background app without rebooting

It was developed on a Mac IIci, running Sys 7.1, MacTCP 2.0.4, with 20Meg of RAM. You won't need that much though. It will not run on a 68000 machine. It does not use any floating point code. It has also been run for many weeks now on a Centris 610, again Sys 7.1, MacTCP 2.0.4

I have made some changes to the code which should allow a native PPC version. Actually I've already made it, but it just crashes all the time, so I haven't bothered to release it just yet :-). However I have had mail from two or three people indicating it is already being used successfully on Power Macs under emulation.

[Back to index](#)

INCOMPATIBILITIES

So far two incompatibilities have been brought to my attention. The first is a program called 'autodoubler' (thanks to phardman@ssci.liv.ac.uk (Peter Hardman)) Compressed files get sent in compressed form. It appears they do not get uncompressed on the fly as they should.

Another incompatibility comes from using httpd4Mac with Netscape while running NowSuperBoomerang on the server (Part of the NowUtilities package). (Thanks to someone for that, I can't remember his name) I have also had one E-Mail suggesting problems with Sys 7.5.1, but no further information was forthcoming on that....

[Back to index](#)

NEW VERSIONS

I will always place new versions on the info-mac archives at Stanford. I do not plan to post them elsewhere yet, so latest copies can be obtained there.

[Back to index](#)

COMMENTS/PROBLEMS

I will not be in a position to answer any E-Mails or USENet postings about httpd4Mac for a little while, and although you can find the WWW pages, there will not be any technical support. I apologise for this situation and I hope I can resolve it soon. The WWW pages include a page about common problems and it is possible this may get updated. On USENet the place to talk about your problems (as regards using httpd4Mac that is) is comp.infosystems.www.servers.mac. I will try to respond to any posting I see on there, but there are other users of httpd4Mac who may be able to help you.

If you have problems you could try the following:-

- Switch on the notification option in the prefs file, this might tell you why it is failing.
- Also switch verbose debugging
- Read this page more thoroughly.

Some of this information may explain what the problem is.

I have sometimes referred to this program as shareware, sorry if that has caused confusion. It is freeware, you have no obligation to me regardless of what you want to use it for.

Bill Melotti 20 October 1995