NCSA httpd

NCSA httpd is a HTTP/1.0 compatible server for making hypertext and other documents available to Web browsers. Please read this copyright notice.

The current version is 1.4. If you are using an earlier version, these upgrade notes will be helpful. We also have an FAQ on NCSA httpd.

A list of know bugs and enhancement information in NCSA httpd 1.4 version can be found in:
http://union.ncsa.uiuc.edu/HyperNews/get/www/httpd–beta.html

Acknowledgement

- Thanks to Robert McCool, who developed the NCSA httpd till version 1.3 and put together this excellent documentation.

A Note to Users

- Any questions or bug reports regarding NCSA httpd should go to *httpd@ncsa.uiuc.edu.*

Why NCSA httpd?

Here is a list of the server's features. Follow the links to read about them and see demonstrations of their use.

- ## Low impact

- Designed to be a small, fast server.

  Has a low impact on system resources, allowing it to serve documents to a large number of users simultaneously with little slowdown. With the amount of traffic `www.ncsa.uiuc.edu` gets, this was required.

- Runs standalone, for faster performance, or under `inetd`.

- ## Compatibility

  NCSA httpd speaks both the HTTP/1.0 and HTTP/0.9 protocols, and will serve documents to World Wide Web browsers speaking either dialect. In addition, NCSA httpd allows easy serving of multimedia documents through both protocols.

- ## Unified Directory Structure

  NCSA httpd allows you to serve your documents from a single directory, which can be any physical directory. The World Wide Web browsers never see the name of this physical directory, allowing you to move your documents into different directories (or to a larger disk) without rewriting your HTML documents and telling everyone who may be using your server about the move.

  In addition, if heavily publicized documents move from your server to a different one, NCSA httpd allows you to easily point the users to the new location of the documents. With some browsers, the user will not even notice the change.

- ## Directory Indexing

  NCSA httpd allows you to create HTML catalogs of your directories, in order to tell users what's in them.

  For instance, on hoohoo, I have a directory containing some neat pictures, and I've written a small index for it. I have another directory whose contents are constantly changing. I don't really want to write and maintain an index. In this case, httpd will generate an index of the directory for me when a user requests the directory. I can configure httpd to ignore certain files, to give a certain icon for certain files, to place a short description after file names, and include either an HTML or plaintext README which tells people what is in the directory.

- ## User Supported Directories

  NCSA httpd allows you to let users make their own directories available for HTTP access, without your having to create links to them. Of course, you can limit what the users can do from their directories for security reasons.

- ## Server Scripts

NCSA httpd allows you to write server scripts, which are executable programs which the server runs to generate documents on the fly. They are easy to write and can be written in your favorite language, whether it be C, PERL, or even the Bourne shell.

These scripts can be used for a number of purposes.

- HTML forms

  If you are using a browser which supports HTML forms, like Mosaic 2.0 for X, you can create scripts to handle form requests.

  If you are familiar with the PH system, you can look a person up in the phone book.

- Simple HTML indexes

  Most browsers support the HTML <ISINDEX> flag. You can use server scripts to respond to them.

  For example, what is hoohoo's load average right now?

  Let's try a simple finger gateway. Try looking for robm@void, you may find a link.

  How about looking something up in archie?

- Image mapping

  If you are using a browser which supports it, you've probably noticed that image mapping is a fun feature. NCSA httpd lets you create your own maps.

- ## Security

  NCSA httpd allows you to control who can access different document trees on your server. It also allows you to define which server functions are carried out in different document trees.

  Note that the following methods of access limitation can work independently, or be combined.

  - Host Filtering

    Host filtering is used to limit document trees to certain machines, most likely local machines or those directly involved in a project.

    For example, I have some which are NCSA only. If you are not from a machine in ncsa.uiuc.edu, you can't access them.

I have another that I don't want NCSA people to read, since they can access it elsewhere.

- User Authentication

  If your browsers are using a browser that supports user authentication, like Mosaic 2.0 for X, you can have browsers authenticate themselves with a user name and a password before they are able to access protected documents.

  For an example, let's create three users.
    - `robm` password `pinhead`
    - `marca` password `conehead`
    - `ebina` password `bonehead`

  I have protected this `directory` so that only `robm` can access it.

  I have protected this `directory` so that only people in the Mosaic X developers group can access it. That means `ebina` and `marca`.

  I have protected this `directory` so that only `robm` can access it, and only from `void.ncsa.uiuc.edu`.

- ## Server Side Includes

  NCSA httpd allows you to include the output of commands or other files in your HTML documents.

  For instance, the current date is:

  At the end of each document, I'd like to include my signature but don't want to have to change every document I wrote when my home document moves. So, I can use a server include tag to include it, like this:

  Note:
  Having the server parse documents is a double edged sword. It can be costly for heavily loaded servers to perform parsing of files while sending them. Further, it can be considered a security risk to have average users executing commands as the server's User. If you disable the exec option, this danger is mitigated, but the performance issue remains. You should consider these items carefully before activating server–side includes on your server.

NCSA httpd Development Team
httpd@ncsa.uiuc.edu