FREQUENTLY ASKED QUESTIONS

An FAQ list is really a cop-out from managed information. You should be able to find everything you want to know by browsing from the WWW project page, as everything should be arranged in a logical way. Here though are things which maybe didn't fit into the structure, with pointers to the answers which maybe did. It’s an experiment, started May 92. The questioners are anonymous.

How does www keep track of the available servers?

How does www keep track of the available servers? How does a user know where to go to get a specific piece of information? According to the description of the http protocol, when a user wants to do a search, the corresponding UDI specifies, among other things, the server's address. How does the user find out about the server's address? Or from the server's perspective, how does a server announce its existence?

The resource discovery problem

14 May 1992

This is what people seem to call this problem in general.
As a physical sever can serve many different types of information from different servers, we talk about finding documents and indexes, as that is what the user sees. To the reader, the web is a continuum. When a new server appears, it may serve many databases of data from different sources and on different subjects. The new data must be incorporated into the web. This means putting links to data on the new server (especially to a general overview document for the server if there is one) from existing documents which interested readers might be reading, or putting it into an index which people might search.

The person publishing the data must go through the same process as the person searching for it. When (s)he has found an overview page which (s)he feels ought to refer to the new data, (s)he can ask the author of that document (who ought to have signed it with a link to his or her mail address) to put in a link. There may be several links from different documents: there is not one master list. Of course, some servers are put up for internal use only, and links are only made from local documents. I only find out about these servers by word of mouth, but they exist.

Currently, there are three parallel trees in the web for finding data starting from scratch. The most interesting one is a classification by subject. I've got an "Other subjects" link from Cern's home page to a master page of information by subject . From that I have links to individual servers of all kinds (W3, WAIS and Gopher), and in cases where there are a lot like physics and biology, a link to a page about one specific subject. In this way you can browse the web by subject like a library. I am looking for people in other disciplines to take over the subtrees for those disciplines as the load gets heavier (I may have candidates for some). The tree tends to be ought of date, and its authors rely on feedback to put in things which are missing.

The other trees are by organisation and by server type. The list by server type is easy, because the people responsible for each protocol keep a list of the servers using it. That is, there is a tree of gophers, and there is an index of WAIS indexes. There is the W3/WAIS/Archie server for FTP sites. This tree isn't so useful unless you know what sort of a server you are looking for, but it tends to be more up-to-date than the subject index. It also has things in which aren't just about subjects. The third tree was going to be a geographic tree of organisations, but that isn't at all up-to-date.

By the way, it would be easy in principle for a third party to run over these trees and make indexes of what they find. Its just that no one has done it as far as I know because there isn't yet an indexer which runs over the web directly.

As you can see, the web is sufficiently flexible to allow a number of ways of finding information. In the end, I think a typical resource discovery session will involve someone starting on their "home" document, following one to two links to an index, then doing a search, and following several links from what they have found. In some cases, there will be more

than one index search involved, such as at first for an organisation, and having found that, a search within it for a person or document. We need to keep this flexibility, as the available information in different places has such different characteristics.
In the long term, when there is a really large mass of data out there, with deep interconnections, then there is some really exciting work to be done on automatic algorithms to make multi-level searches.

W3 vs WAIS and Gopher

What's the difference between W3 and WAIS? What's the difference between W3 and Gopher? Why invent yet another system? Which one should I use?

The data model

W3 is comparable to both WAIS and Gopher , in that it is a client-server information system running over the internet. There is a difference in the data models. The W3 model is that everything (document, menu, index etc) is represented to the user as a hypertext (hypermedia) object. There are two navigation operations are available to the user: to follow a link or to send a query to a server. Only certain documents are flagged as having a search facility, and not all documents have links, but some documents have both. That's a pretty simple model, and results in a pretty simple user interface.

Two neat things fall out of this model. One is that it turns out that almost all other information systems can be represented in terms of W3 documents. A W3 user can interrogate WAIS indexes (example) and Gopher servers (example). This comes from the flexibility of the W3 model to describe other structures. A WAIS database is a searchable document. The hit-list returned by a WAIS server (or any other query engine) is a hypertext document with links to the documents found.

Gopher menus (or any other hierarchical menu system, including a file system) are represented as lists of items linked to other objects. The W3 system has an open addressing scheme allowing links to be made to any objects on W3, WAIS, Gopher, FTP, NFS, or Network News servers.

This flexibility has allowed lots of different kinds of data to be put on-line by writing a simple script to generate a hypertext "view" of the database. The hypertext model, then, is flexible. It is also powerful as a communications medium. To author a document in hypertext is to communicate better. It allows one to put in a link whenever the reader might need background information.

WAIS without links

You miss the links in WAIS in two ways. One is when you are looking for an index. You can't follow links from an overview page to "browse" through different indexes. You can only use a master index (the directory of sources) to find indexes. The other way is that when you have retrieved something, whether part of the FORTRAN manual or part of a mail discussion, you get it in isolation. You can't follow links from that document to related documents.

Gopher without text

A Gopher menu is a dry list of items. Each line has 80 characters in which to describe an option. In practice, to communicate with the reader, one needs the full power of text formatting in a number of styles. A plain list turns out to be relatively infrequently used when the author or the program generating the document has a choice. Note that the "Panda" project adds some plain text to Gopher menus, but this is only a small step toward the flexible blending of links and text which is hypertext.

Group work

The second big difference is that W3 is designed to include collaborative authoring (CSCW) so that groups can share information, rather than simply individuals disseminate it. We only have a first stab at this on the

NeXT platform, as we were overtaken by the web's success in dissemination mode.

Deployment levels

The W3 software is not (May 92) as deeply deployed as WAIS and Gopher software. This is basically because it takes more time to write a hypertext client than a menu or query client. (Also, because the initial W3 instigators are paid to work for the world of High-Energy Physics primarily).

However, the W3 world is growing very fast. There is widespread recognition that hypertext is essential for the next generation. It is planned to merge the W3 and Gopher systems, and there is no reason (apart from server simplicity and, perhaps, response time) why both of these systems could not use the WAIS protocol when it settles down.\

The Choice

Bear in mind:

- A W3 client can read data from any other system.
- If you run a W3 server you can upgrade certain parts of the documentation to hypertext later.
- Hypertext is neat for representing existing data easily.

So install W3 clients, and W3 servers. If you want to install a Gopher or WAIS server, fine: the W3 clients will access it. If you install a WAIS server, then you could install the W3-WAIS gateway locally to save bandwidth.

How to create a W3 server

What's the difference between a server and a gateway ? Which one do I need ? What's the easiest way to create a server ? How do I customise the distributed software to my needs ? How can I write an index server ?

Basics

Since this FAQ came up, a lot more documentation about starting servers has come onto the web, so it might pay you to browse a bit more. There are so many ways, that it might seem complicated, but in fact most of the methods are very easy.

The server that we distribute as WWWDaemon is in two parts: a common HTDaemon program taking care of the communications, and a variable HTRetrieve function called by HTDaemon with the significant part of the address (i.e. after host) split as argument and keywords (look at the code).

The default HTRetrieve function, located in HTRetrieve.c, only knows about finding files and sending them. To make an index server, you need an HTRetrieve that knows how to query a database, or otherwise find info, from keywords supplied after the '?' in the address. It can be as simple as a 'grep' in a series of files.

We provide 3 example index servers built this way: VMSHelpGate, which gives access to VMS help, FindGate which gives access to a mainframe search engine called XFind, and WAISGate, which speaks the WAIS protocol to contact their search engines. You can read about all this in the Web…

Oh yes, the ISINDEX tag should be returned by the server to notify the client that it accepts searches.
Page under construction.

No internet connection?

You are not on the internet? You know you aren't because none of the usual commands are available to you like telnet and ftp. Don't worry: all is not lost. There are two possibilities.

If you are on DECnet

If you are on a DECnet, then you can use DECnet versions of the W3 software. You can get information from outside the DECnet you are on so long as somebody somewhere runs a gateway into the internet on a machine which is connected to both.

By mail

If you have electronic mail, then it possible (though slow!) to get W3 information by mail. try sending a mail to listserv@info.cern.ch with a line in it saying just
HELP
to get back instructions. Your mail system must have a mail gateway onto internet mail, but hat is quite likely. You might have to take the internet address above and ask your friendly system manager how to convert it into the equivalent mail address on the system you are using. See more about the robot.