

C++ And The World-Wide Web

by [M. Speh](#), Hamburg, and [C. Dinnell](#), London.

(Latest version of an [article submitted](#) to C++ Report, SIGS Publication, ed. Stan Lippman/AT&T, accepted for publication January/February 1995)

Abstract

C++ documents and resources are made available on the Internet via the World-Wide Web. It is argued that and explained how an international community of C++ users and vendors can profit from this tool for information retrieval, training and service.

These days, newspapers and journals are full of articles on the Internet and its phenomenal potential. Occasionally the authors discuss the World Wide Web (WWW), but fail to explain how the Internet differs from the Web. Even more often, the reader's justified question "*Why should I care?*" remains unanswered. This article will deal briefly with getting on the Web, and discuss information resources of particular interest to the C++ users' community. At the end, you should have a rough idea of what's waiting for you out there on the Web. Even more importantly, you will know how you can contribute to the Web yourself, as a lone C++ programmer or as a software company, and how you can use it as a sophisticated tool for teaching and learning C++.

Only a minimum of technical information is necessary to get to the Web and use it effectively. The World-Wide Web is a "wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents" which was developed at [CERN](#), the European particle physics facility. First proposed in 1989 by Tim Berners-Lee (CERN) to promote the availability of information within the high energy physics community, it has by now expanded throughout the whole Internet and embraces all scientific disciplines, the liberal arts and the business world. The fantastic growth rate of the Web users' community has made people recognize it as the Internet tool with the greatest potential for the future of the electronic information age. There are other retrieval systems that, individually, provide more limited services on the Internet ([gopher](#), [netnews](#), [ftp](#), [telnet](#) etc) but the Web gives access to all of those services combined (provided of course that the respective programs are installed on your computer).

Most WWW documents are written in [Hyper-Text Markup Language \(HTML\)](#). This is normal text, embedded with links to other documents. A link appears as a highlighted or underlined word, and you move between linked documents by clicking on those words. Hidden behind each link is something known as a "[URL](#)" ([Uniform Resource Locator](#)), an address attached to the file to be accessed (think of it as a networked extension of the standard filename concept). If you only followed links from one document to another, you would not even need to know the concept of URLs, as the link automatically goes to that address and plugs you in to the document. However, you will certainly want to access documents directly at some point. URLs can be intimidatingly long - the file containing a HTML version of this article has the address

`http://www.desy.de/user/projects/C++/report.html`

The string "http" tells you that this document is delivered using the Hyper-Text Transfer Protocol. The second part, "www.desy.de", is an internet address domain name, and the last part is the local path to the requested document. A WWW history mechanism called a "*hotlist*" (not available for all browsers) simplifies the process considerably by allowing you to store automatically, without ever seeing it, the URL of a document you are in with a single click or keystroke. This is extremely useful when you have followed a long path through a number of documents. Most of the references to this article consist in URLs.

The two crucial pieces of software for the Web are document *servers* and *browsers*. Servers are for those providing information, but are irrelevant for those merely interest in retrieving it (see [here](#) for more important information, in particular relevant to companies with access restrictions). However, you must have a browser, also known as a client, to use the Web. The Internet links computers together, and the Web links the documents stored on those computers. A browser allows you to actually look at the documents. The choice of browser depends on your computer and how you are connected to the net. Among the most popular clients is [NCSA Mosaic](#), a graphical client, and [Lynx](#), a simple line-mode browser, both of which are freely available on a variety of computer architectures. You can quickly check out the Web without installing any new software on your system in one of two ways. If you have the "telnet" program and access to the Internet, you can get onto the Web with Cern's simple browser by telnetting to "info.cern.ch". At this point you can follow links by entering the numbers conveniently placed next to them in brackets, or you can use the "go" command to reach a particular URL. In the telnet session, the command

```
go http://www.desy.de/user/projects/C++.html
```

would bring you to the [C++ Virtual Library](#). On the other hand, if you don't even have telnet, you may have e-mail with Internet access. Then you can use a mail robot to serve particular WWW documents to you in ASCII format. The email returned by this program contains the page text and, more importantly, a list of references (URLs) at the bottom, which you can use to climb further down a documentation tree. For example, to get to the C++ Virtual Library HomePage, send email to "test-list@info.cern.ch" with only the line

```
send http://www.desy.de/user/projects/C++.html
```

in the body of the message. The generic format of the "send" command is "send <complete URL>".

Once you are familiar with Web pages, the simplicity of HTML and the special appeal of pages seen through graphical browsers like Mosaic tools are very likely to make you wish to write your own hypertext pages. Once people and companies realise that the Web is a strong paradigm whose usefulness and fascination is not tied to the Internet, they often start out by providing local Web services. These might include self-descriptions (the so-called "[hyplans](#)"), [virtual resumes](#), [diaries](#) and pictures, or a whole environment, software product palette or campus-wide information system. In between these stages may lie several months of getting used to editing HTML, convincing reluctant collaborators of the Web's irresistible charm, and finally a jump onto the Internet, since only there does the full power of WWW unfold. Whether the new friends of the Web realise it or not, they have taken the first step towards virtual management. The [C++ virtual library](#) grew out of a need to be in touch with the C++ world outside a single research institution and to organise search results in an orderly way. It is the power of the Web that it has become a resource used by over 4,000 people a week, and it is clear that the Web is an important resource for making distributed information centrally available in a networked environment. For example, the Web can instantly solve a problem that plagues the daily life of a C++ programmer, namely, dealing with out of date information: no FAQ, library or README information needs to be copied to a single site. All that needs to be maintained (often with the help of sophisticated scripts ([1](#))) is the correctness and accessibility of the link.

In early 1993, the [C++ Virtual Library](#), a central repository of C++ related documents and resources, became available to the public via WWW. Before entering the WWW subject index maintained at CERN, it contained only specific information on the use of C++ in High Energy Physics, but since then has evolved into a huge catalogue of over 250 documents on C++, mainly in HTML format. Examples from the home page include links to:

- A long list of separate pages with C++ and OOP FAQ documents for "[Getting Start\(1\)ed](#)"; interesting C++ applications, book reviews, other archives (mainly with FTPable material), topical resources of interest to C++ users such as parallel applications, the current list of ANSI/ISO resolutions etc.
- "[Learning C++](#)", a page with links to V. Carpenter's course resource list and the Virtual GNA C++ Course (see also below).

- A page with links to access (although not to post to) Usenet newsgroups dealing with [discussion about C++](#).
- Lists of freely available and commercial C++ libraries and packages.
- List of C++ and OOP conferences.
- [Editing](#): An introduction to Barry Warsaw's clever "c++-mode" for the GNU Emacs editor.
- Links to other programming and application hierarchies (like general OOP and Literate Programming), and to pages prepared by software companies offering C++ products.
- A list of upcoming [OOP and computing conferences](#).

The more frequent visitor to this tree is informed about recent changes through a ["What's new"](#) page. An additional essential requirement, namely quick access to the desired resource, is met by a [keyword-searchable index](#), based on the ICE indexing package by C. Neuss, which is part of the [CERN httpd server distribution](#). At the bottom of the page, there is a link to an [HTML fill-out form](#), which can be used to report errors or suggest additional resources directly to the author. Fill-out forms and indices, based on [CGI \(Common Gateway Interface\) scripts](#), are among the most useful of the more recent additions to the technical equipment on the server side.

Forms are becoming an indispensable mechanism for feedback from the Web user to the information provider. Companies advertising their products can put out a "guest book"-like form to be signed by those visiting their pages, or advertise jobs for programmers.

Since its inception and fantastic growth, the Web has been plagued by an "old" Internet disease: the lack of overview for the surfing user. Therefore, the thoughtful use of indices can strongly be recommended to everyone who desires to offer information via WWW.

Still, the kind of index search generally offered by the information provider does not take into account the fact that the resources on the Web are often quite different in character: a single document should be distinguished from a list of documents in an FTP area, or from a volume of issues of a particular magazine. To account for these differences, one must introduce a "coverage code" concept, assigning different coverage indices to resources and forcing the indexing program to follow their hierarchy. The price to be paid is that one cannot fully rely on automated indexing for that - a human has to evaluate a resource to assign the proper coverage code to it, thus creating the problem of outdated indices again. The only Internet (and Web document) index known to the authors which follows a coverage code principle is the ["Meta-Library" "Meta-Library"](#) from the [Globewide Network Academy](#) (2).

The initial confusion which arose from the fact that different servers provided for a widely varying arsenal of tools has disappeared in 1994, and you can now choose between a number of equally well-supported software products (all of which are [freely available](#)).

On the client/browser side, it will remain true that not all software supports all the different gadgets, but which one you can use strongly depends on the hardware available to you - it is the responsibility of the information provider to ensure usefulness of a resource for as wide an audience as possible. Every public page should be cross-checked with at least one graphical and one non-graphical browser (like Mosaic and Lynx or [Emacs-W3](#)), and servers should not forget the visually impaired user when considering a fancy graphical HTML page layout.

The Web has a special attraction if you want to make C++ code available to the world, or even just to a group of developers, since access authorisation or strictly local service is possible. Information is accessible not only as a packed file for complete retrieval, but also by using the Web to provide an intelligent Hyper-map thorough code samples, class libraries and complicated design solutions. Available examples fall into two different groups: straightforward, manual preparation of HTML pages for the code/design or HTML front-end (either integrated in the coding environment or as script-collections). An example for straightforward HTMLisation are [T. Burnett's GISMO development code](#) and the [AIPS++ library for Radio Astronomy](#) - here the Web solution merely consists of a nice embedding of raw source code in a hypertext guide. Another method is to interface C++ code and HTML using a manually run script like D. Bruck's [classdoc.awk](#), turning the class header files into Unix manual pages which can be HTMLised on the fly using suitable scripts. A nice presentation of various C++ projects prepared this way can be inspected through the ["High Energy Physics C++ Catalog](#) (see also [script collection](#)).

A bit fancier is the use of [clickable image maps](#) (an example is [here](#)). allowing for the graphical display presentation of your C++ design process. These can only be accessed with a WWW browser that supports this feature, like NCSA Mosaic. Such a map can contain links either to more design documents or directly to the code, which may or may not be marked up with HTML, or which may be combined with a class2man translation tool like the one mentioned above.

The solution I personally prefer is the use of an HTML front end to code which is written in a [literate programming environment](#) like N. Ramsey's [NOWEB processor](#): here, the code document formatted in LaTeX (a package of macros for Knuth's TeX program which is very popular among scientists) gets turned into HTML with N. Drakos' [latex2html translator](#). As an example, a HTML version of this article was [prepared using latex2html](#), and some [C++ examples are here](#). If you do not want to follow the literate approach, but use latex2html nevertheless for automatic translation into HTML, you can e.g. try the c++2latex program by J. Heitkötter (available from your local [comp.sources.misc newsgroup](#) archive).

Another interesting solution to the presentation of C++ libraries on the Web has been given by P. Murray-Rust from the Glaxo Protein Research group, UK, together with his [Democritos](#) library for bioinformatics: he runs a [tk/tcl](#) (3) script on header files consistently formatted for documentation purposes to produce a very appealing marked-up C++ library reference. To display information, another tcl script can be used which runs a simple, home-made class browser.

For all the different approaches mentioned, the main reason that software authors are interested in publishing their code in hypertext format is that they are seeking international collaboration: the Web material often addresses a user community which is still debating the usefulness of C++ (or even of object-orientation, since Fortran is still the dominating language in scientific applications). The library on the Web often consists of research code that is far from being finished and this attracts hackers, beta-testers and programmers worldwide. Only a few software companies have already smelled the market here and offer C++ products over the Web, but their number is growing. They then often feel the need to contribute to the maze of resources for C++ developers. An example is the [C++ Forum](#) from Quadralay Corp. and product information from [ParaSoft Corp.](#) Valuable free resources often have attractive home pages, like J. Smart's free GUI toolkit ["wxWindows"](#) in C++.

With the present global labor market changes, the need for training and retraining is growing. Distance education is an attractive means to face these changes adequately in an increasingly networked working environment. On this premise the [Macvicar School of Educational Technology \(MSET\)](#), a school under the umbrella of the [Globewide Network Academy, Inc. \(GNA\)](#), a young online university, offered a [C++ course](#) fully delivered via the internet, with the World-Wide Web as its backbone. The course was taught for the first time from May to August 1994, with 75 students from over 18 countries and a faculty of 9 consultants from Canada, Germany, Korea, the UK and the USA. During the course, there was no real life contact between the teachers and the students: instead, realtime interaction was offered online using [Diversity University MOO](#), a multi-user environment with OOP capabilities, originally developed at [XEROX Parc, Palo Alto](#), in which a "virtual classroom" had been built. [Transcripts of interesting MOO discussions](#) were added to the course Web as well. In addition, the students and teachers were able to have discussions on a dedicated mailing list archived on the WWW using K. Hughes' [HyperMail](#) program. The course units followed a [hyper-textbook](#) served on the Web and based on the [Coronado tutorial by Gordon Dodrill](#).

This hypertextbook, which is keyword-searchable as a whole, contains links to compilation notes prepared by the teachers, a glossary of C++ terms, solutions to exercises sent in by students and a fair amount of additional material based on student questions and prepared during the duration of the course. All sample programs are available both as hypertext (marked up as C++ code) to be read parallel to the text by opening several browser windows (compared to reading two pages of a printed book), and in raw ASCII format for quick pasting and compilation ([see example](#)).

There was no tuition or fee asked from the students for this prototypical course. Instead, they were expected to [contribute to the Web material](#), and most of them did - 20 people alone were involved in the preparation of the hypertext notes, and some applied their fresh knowledge of C++ to writing [helpful conversion programs](#). During the course, some interesting programming problems materialized as [student projects](#) that lasted beyond the end of the course, covering a CGI wrapper library, matrix and string classes, etc. These are now organized as the ["GNA Global C++ Library" \(GCL\) project](#), which was sponsored by MSET to conduct research and

development on reusable software tools. This project is being developed in the same spirit as GNU software development, with copyleft protection and drawing on from the vast internet resources of programming talent, free software and emerging advanced teleconferencing software.

The response to this new way of learning using the World-Wide Web was overwhelmingly positive: at the 1st International WWW conference at CERN, Geneva, in May, the course won the 1994 award for "Best Educational Service on the Web". Its successor, planned for October 94, was already oversubscribed before the first course had terminated, and the massive feedback from the first class is leading to a major review and restructuring of the material. The way the different components of the course - WWW, maillists and online consulting - can work together still needs to be optimized, but the concept of distributed learning with the Web, even for a topic as complex as the C++ language, has undoubtedly proven to be very successful and is already giving rise to imitation. At workshops like the one on ["Teaching and Learning with the Web"](#) during the 1st WWW conference, the long- distance education community is now meeting and discussing with WWW wizards how to profit from this new approach to teaching and learning.

Programming courses in particular promise to be rewarding targets for this kind of learning on the Web: the students can initially be assumed to be computer literate, and the presentation of the material is usually easy to markup as HTML (compared to, for example, a course relying on complicated mathematical formulae). Also, no field is better represented on the net or the Web than computer science, so that any course can immediately offer a enormous pool of secondary resources. (Here is another [paper on the virtual C++ course](#) and its implicit vision, by D. Perron.

More and more software vendors and academic institutions are trying to pace their way into the electronic world of the 21st century. The World wide web, as the most powerful paradigm around on the internet, is here to stay. Through various services, both from research labs (which still make up the majority of server sites) and companies, C++ is already well represented on WWW. As automatic tools to translate existing documents into HTML - the standard Web markup language - become more widely available, whole software product palettes can conveniently be advertised and distributed on the Web. There is no limitation of WWW to be used only on the Internet: rather, companies may gain considerably from an in-house Web server already. On the design and development side, there are several interesting proposals of how to lucidly display C++ library information. For C++ training and programming courses in general, virtual coursework following the example of the GNA C++ course can be organized and delivered to customers, employees and students using the Web. Additional reading: K. Hughes' freely available text ["Entering the World-Wide Web: A Guide to Cyberspace"](#) answers most of the immediate questions which this article with its limited scope cannot address.

(1): Oscar Nierstrasz' [collection of scripts](#) is marvellous. See also [here](#).

(2): The GNA's home on the Web was awarded "Best Campus-Wide Information System" in the WWW contest 1994.

(3): tk/tcl is a widely used programming system for developing and using graphical user interfaces.

Copyright (C) 1994 [M. Speh](#) and [C. Dinnell](#).