

# Contents

About HTML Assistant Pro

Getting Started

Beginners' Guide to HTML

Creating your First HTML Document

Organizing your Internet Experience

HTML Assistant Pro Menus

HTML Assistant Pro Tools

Background Assistant

QuickFrames

URL Assistant

Table Assistant

Form Tool Bar

The URL Editor

The Automatic Page Creator

Viewing your File

HTML Reference

Figures

# About HTML Assistant Pro

[Introduction](#)

[Subscribe to the HTML Assistant Newsletter](#)

[Disclaimer](#)

[Copyright Notice](#)

# Getting Started

Software Installation and Registration

Setting your Test Browser

# **Beginners' Guide to HTML**

**Preamble**

**Introducing HTML**

**HTML Tags**

**The Document Title**

**Headings**

**Paragraphs and Text Formatting**

**Preformatted Text**

**Lists**

**Text Formatting by Function**

**Text Formatting by Font Style**

**Links to Other Documents**

**URLs**

**Linking to Anchors Within the Same Document**

**Displaying In-line Images**

**Displaying Images With a 'Viewer'**

**Using In-line Images as Links**

**A Brief Tour of Forms**

**Creating Tables**

**The Structure of an HTML Document**

**To Learn More About HTML**

# Creating your First HTML Document

[Before You Begin](#)

[Opening a Document Window](#)

[Set Up a Standard Document Template](#)

[Create a Title](#)

[Add a Heading](#)

[Testing your Work](#)

[Possible Source of Confusion While Testing](#)

[Adding Text](#)

[Creating Lists](#)

[Linking to Another HTML Document](#)

[Displaying an In-line Image](#)

[Displaying an Image with a Viewer](#)

[Going Further](#)

# Organizing your Internet Experience

Overview

Collecting URLs

Editing and Organizing URLs

Building Browser Pages

Using Local Files with HTML

# HTML Assistant Pro Menus

## Overview

### The "File" Menu

### The "Edit" Menu

### The "Options" Menu

### The "Filters" Menu

### The "HTML" Menu

### The "Command" Menu

### The "Special" Menu

### The "URL" Menu

# **HTML Assistant Pro Tools**

**Tools Overview**

**Using Bracketing (Type 1) Tools**

**The 'Link' Tool**

**Using Insertion (Type 2) Tools**

**User Tools**

**Editing User Tools**

**User Tools Files**

**Creating Tools that Insert an End-of-Line**

**The Auto Repeat Function**

**Autoinsert Paragraph Markings**

**Printing**



# **Background Assistant**

**What is Background Assistant?**

**Starting Background Assistant**

**Using a Tiled Image as a Background**

**File Conversion: Use of Files Other Than GIF or JPG**

**Transferring the <BODY ... > Tag to your File**

**Preselecting an Existing <BODY ....> Tag**

**Where to Insert the <BODY ... > Tag**

# QuickFrames

Overview

Using QuickFrames

# URL Assistant

[URL Assistant Overview](#)

[Transferring URLs from the URL Assistant List](#)

[Searching the List](#)

# Table Assistant

Table Assistant Overview

The Table Assistant Tabs

# Form Tool Bar

Form Tool Bar Overview

The Form Tools

# **The URL Editor**

**Edit/Build URL Files**

**Operation of the URL Editor**

**About Netscape (and other browsers') "Bookmark" files**

**Combining Files**

**Direct Conversion of Files to HTML Text**

**Cello Bookmark files (\*.bmk files)**

**HTML Assistant URL Files (\*.url files)**

**Mosaic Initialization Files (Mosaic.ini)**

# **The Automatic Page Creator**

**What is the Automatic Page Creator?**

**Starting the Automatic Page Creator**

**Using the Automatic Page Creator**

**Building HTML Displays by Appending New Pages**

## Viewing your File

Using a WWW Browser for Testing

Using Dynamic Data Exchange (DDE) with a Browser



# HTML Reference

[Overview](#)

[Comments](#)

[Document Structure Elements](#)

[Block Formatting](#)

[Character Formatting](#)

[List Elements](#)

[Links](#)

[Images](#)

[Forms](#)

[Tables](#)

[Netscape Extensions](#)

# Figures

Initial document window

Link to file dialog

Inline image dialog

Sample form

Sample table

URL Manager window

Text Search dialog

File Search dialog

# Table of Contents

# Introduction

HTML Assistant Pro ©1993-1996 by Howard Harawitz and Brooklyn North Software Works, Inc., All Rights Reserved

**Important: Click here to see license agreement.**

***HTML Assistant Pro* has Context Sensitive Help-When using the program, press the F1 key for help with a selected control**

*HTML Assistant Pro* is a simple text editor with extensions to assist in the creation of HTML hypertext documents. HTML documents are used to access a variety of information sources with a World Wide Web browser.

The Automatic Page Creator included with *HTML Assistant Pro* permits very rapid creation of HTML text pages for use with WWW browser programs. Minimal knowledge of HTML is required, and attractive screens with your own selected hypertext links can be created in a few seconds.

*HTML Assistant Pro's* other features include:

Ability to quickly convert formatted word processor files to HTML

Tool bar support for HTML 2 and HTML 3 features including forms, backgrounds and tables

Automatic creation of frame pages with a powerful *QuickFrames*™ feature

A built-in HTML viewer plus the capability to test your work with a WWW browser at the click of a button - without leaving the editor

A user defined tool box that permits creation and use of new HTML tags as they become available

Seamless integration with the instant WWW page publisher, *LinkStar*™ *Site Launcher*

Point and click creation of HTML Hypertext markings

Ability to load and edit files of any size

The ability to remove HTML markings and reformat text to create readable documents from HTML files.

File search features-you can search for and load files containing key words which you select.

A multiple document interface (more than one file may be opened at one time)

Automatic conversion of browser Bookmark, Mosaic '.INI' and URL text files to hypertext documents

Easy transfer of URLs from browser Bookmark files to your own HTML files

An option to make UNIX text files more readable by converting them to DOS text

An option to save DOS text files as UNIX text

Ability to print HTML text files

## **Subscribe to the *HTML Assistant Newsletter***

For continuing updates and news about *HTML Assistant* software, you can subscribe to an Email publication, the *HTML Assistant Newsletter*. Published "semi-irregularly", and delivered (free!) directly to your (virtual) door, the *Newsletter* includes announcements, bug reports and news about *HTML Assistant* software. It also is a source of information of general interest to users of the World Wide Web -- including text and HTML pages about learning HTML, interesting WWW sites, commentaries on the use of the Web, feedback and suggestions from users, etc.

To subscribe, send Email, with the word "subscribe" (quotes are not necessary) in the 'Subject:' field, to: [news@brooknorth.com](mailto:news@brooknorth.com). You may also include comments or suggestions in the text field of your message.

To find out what's new, be sure to check our Web site:

<URL: <http://www.brooknorth.com>>

## Disclaimer

Brooklyn North Software Works, Inc. disclaims any liability for special, direct, indirect, incidental, consequential or any other damages believed to have been caused by the use of *HTML Assistant Pro* or any files derived from its use.

# Copyright Notice

**HTML Assistant Pro and its associated documentation is Copyright (©) 1993-1996 by Howard Harawitz and Brooklyn North Software Works, Inc., All rights reserved.**

**Use of *HTML Assistant Pro* implies consent that this notice is a legal agreement between the purchaser of *HTML Assistant Pro* and Brooklyn North Software Works, Inc.**

This software and its documentation is owned by Brooklyn North Software Works, Inc. and Howard Harawitz and is protected by Canadian and US copyright statutes and international treaty provisions. Reproduction of this software or its documentation, other than as described below, without express consent of the owner is not permitted.

Purchase of an authorized copy of the *HTML Assistant Pro* diskette constitutes possession of a single license for use of one copy of the program by one person at a time.

This means that the licensee has the right to

1. Install a copy of the software on each of those computers reserved for the licensee's own use, as long as no more than one person uses the software at the same time.
2. Make copies of the software to be used for normal backup purposes

Networks: In a networking environment only one copy of the software may be running at one time for each license purchased.

Respect for the terms of this license will ensure that Brooklyn North Software Works can continue to provide new software products at reasonable prices.

Multiple licenses, bulk purchase options, and additional copies of the documentation are available.

Thank you for taking the time to read this.

**Important: [Click here to see disclaimer](#)**

For further information please contact Brooklyn North Software Works, Inc.:

Email: [sales@brooknorth.com](mailto:sales@brooknorth.com)

Telephone: (902) 425-0900

FAX: (902) 835 - 2600

## Software Installation and Registration

Before installing *HTML Assistant Pro*, be sure to backup the original program diskette(s).

Because the installation procedure may change, installation instructions are supplied separately. Follow the instructions to install the software.

This would also be a good time to fill out and send off the registration form. You can register via Email by sending an Email message that includes all the information requested in the form. Be sure to send the version, version date and serial number as shown on the diskette label.



## Setting your Test Browser

You can begin using *HTML Assistant Pro* as soon as it is installed and running.

However, if, as you are editing a document, you want to view or "test" your work with a World Wide Web browser, you need to let the software know where your test browser can be found.

You can tell HTML Assistant Pro which browser to use by selecting the option, "Set test program name...", from the "File" menu. The dialog boxes will lead you through the process. Using the appropriate list boxes, select the path and filename for your browser. If you want the software to "remember" where the browser can be found, so that you don't have to repeat the process each time you start the program, select the "Yes" button when asked if you want to save it as a "permanent test program name".

If you don't set your test program name, *HTML Assistant Pro* will ask you for it the first time that you attempt to test a file.

## Preamble

Be aware that the Internet is always changing. Because the World Wide Web is so new, and growing so quickly, it is even more susceptible to the vagaries of change than are other aspects of Internet life.

The unusually rapid growth of the Web means that you will encounter (virtually) lots of people who are experimenting and learning, just as you are.

Most companies, schools, individuals and other agencies are beginners at setting up WWW sites. Some of them may change service providers from time to time. They will be working on their sites a lot. They will move things around.

This can result in the disappearance of documents and their mysterious reappearance in different directories or even on servers in other parts of the world. As a result, links that yesterday, or even fifteen minutes ago, led you to beautiful works of art, may actually cause your system to crash at this moment.

Fledgling HTML writers are creating many of the pages that you will want to access. Due to their errors as beginners, some of the links on those pages may not work as you might expect -- or they might not work at all. This, too, will often lead to "unexpected" (when this happens, other words will most certainly come to mind!) results.

It may be hard to believe, but even the authors of HTML editors are still learning, and can be responsible for inadvertent errors that creep into HTML documents found on the Web!

When it comes to working with the World Wide Web most of us are newbies.

So, please have patience--

with those who display WWW documents

with your Internet service provider

with software authors (!)

with everyone else you encounter as a result of your decision to explore the World Wide Web

And, of course, have patience with yourself.

# Introducing HTML

*Note: This guide was prepared so that HTML beginners could get a quick-start at creating HTML documents. It is by no means a complete exposition of what can be done with HTML.*

*For more information about advanced HTML features, check the World Wide Web itself, along with the increasing number of books on the subject that are appearing on bookstore shelves.*

World Wide Web browsers use hypertext to access and display information available on the World Wide Web. Hypertext is text that incorporates pointers to additional information. Microsoft Windows "Help" files are examples of hypertext documents.

The hypertext documents used by World Wide Web browsers are ordinary text files that include special markings that tell the browser software something about how the text should appear to the user, as well as how to retrieve information when the user requests it.

The language, or code, used for these special markings is "Hypertext Markup Language", abbreviated as "HTML".

An Example

## An Example

A very simple HTML document would look like this when displayed in a text editor:

```
<title>An elementary HTML document</title>
```

```
<h1>The Main Heading for the Document</h1>
```

This is the text that would follow the heading. It is plain text and must be separated from the following paragraph with a special tag. <p>

This is the following paragraph. <p>

A browser might display the above HTML text as follows :

An elementary HTML document

### The Main Heading for the Document

This is the text that would follow the heading. It is plain text and must be separated from the following paragraph with a special tag.

This is the following paragraph.

The HTML markings indicate only what the function of the text is, (i.e., a heading, plain text, a link to a document, etc.) not how it will appear.

*The browser determines what the text will look like.*

The same HTML document may look very different when displayed by different browsers (or even the same browser as configured by another user).

# HTML Tags

The markings used in HTML documents are called "tags."

An HTML tag consists of a left angle bracket (<), called a "less than" symbol, followed by some text, and ends with a right angle bracket (>), called a "greater than" symbol.

Tags are most often used in pairs, i.e., <H3> and </H3>.

The most significant exception to this has been the tag that separates paragraphs, the "<P>" tag. In the original HTML specification there is *no* ending, "</P>", tag. However, this may change in the future.

In the example, the starting tag, "<h1>" tells the browser that a heading is about to begin. The end tag, "</H1>" lets the browser know that the end of the heading has been reached.

It should be noted that HTML tags may displayed in either upper or lower case (or a mixture of the two). "<H1>" is equivalent to "<h1>". HTML is *not* case sensitive.

# The Document Title

Every HTML document needs a title.

A title, bracketed by its tag pair, looks like this:

```
<TITLE>This is the document title</TITLE>
```

The function of the title is to identify and describe the document (for use by searching programs, for example) rather than as a display element ("*Heading*" tags are used for display purposes).

There should only be *one* title in an HTML document.

# Headings

Headings are used to separate and call attention to different parts of a document.

HTML incorporates six levels of headings. They are numbered 1 through 6, with heading level 1 being the highest level and the most prominently displayed.

Headings are displayed with larger, bolder typefaces than normal text. A heading is displayed on its own line and does not need a paragraph tag to separate it from the following text.

Normally, the first heading in a document should be tagged as a level 1 heading, e.g.,

```
<h1>This is the First Heading</h1>.
```

The text portion of the heading tag is "H" followed by the heading level. If "n" represents the level, a properly tagged heading looks like this:

```
<Hn>This is a heading<Hn>
```

where "n" represents a number between 1 and 6.

## Paragraphs and Text Formatting

Beginners are often confused by the way that HTML treats white space and carriage-returns (newlines) in the text.

Browsers generally ignore carriage-returns and white space in your HTML documents. When your file is displayed, repeating spaces may be shown as a single space and line breaks can occur anywhere. That is why each paragraph in an HTML document *must* end with the "<p>" tag.

Without paragraph tags, what looks like a well formatted text file will be displayed by a browser as a single continuous paragraph!



## Preformatted Text

To display text so it is formatted the way it looks when viewed with an ordinary text editor, you need to use a special tag to indicate that the text is "preformatted".

The tag looks like this: `<pre>`. The closing tag is: `</pre>`.

Use of this tag will cause the text to be displayed with spaces, new lines and (usually) tab characters as they have been entered., that is, as it has already been formatted. A fixed width font (i.e., Courier) is normally used.

# Lists

You can use HTML to display lines of text as a list.

Text may be displayed as a numbered or an unnumbered list.

A list is bounded by an opening and a closing tag. Numbered and unnumbered lists have different tags.

Paragraph separators are not required to separate the individual list items. Each list item begins with its own tag which serves as a separator. This "list item" tag is the same for numbered and unnumbered lists.

The HTML for an *unnumbered* list looks like this:

```
<UL>
<LI> List item one
<LI> List item two
<LI> List item three
</UL>
```

A browser might display the unnumbered list as follows:

- List item one
- List item two
- List item three

The HTML for a *numbered* list (also called an "ordered" list) looks like this:

```
<OL>
<LI> List item one
<LI> List item two
<LI> List item three
</OL>
```

A browser might display the numbered list as follows:

1. List item one
2. List item two
3. List item three

Lists may be nested, i.e., lists within lists are permitted.

## Text Formatting by Function

Using HTML, you can indicate formatting for several different kinds of text functions.

The following are some text functions and the HTML tags used to display them:

- Extended quotation (usually several lines of quoted text):

`<BLOCKQUOTE> </BLOCKQUOTE>`

- Address (usually an Email address):

`<ADDRESS> </ADDRESS>`

- Citation (as in referring to another document):

`<CITE> </CITE>`

- Strong (indicates text is important):

`<STRONG> </STRONG>`

A browser might display this HTML text,

`<ADDRESS>harawitz@fox.nstn.ns.ca</ADDRESS>`

like this:

*harawitz@fox.nstn.ns.ca*

## Text Formatting by Font Style

You can also indicate formatting by font style, as you would in a word processor. That is, text can be displayed as italic, bold, etc. Formatting by font style may be discontinued in future versions of the HTML specification.

Some of the styles you can use, along with their HTML tags are as follows:

- **Bold** (text is displayed in bold face type):  
`<B> </B>`
- **Italic** (text is displayed in italic type) :  
`<I> </I>`
- **Underline** (text is displayed as underlined):  
`<U> </U>`
- **Fixed width** (text is displayed in a fixed width font):  
`<TT> </TT>`

This line:

An example of `<I>italic text.</I>`

would be displayed as:

An example of *italic text*.

## Links to Other Documents

HTML documents are "hypertext" documents. This means that they contain text that permits browsers to access additional information when the user requests it.

Browsers indicate that text is linked to more information by displaying it in a different color, underlining it, displaying it in a box, or some combination of these. With a graphical browser like Netscape or Microsoft Internet Explorer, a mouse is used to 'click' on the highlighted text. This tells the browser to retrieve the information associated with the link.

HTML uses a left-angle bracket, followed by the "anchor" code, "A", to indicate the beginning of a hypertext link.

The word "HREF", following the anchor character is used to indicate that the anchor is sensitive text. That is, the text that is displayed is linked to other information and will respond to user input such as a mouse click.

For example, a link that would instruct a browser to fetch and display another HTML hypertext page looks like this:

```
<A HREF= "http://cs.dal.ca/ftp/htmlasst/htmlafaq.html"> View the HTML  
Assistant FAQ</a>
```

The browser would display the text between the first right-angle bracket and the "</A>" end tag. The user would see only the words, "View the HTML Assistant FAQ". It would be highlighted to indicate that it was a link.

Clicking on the text would display the document referred to by the quoted text below:

```
"http://cs.dal.ca/ftp/htmlasst/htmlafaq.html".
```

The above text (within the quotes) is a "Uniform Resource Locator", and would *not* be displayed by the browser. Uniform Resource Locators are pointers to information. In this case the Uniform Resource Locator points to another hypertext document.

# URLs

"URL" is an abbreviation for "Uniform Resource Locator".

URLs are the pointers to information to be retrieved when a hypertext link is activated by the user. They tell the browser how the information is to be obtained and where it is located.

URLs enable a single WWW browser program to incorporate many of the functions that normally require several other software packages, including FTP, news, Email and gopher.

The general format of a URL is :

```
prefix host.domain[:port]/path/filename
```

Note: The space after the word, "prefix", is used only for clarity. In a URL, there is no space following the prefix.

The port number may be omitted from the URL, unless it differs from the recognized standard port for the indicated service.

The following are some of the prefixes that are commonly used, along with their function:

`ftp://` [retrieves a file from an FTP server]

`http://` [retrieves a file from a World Wide Web server]

`file://localhost/` [retrieves a file that resides on the same computer as the browser - a "local" file]

`gopher://` [retrieves a file from a Gopher server].

Some URLs do not use the double slash (//).

`mailto:` [requests the browser to enable the transmission of an Email message]

`news:` [retrieves a Usenet news message]

The following are examples of properly constructed URLs:

`"ftp://ftp.cs.dal.ca/htmlasst/htmlasst.zip",`

would be used in an HTML document to enable a browser to retrieve the file, "htmlasst.zip", from an FTP site;

`"file://localhost/c:/www/myfile.txt",`

would cause the browser to retrieve and display the file, "myfile.txt" from the user's local disk drive;

`"http://www.cfn.cs.dal.ca/Media/News/TodaysNews.html",`

would result in the retrieval and display of an HTML file from a WWW server.

## Linking to Anchors Within the Same Document

There is an anchor that can be used to permit text *within* an HTML document to be the *destination* of a link. This is useful for creating links that refer to text inside the current document. A "point and click" table of contents, for example, is created by using internal links.

In destination anchors, the word, "NAME", follows the anchor character ("A") as shown below:

```
<A NAME=math>Algebra</a> is a very old branch of mathematics.
```

This would give the word "Algebra" the anchor name, "math".

The following HTML code creates a link to the anchor referenced by that anchor name:

```
<A HREF="#math">Click here to learn about algebra</a>
```

Clicking on the displayed text, "Click here to learn about algebra", would move the browser's viewing cursor to the place in the current document beginning with the word, "Algebra", because it has the anchor name "math".

The "#" sign is used to indicate that a link is a reference to a named anchor within a document.

Using named anchors is useful for creating a table of contents. The table of contents would have "href" links to the appropriately named anchors that mark topics within the body of the document.

The first word in the table of contents could itself be a named anchor. This would permit links further down in the text to return the user to the table of contents.

## Displaying In-line Images

An important feature of World Wide Web browsers is their ability to display images.

The "IMG" element is used to allow the display of "in-line" images.

In-line images are images displayed by the browser on the browser page itself, as opposed to the display of images by special display programs called "viewers." HTML permits both kinds of displays.

Most browsers can display in-line images stored in "graphic interchange format" (referred to as "gif" files). Other image formats such as "tiff", "jpeg" and "xbm" can be displayed in-line by some browsers.

An in-line image, from a local disk drive, may be included in a document by using the following code:

```
<IMG SRC="file:///localhost/C:/images/sailboat.gif">
```

Execution of the above HTML code by the browser would cause the file, "sailboat.gif", residing on the user's disk drive, to be displayed as an in-line image.

Note that the image element has no closing tag.

Text that is adjacent to an image is by default aligned to the bottom of the image.

The "align" attribute can be used to alter the placement of adjacent text.

The HTML code,

```
<IMG ALIGN=top SRC="file:///localhost/C:/sailboat.gif"> The good ship Lollipop
```

will cause the text, "The good ship Lollipop", to be displayed aligned at the top of the image. Use of "align=middle" will result in text alignment at the image's vertical center.

Use of the "alt" attribute is used for alternative text that will be displayed by browsers that are unable to display images.

For example, the following HTML,

```
<IMG SRC="file:///localhost/C:/sailboat.gif" ALT="Picture Here"> ,
```

will cause the words "Picture Here" to be displayed by browsers having only text capability.

Both the "alt" and "align" attributes may be used at the same time with the same image element.



## Displaying Images With a 'Viewer'

Sometimes it is desirable display an image outside the browser in another window.

You can cause an image to be opened in a separate window, using a program called a "viewer", with the following HTML text:

```
<A HREF="file://localhost/c:/images/prince.gif">The Prince of Norway lighting  
the Olympic Torch</A>.
```

When the user clicks on the highlighted text, "The Prince of Norway lighting the Olympic Torch", a GIF viewer will be launched and the picture will be displayed.

For this to work properly, the browser must be set up to find and run the appropriate viewer for GIF images. For MS Windows browsers this information is usually stored in the browser's Windows Initialization (".ini") file. Check your browsers instructions for the specifics.

Viewers for video and sound files are called up in the same way as are viewers for images. Again, the browser's initialization file must be set up properly in order for this to work.

## Using In-line Images as Links

External viewers are often used to display a larger image when the user clicks on an in-line, but smaller, version of the same picture, called an "icon".

This is possible because in-line images can be used in anchors, as follows:

```
<A HREF="big_image.gif"><IMG SRC="small_icon.gif"></A>
```

The line above would cause the file, "small\_icon.gif", to be displayed as an in-line image which *is also an active link*. When the user clicks on the icon, the file "big\_image.gif" would be loaded to a viewer and displayed.

Similarly, the following code,

```
<A HREF="ftp://ftp.cs.dal.ca/htmlasst/htmlasst.zip"><IMG  
SRC="button.gif"></A>,
```

would result in the retrieval of a file from an FTP site when the user clicked on the "button.gif" in-line image.

## A Brief Tour of Forms

As the World Wide Web is being used more and more by companies for on-line business transactions, the use of forms is becoming more prevalent. HTML forms allows the Web page designer to create input screens that are used for such things as questionnaires, product ordering forms, and so on.



**Product Ordering Form**

Products to order:

- Milk
- Butter
- Eggs
- Juice
- Apples
- Oranges

Shipping Address:

Payment Method: ☐ VISA ☐ Master Card ☐ COD

**Send Order** **Reset**

How are the special items on this form created? There are a number of different types of fields which can appear on forms, including text entry fields, lists, radio buttons, check boxes, and buttons. HTML tags are provided to define each of these field types in a document.

Each form must be enclosed by <FORM...> ... </FORM> tag pair. A number of attributes are required to specify how the data in the form is processed when it is submitted. For example, the form might use the following FORM tag:

```
<FORM METHOD=POST
      ACTION="http://oranges.com/orders">
...
Remainder of form definition goes here.
...
</FORM>
```

When a user fills in a form and submits it for processing, the data is encoded in a special format called CGI (Common Gateway Interface) and sent to the server which then passes it onto the program or "gateway script" specified by the ACTION attribute. Unfortunately, it is beyond the scope of this tutorial to describe how these scripts are written. Suffice it to say that the general function of these scripts is to decode the form data which it receives and based on the information it extracts, determine what actions to take.

To further elaborate this example, this form has five main fields: a list, a text field, a set of radio buttons,

and submit and reset buttons.

A list such as the one in this form is created using the SELECT element of HTML. This particular example might look like this:

```
<FORM METHOD=POST
      ACTION="http://oranges.com/orders">
<P>Products to order:<BR>
<SELECT NAME="products" MULTIPLE>
<OPTION>Milk
<OPTION>Butter
<OPTION>Eggs
<OPTION>Juice
<OPTION>Apples
<OPTION>Oranges
</SELECT>
...
Remainder of form definition goes here.
...
</FORM>
```

A <SELECT> ... </SELECT> tag pair can only contain multiple <OPTION> tags defining the elements which make up the list. The exact manner in which the form is rendered is determined by the browser. Some present it as a scrollable list and others as a pop-up list. The MULTIPLE attribute further defines the behavior of the list. If specified, it allows the user to select more than one item from the list; if omitted, only a single item can be selected. The value specified with the NAME attribute is used during the encoding phase when the form data is submitted for processing.

The next field is called a TEXTAREA. Such fields allow multiple lines of text to be entered by the user. In this particular example, the following HTML code would be used to create the text field:

```
<P>Shipping Address:<BR>
<TEXTAREA NAME="address" ROWS=4 COLS=35>
</TEXTAREA>
```

This creates a text field 35 characters wide by 4 lines high. The exact manner in which this field is rendered is determined by the browser. Often, horizontal and vertical scroll bars are included to allow the user to scroll the text as it is entered. If text is included between the <TEXTAREA> ... </TEXTAREA> tags, it is used to set the initial contents of this field. As with the SELECT tag, the NAME attribute is used during the encoding phase.

Following the text field is a standard set of radio buttons. Only one of these can be on at any one time. The following code is used to create these buttons:

```
<P>Payment Method:
<INPUT NAME="pay" TYPE=RADIO VALUE="visa">
      VISA
<INPUT NAME="pay" TYPE=RADIO VALUE="mc">
      Master Card
<INPUT NAME="pay" TYPE=RADIO VALUE="cod">
      COD
```

The INPUT tag is used to create a number of different fields types, as specified by the TYPE attribute.

The VALUE attribute is used for encoding purposes, as is the NAME attribute. This attribute is also used to define the grouping of the radio buttons; any RADIO type input field with the same value for NAME as another RADIO type field is considered to belong to the same group implying that only one of those radio buttons can be selected at one time.

The form ends with two standard looking buttons, one labelled "Send Order" and the other "Reset". These buttons are defined using the following HTML code:

```
<P><INPUT VALUE="Send Order" TYPE=SUBMIT>  
<INPUT TYPE=RESET>
```

A RESET type input field creates a button called "Reset". If this button is clicked by the user, any values entered into any field of the form are cleared and the form's initial state is restored.

A SUBMIT type input field creates a button as well. By default, a submit button is called "Submit Query", although in this case the VALUE attribute is used to provide the button with an alternative name. When the user clicks a submit button, the current data existing in the form is encoded into CGI format and submitted to the form's specified gateway script for processing.

## Creating Tables

This section gives an overview of how tables are created in HTML. The basic approach is to define a structure which provides information about each row of a table and each cell in that table. Tables can also have headers, both for the table itself (called the caption) as well as for the rows and columns. HTML tags are used to describe all this information. As an example, take a look at this table.

Product	Warehouse 1	Warehouse 2
Milk	8	12
Butter	19	17
Juice	23	20
Total	50	49
Grand Total	99	

This demonstrates a simple table listing products stored in two separate warehouses, along with the total number of products in each warehouse as well as the combined grand total.

All tables must be enclosed in `<TABLE> ... </TABLE>` similar to forms. All other tags must appear inside these enclosing tags. The table has a title or "caption" associated with it. This is an optional element which can appear inside a table definition:

```
<TABLE BORDER>
<CAPTION>Product Inventory</CAPTION>
...
Remainder of table definition goes here.
...
</TABLE>
```

The `BORDER` attribute of the `TABLE` tag specifies whether borders should be drawn around the cells of the table. A value can also be supplied with `BORDER` to specify the width of the border when the table is rendered. The default is 1; if `BORDER` is omitted, it is the same as specifying `BORDER=0`.

The individual rows of a table are each enclosed inside `<TR> ... </TR>` tags. There are options which control how the data in each cell of a row is aligned within the cell horizontally and vertically and how it is formatted, for example, whether or not the text can be wrapped in the cell over multiple lines. It is also possible to specify how many columns and rows an individual cell spans. Netscape adds a few other options, such as being able to specify the exact width of a cell in pixels.

The first row of the table consists entirely of *table headers*. It is defined with the following HTML code:

```
<TR>
  <TH>Product</TH>
  <TH>Warehouse 1</TH>
  <TH>Warehouse 2</TH>
</TR>
```

Each header cell is defined with `<TH> ... </TH>` tags. Each such tag pair appearing within a row definition represents a cell in that row. Text enclosed by this pair is automatically boldfaced and centered within the

cell when the table is rendered.

Other table cells are created in a similar manner, using the *table data* <TD> tag instead of the <TH> tag. The second row of the table, for example, can be defined using the following HTML code:

```
<TR>
  <TH>Milk</TH>
  <TD ALIGN=CENTER>8</TD>
  <TD ALIGN=CENTER>12</TD>
</TR>
```

The <TD> tag can take optional attributes specifying characteristics about an individual cell. In this case, the ALIGN attribute is used to center the contents of the cell. Without this option, cell data is left aligned. This differs from header cells defined with <TH> tags which are centered by default. Each <TD> ... </TD> appearing in a row definition represents a cell in that row. The total of table header tags and table data tags in a row definition represents the number of cells in that row.

The other rows in the table are defined in exactly the same way, with only the data changing as required. The exception is the last row, which contains just two cells instead of three. One of the attributes which can be specified with a cell definition is the number of columns it takes up. In this case, the second cell of the last row, containing the grand total value, has been defined to take up two columns of the table instead of the normal single column. This is done as follows:

```
<TR>
  <TH>Grand Total</TH>
  <TD ALIGN=CENTER COLSPAN=2>99</TD>
</TR>
```

The COLSPAN attribute in the <TD> element above specifies that this cell should span two columns. A cell can also be modified with the ROWSPAN attribute to specify the number rows the cell spans, and both COLSPAN and ROWSPAN can be used at the same time.

# The Structure of an HTML Document

The HTML specification requires that HTML documents be composed of structural elements that are bounded by special tags.

The entire document must be bracketed by the "HTML" tag pair as follows:

```
<HTML>
Document contents
</HTML>
```

Within the HTML pair, three other pairs of tags are obligatory. First, the "head" pair. Within the head pair, a *title* with its tag pair must be used. The remainder of the document should be bracketed by the "body" pair.

The following is a template for a properly structured HTML document:

```
<HTML>
  <HEAD>
    <TITLE>Document Title goes
    here</TITLE>
  </HEAD>
  <BODY>
    The actual content of the
    document goes here.
  </BODY>
</HTML>
```



## To Learn More About HTML

While the above is by no means an exhaustive tutorial, it includes sufficient information for a beginner to create useful, interesting and attractive HTML documents.

Several books that can help you to learn more about HTML are available at most book shops that stock computer books. This manual is only a preliminary guide to the use of basic HTML tags. To learn more, check your local bookstore.

Use the Web to obtain additional information. Check the major WWW home pages, i.e., W3O (home of the World Wide Web), Microsoft, Netscape Communications or NCSA (home of Mosaic) - along with search engines and indexes such as *Yahoo*. If you follow the links to "HTML", "Learning HTML", "Creating WWW Documents" or similar subjects, you will find tutorials, information on specialized uses of HTML (i.e., forms, tables, image maps, etc.), along with guides to good HTML style.

An excellent way to learn how to use HTML effectively is to get the source code for documents that have features of interest to you. Most browsers will permit you to save the HTML tagged documents you are viewing. You can then examine the document's HTML code to see how particular kinds of displays have been created.

Finally, for a thorough and up to date treatment of the subjects covered here, use the Web to acquire the current HTML and URL specifications.

## Before You Begin

Before you start to create a document, you should become familiar with the elements and structure of HTML. The section of the *HTML Assistant Pro Manual* titled "Beginners' Guide to HTML" provides all the background information you need to make interesting and informative HTML pages. If you haven't already done so, please read it before going any further.

Once you have learned something about the syntax of HTML, you can use this section to learn how to use *HTML Assistant Pro* to actually build an HTML document.

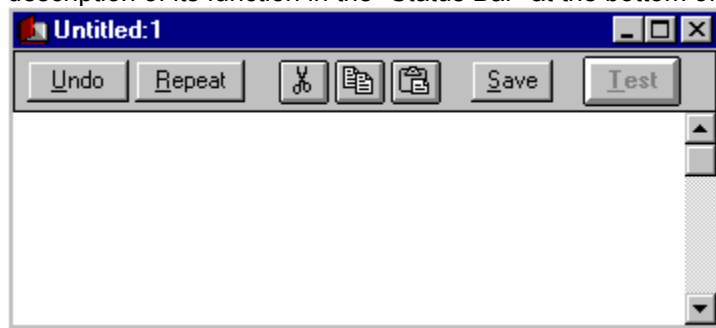
The text that follows is a step-by-step guide to HTML document creation. It is meant be used while you are working at your computer and having started the *HTML Assistant Pro* software.

## Opening a Document Window

To create a new document, open an empty document window by clicking the button with a picture of a blank sheet of paper on it. This is the "New file" button and can be found at the left end of the tool bar. Alternatively, you can select the "New" option from the "File" menu. Using the button is quicker, but both methods accomplish the same thing. An empty document window is created as a result of this action.

*HTML Assistant Pro* document windows behave very much like the text window in the *Notepad* text editor included with *Microsoft Windows*. You can enter, select (using the mouse), and delete text. You can also cut or copy text and paste it to other parts of a document.

If you are in doubt about the function of any control on *HTML Assistant Pro*'s tool bars, document windows, or special function windows, simply move the mouse pointer over the control and you will see a description of its function in the "Status Bar" at the bottom of the main window.



# Set Up a Standard Document Template

After opening the document window, open the "Command" menu and select "Display standard document template".

This will cause the following text to appear at the top of your document:

```
<HTML>  
<HEAD>  
<TITLE> </TITLE>  
</HEAD>  
<BODY>  
</BODY>  
</HTML>
```

These tags, which delineate the sections and boundaries of an HTML file, are required in all your documents.

## Create a Title

To create a title for your document, type the title text between the title tags:

```
<TITLE>Test Document Number 1</TITLE>
```

## Add a Heading

The "body" tags define the boundaries of your document's contents. The remainder of your document, including the main heading, should be created between the starting and ending body tags.

Within the body tag pair, type "My First HTML Page" (omit the quotes). It should look something like this:

```
<BODY>  
My First HTML Page  
</BODY>
```

To make the text into a heading, first select it with the mouse by pressing the left mouse button and dragging the cursor across the text. The text will be highlighted to indicate that it has been properly selected.

Next, use the left mouse button to 'click' on the small button with the number "1" on it in the "Heading" section of the tool bar.

The heading text will be bracketed by "Heading 1" tags as follows:

```
<H1>My First HTML Page</H1>
```

That's all there is to it.

If you select the wrong tool and erroneous HTML tags are inserted, click the "Undo" button on the document window.

## Testing your Work

You can test your work-in-progress at any time. "Testing" means viewing your work with a browser so that you can see how it will look to people in your intended audience when they retrieve it.

In order to display your file, the test browser loads your HTML document from a disk file. Therefore, you must always save the file before testing it for the first time. For the same reason, before testing again, you need to save any changes that have been made to the text.

To save your new HTML document, click the "Save" button on the document window. If the document has not yet been saved, you will be prompted for a file name. Select a path name from the list boxes and enter a filename like "Test1.htm". The suffix, ".htm" is chosen to let *HTML Assistant Pro* (and you) know that this is an HTML file.

After saving the file for the first time, click the "Test" button on the document window. If you have followed the steps outlined above to select your test browser, in a few seconds the browser will be loaded and your file will be displayed.

You should see the title and the properly formatted main heading.

When you have finished viewing your work, you can bring your editing window to the front again, make whatever changes you require, then again test the changes in your browser. If your browser is Netscape V1.1 or later, to return to the editor simply click the small "To Editor" button in the lower right corner of the screen. Netscape will remain open, waiting for your next "Test". This makes editing and testing with Netscape almost seamless. You can repeat this process as often as you like while you are working.

If you have changed the document between tests, you will be reminded to save it again unless you have already done so (by clicking the save button). After the file has been saved once, it will be resaved with the same name whenever the "Save" button is clicked.

*HTML Assistant Pro* can automatically resave your file for you whenever you select the "Test" button. To use this feature, simply select the "Autosave file before test" item from the "Options" menu.

## Possible Source of Confusion While Testing

### Updated Contents of a File are not Displayed

Sometimes, when retesting the same file, the browser will not display the updated contents. Instead, it redisplay a previously tested version.

This usually happens when the browser is not closed and reloaded between tests -- i.e., when using the editor's *Dynamic Data Exchange* feature, or when loading and reloading the file from a browser menu.

This occurs because most browsers use a technique called "caching" to avoid accessing the same information more than once. When caching is used, if a file on a remote machine, or your own, has already been loaded by the browser, it will redisplay the copy of the file that is *already in its memory*. That is, *it will not reload the file*.

This usually works to your advantage. If a large file has already been downloaded from a remote server, the browser doesn't need to take up either your time or Internet bandwidth to retrieve it again. However, when testing and retesting your own files this can be a problem.

Presently, most browsers are equipped with a feature that will permit you to manually cause them to retrieve the file again. In the future, browsers will probably offer an option to turn caching off.

Whenever a browser used for testing fails to display the changes you have made to your file, *use the browser's file reloading option*.



## Adding Text

Next, add some text to your document. Type a paragraph, starting on a line following the heading. When you are finished, position the cursor at the end of your paragraph and click on the large tool bar button with the "Paragraph" symbol on it. The paragraph symbol looks like a backward "P". This will cause a "<P>" tag to be inserted at the end of your text. You need to do this at the end of every paragraph.

Type another paragraph and add its "<P>" tag.

To speed things up you can add the "<P>" tag by entering the shift-Enter key combination. That is, press the "Shift" key and hold it down just before you press the "Enter" key -- as if you were typing an upper case letter.

At this point, your document should look something like this:

```
<HTML>
<HEAD>
<TITLE>Test Document Number 1</TITLE>
</HEAD>
<BODY>
<H1>My First HTML Page</H1>
```

This is the first text paragraph. It can be as long as you like. Be sure to enter the paragraph tag.

```
<P>
```

This is the second paragraph. Remember, white space and blank lines are irrelevant as far as your browser is concerned. Extra lines will be ignored. That is why you need to tag the end of each paragraph. You can even run all your paragraphs together. As long as a paragraph tag is inserted between them, they will be displayed properly. Note that the paragraph tag can be on its own line, or on the last line of the text. You can even skip lines if you like.<P>

```
</BODY>
</HTML>
```

# Creating Lists

*HTML Assistant Pro* makes it easy to create formatted lists with HTML.

Before you begin this section you might want to review the section on lists in the "Beginners Guide to HTML".

To create a list, add a few more lines of text as shown below. Press the "Enter" key after each one. Your text should look like this:

```
List item one  
List item two  
List item three
```

Next, use the mouse to select all three lines. Be sure that no empty lines are selected.

After selecting the lines that are to be formatted as a list, click on the small button on the left half of the tool bar that has the numbers 1,2 and 3 and some lines on it. This will format the selected text as a "numbered" or "ordered" list.

The result will look like this:

```
<OL>  
<LI> List item one  
<LI> List item two  
<LI> List item three  
</OL>
```

Next, retype the same three lines (without the HTML markings) immediately following the numbered list. Select the text and click the toolbar button located just below the one you used to create the ordered list. This button will create an "unnumbered" or "bulleted" list.

The resulting HTML code is:

```
<UL>  
<LI> List item one  
<LI> List item two  
<LI> List item three  
</UL>
```

Test your code with a browser to see how the two types of lists are displayed.

## Linking to Another HTML Document

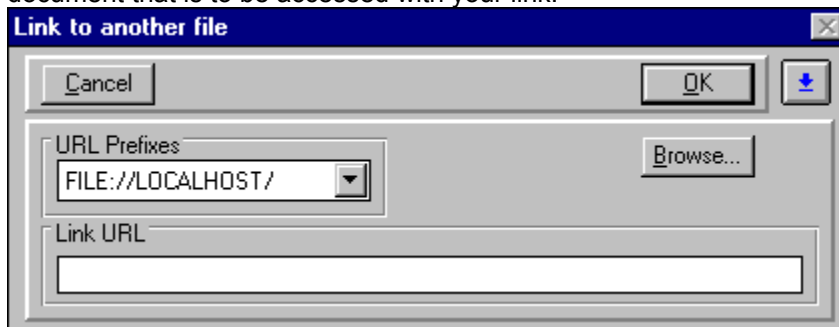
Next, let us create a link to an HTML document at NCSA in Chicago, home base for the *Mosaic* browser.

The link has two components. One is the actual pointer to the document, called a "URL", an abbreviation for "Uniform Resource Locator". The other is the text that users can "click" when they want to retrieve and display the document.

At this point, you might want to refer to the "Beginners Guide To HTML" for more information about URLs.

To set up a link, start by entering the "clickable" text on an empty line. Type, "NCSA Home Page". Then, using the mouse, select the text you just typed.

Next, click the "Link" button on the main toolbar. A dialog box will appear with several controls on it. In the lower half of the dialog box is a text box, labeled "URL Text", where you can enter the URL for the document that is to be accessed with your link.



For purposes of this exercise, we want to start from scratch, so delete whatever text may be in the text box.

Next, select the URL prefix. Click on the down arrow on the right hand side of the control labeled "URL prefix". A list of prefixes will drop into view (the control is called a "drop down list box"). Select the item, "http://", and the text will appear in the URL text box below.

The "http://" prefix tells the browser to retrieve a document from a World Wide Web (or "HTTP") server. ("HTTP" stands for "Hypertext Transport Protocol.")

Now, add the location of the document to the URL by entering its Internet address and file path. For the Internet address type: "www.ncsa.uiuc.edu" (omit the quotation marks).

The partially completed URL should look like this:

```
http://www.ncsa.uiuc.edu
```

Finally, add the file path and name, "/SDG/Software/Mosaic/NCSAMosaicHome.html" (without the quotes).

Note that *case is important*. Unlike MS-DOS and Windows, file and directory names on the Internet *are* case sensitive.

Type carefully.

"Software" is *not* the same as "software"!

The complete URL is:

```
http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/NCSAMosaicHome.html
```

When the URL entry is finished, click the "OK" button to return to your document.

Note that *HTML Assistant Pro* has created the following HTML code for you:

```
<A  
HREF="http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/NCSAMosaicHome.html">NCSA  
Home Page</A>
```

(Note: Ignore any line breaks shown here in the HTML code.)

Insert a paragraph tag at the end of the line. This will ensure that it will be separated from additional lines of text that may be added later.

Your page should now look just about like this:

```
<HTML>  
<HEAD>  
<TITLE>Test Document Number 1</TITLE>  
</HEAD>  
<BODY>  
<H1>My First HTML Page</H1>  
This is the first text paragraph. It can be as long as you like. Be sure to  
enter the paragraph tag.  
<P>  
This is the second paragraph. Remember, white space and blank lines are  
irrelevant as far as your browser is concerned. Extra lines will be ignored.  
That is why you need to tag the end of each paragraph. You can even run all  
your paragraphs together. As long as a paragraph tag is inserted between  
them, they will be displayed properly. Note that the paragraph tag can be on  
its own line, or on the last line of the text. You can even skip lines if  
you like.<P>  
<A  
HREF="http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/NCSAMosaicHome.html">NCSA  
Home Page</A><P>  
</BODY>  
</HTML>
```

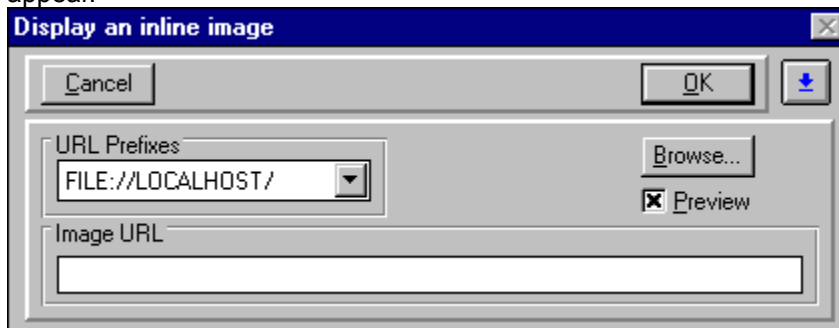
This would be a good time to test your file to see what it looks like when displayed by a browser. If you are connected to the Internet, and the URL for the NCSA home page *is still valid*, clicking anywhere on the text, "NCSA Home Page", should result in the display of the NCSA Home Page by your browser. Don't worry if it doesn't work. There are many reasons why a properly formatted link may fail.

## Displaying an In-line Image

The following sections, about displaying images, require that you have an image file in Graphic Interchange Format (GIF) on your local disk drive. GIF files have the suffix, ".gif". (How to obtain or create "\*.gif" image files is beyond the scope of this text.)

To cause a GIF file on your hard drive to be displayed as an *in-line* image, start by positioning the text cursor on the next empty line in the "body" section of your new HTML document.

Next, click on the "Image" button near the center of the *HTML Assistant Pro* tool bar. A dialog box will appear.



Delete whatever text is in the "Image URL" textbox.

Use the down arrow on the "URL Prefix" drop down list box to view the available prefixes. Select the prefix, "file://localhost/" by clicking on it, and the text will appear in the URL text box below. The cursor should be positioned immediately following the URL prefix, without any spaces.

The "file://localhost/" prefix tells the browser to retrieve a file from your local disk drive. Next, add the location of the GIF image you want to display by clicking the "Browse..." button near the bottom of the dialog box. When the file dialog box appears, use the list boxes to select the image file you want to display. When you have selected your file, click the "OK" button.

If the "Preview" check box is selected, a preview window is opened to let you view the image you have selected. Click OK in this window to accept the image or Browse to select another image.

When you have selected an image, the file name will be appended to the URL prefix already in the URL textbox.

The full URL should look something like this:

```
file://localhost/C:/graphics/my_pic.gif
```

Notice that the slashes in the DOS file path are *forward* slashes rather than back slashes. That is because most browsers prefer forward slashes. If you want to use the familiar MS-DOS back slashes, deselect the default, "Display forward slashes..." option, by clicking on its check box.

When you are satisfied that the URL is correctly formatted, click the "OK" button near the top right of the image dialog box.

The following HTML code will appear at the cursor in your file:

```
<img src= "file://localhost/C:/graphics/my_pic.gif">
```

Notice that there is *no* closing "</img>" tag.

Try typing text after the image code, with and without inserting a </P> tag in front of it (remember, the text

can be on the next line in the editor -- as long as there is no "<P>" tag, it is displayed on the same line as the image), and use the "Test" button to view the result.

If you open the Image dialog again, you'll notice a button with an arrow on it. Clicking this button causes additional options to be displayed which can be used to specify how the image is formatted. For example, you might want to experiment with the width and height options or alignment option. Of course, for the alignment property to be meaningful, the image must not be so wide that there is no room for the text.

In case you are wondering, the text box in the image dialog, labeled "Alternate text", is for entering text to be displayed by browsers that are unable to display images -- i.e., "text only" browsers. Bring up the image dialog, type some text there (a short description of the image, for example), and see what happens to the HTML code. The text will have no effect with browsers, like Netscape or Internet Explorer, which can display in-line images.

## Displaying an Image with a Viewer

To display an image on your local drive with a viewer, use the "Link" button on the tool bar. You may want to refer to the instructions for "Linking to another HTML document", above.

Proceed as follows:

Before selecting the "Link" button, enter the "clickable" text, i.e., the text the user will click on to cause the image to be displayed. For example, you might use, "See the Good ship Lollipop".

Select the text with the mouse. Then, click on the "Link" button to bring up the link dialog.

In the link dialog, from the "URL Prefix" drop-down listbox select the "file://localhost/" prefix.

Use the "Browse..." button to select your image file.

After clicking the "OK" button on the link dialog, your HTML code should look like this:

```
<A HREF="file://localhost/C:/sailboat.gif">See the Good ship Lollipop</A>
```

Remember that in order to view in-line images, you *must* setup your browser to use the appropriate viewer. See the instructions that came with the browser software for information on how to accomplish this.

When you test your file and click on the text "See the Good ship Lollipop", the "GIF" viewer you selected for your browser will be loaded and it will display the image in its own window.

## Going Further

If you have read the "Beginners Guide to HTML" and followed this tutorial, you already have enough information to create simple HTML documents which include headings, plain text, lists, images, forms, tables, and active links to other information.

This document does not cover all the available features of HTML. For more complete information, we suggest you check your local bookseller for a recent book that includes information on the latest HTML features.

Moreover, the most current information is available on the World Wide Web. You can use one of the search options available with your browser to find it. Start your search with one of the following terms (or something similar): "HTML", "HTML Authoring", "WWW Authoring".

You should also look at the HTML code for documents you see which have features that you don't yet know how to implement. You can use your browser's "View source" (or equivalent) option to accomplish this.



## Overview

Keeping an accurate record of Internet journeys is not a trivial task. In fact, one of the major criticisms of the Internet is that information is not well organized and it is difficult to retrace your steps if you want to repeat a search. Many (if not most) Internet users have found themselves using computers to scan the Internet for information while keeping a pencil and paper handy so that they could make notes about the places they had been.

Some Gopher and WWW browser packages permit saving place markers by using 'bookmarks' or a 'hot list' to collect address information. As useful as this is, after several searches or browse sessions these collections quickly become a *pot pourri* of tags, markers and pointers to a varied assortment of Internet places that have no particular relationship to each other.

By making it easy to create your own browser pages, *HTML Assistant Pro* provides a way to organize and keep track of your Internet journeys.

## Collecting URLs

World Wide Web browsers use Uniform Resource Locators (URLs) to find and access all forms of information on the Internet -- including text files, news groups, mail lists, gopher directories, HTML documents, program files, images, sounds, movies, etc.

To use *HTML Assistant Pro* as an aid to organizing your WWW experience, you first need to begin collecting the URLs that point to the WWW sites that interest you.

Most browsers permit transferring the URLs which they use for Internet access to the Windows clipboard. Check the instructions that came with your browser to find out how to do this. From the Clipboard the URLs can easily be copied to any Windows text editor.

Once the URL text is in the clipboard it can be 'Pasted' into a text file using a simple text editor like MS Notepad (supplied with Windows) or *HTML Assistant Pro*. Be sure that there is an End-of-Line following the URL. An End-of-Line is a carriage-return/linefeed pair which is added when you press the 'Enter' key after inserting text. That is all that is required to begin collecting URL links for use by *HTML Assistant Pro*.

Alternatively, as you explore with a browser, you can save URLs by creating 'bookmarks'. *HTML Assistant Pro's* URL file editor will permit you to convert your browser bookmark files to HTML Assistant URL files. In this way, HTML Assistant permits you to create URL files that combine information collected with several browsers.

Files containing URLs should be saved with the suffix, ".URL". While this isn't absolutely necessary, it does make it easier for you (and *HTML Assistant Pro*) to identify them.

## Editing and Organizing URLs

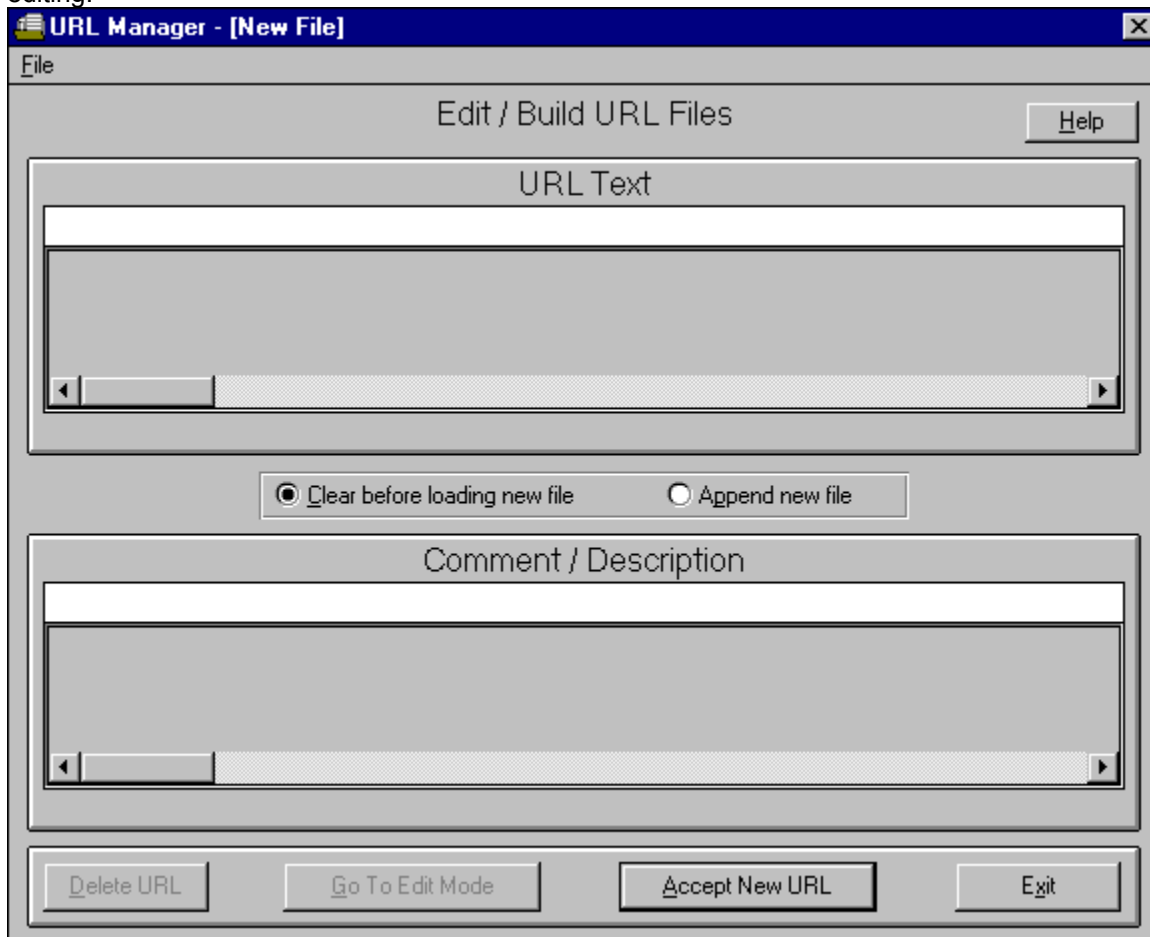
Once you have collected URLs that point the way to the places you have visited, the next step is to select and organize them into separate files that suit your particular needs. For example you might want to create a file containing URLs that can help you learn about Internet resources, or interesting art exhibits, or information on animal breeding.

*HTML Assistant Pro* provides two ways to do this. Use whichever works best for you.

The most familiar way for most people will be to open a new edit window (by selecting 'New' from the 'File' menu) for each subject and copy/paste selected URLs (and their optional comments) from your general URL collection or other URL files to the new one.

The second method involves using the URL file edit popup window that can be viewed by selecting 'Edit/Build URL Files' from the 'URL' menu.

This permits you to combine commented URLs from different sources and to use scrolling lists for editing.



## Building Browser Pages

The URLs you have collected are pointers to the places you have visited on your Internet journeys. The final step in organizing your personal Internet experience is to incorporate selected URLs into HTML documents to be used as WWW browser pages. These pages will be the tour guides for yourself and others who want to retrace your steps. With *HTML Assistant Pro* you can quickly create browser pages with titles, headings and descriptive text for the displayed URL links.

*HTML Assistant Pro's* Automatic Page Creator makes this part of the job very easy.

It guides you through the process of creating a title, headings and descriptive text for your page, and enables you to incorporate a list of URLs (from a file of your choice) which will become the links to the places you want to visit. With the 'click' of a button your new page is automatically created and displayed in an edit window ready for saving as an HTML document. Your new page can then be tested with a WWW browser of your choice without leaving *HTML Assistant Pro*.

If you wish, the Automatic Page Creator will append additional descriptive material and/or URL links to the newly created page. In this way you can allow *HTML Assistant Pro* automatically to build pages with any number of different formats and degrees of complexity. Of course, the resultant page can be edited using any of the other tools provided with *HTML Assistant Pro*.

Using your local files with HTML

# Using Local Files with HTML

As you create your own HTML files with links to remote sites, you may also want to create other browser pages consisting of links to these local files. By doing this with successive generations of locally created files you can create a hierarchical system of 'menu pages' for organizing and indexing your work.

For example, you may have built separate files (with remote links) concerning the following subjects:

Using HTML

WWW and related FAQs

Selected sites via CERN

You might want to create a menu file concerned with general WWW topics that includes the above files as selectable links.

## Creating a Menu Page

You can build such a file one step at a time, by opening a new edit window and creating a local file link for each file using the 'Link' button. When the link popup appears, select the "file://localhost/" prefix from the prefix listbox.

Next, enter the name of a file, or use the 'Browse' button to select a file name from a file select dialog box.

Clicking the 'OK' button (on the link popup) will cause *HTML Assistant Pro* to create (and insert into the edit window) the URL for a link to the file you selected. It will look something like:

```
file://localhost/C:/html/myfile.htm
```

You can continue in this manner to add additional URL links. Using *HTML Assistant Pro*'s list formatting capability, you can select and format the file links as one or more lists (numbered or unnumbered).

When the process of adding and formatting file links is finished you can complete your menu page by using *HTML Assistant Pro*'s tools to add a title, headers and explanatory text to your display.

## Another way to create a Menu Page

Another way to create a menu page for your local files is to create a URL file and use that file with the Automatic Page Creator to build your menu display page.

*HTML Assistant Pro* will insert a URL in the active edit window if you select the 'Create Local File Reference...' option from the 'Command' menu and select a file from the file select dialog that appears. You can add the link display text (the 'clickable' text) as a comment after the URL has been displayed (insert a comma following the URL before you enter the comment).

You may also create a URL file by selecting the 'Edit/Build HTML Assistant URL Files...' option from the 'URL' menu.

Once the URL file has been created and saved, you can open the Automatic Page Creator and use it to create your HTML display file.

## Overview

Note: Not all menus or menu options are visible upon startup. The menu bar, and the contents of most menus, changes when at least one document window is open. To view all the menus, if they are not already visible, either open a file or a new document.

*HTML Assistant Pro's* menu's may seem formidable at first. There are a number them, and each one presents many options.

While this bewildering array of choices can be confusing, once you become familiar with the menu structure, you will learn that they present many opportunities for you to simplify and speed up the creation of your HTML documents.

Many of the options, such as those in the "File" and "Edit" menus, will already be familiar to you because they are found in commonly used word processors and text editing software.

The fastest way to learn to use any of the menu options is to try them. Just be sure that you have another, saved copy of the document with which you are experimenting.

Most of this section will be devoted to explaining how to use the menu features that are unique to the creation of HTML documents with *HTML Assistant Pro*. Menu items that are self-explanatory or are common in many applications are not described here.

# The "File" Menu

Open

Insert Start Template in New Document

Import Word Processor (RTF) File

Automatic File Save

Multiple File Replace

Text Search

File Search

Insert File at Cursor

Set Test Browser

## Open

This option permits you to open an existing document file. When the file is opened, a backup copy will automatically be saved in *HTML Assistant Pro's* backup subdirectory, "proback", which resides in the *HTML Assistant Pro* program directory. The backup file will have the same name as the opened file, but the suffix will be changed to ".BAK".

To conserve disk space, the backup file will be *replaced* each time a document with the same file name is opened.

**This means that if, for any reason, you want to use the previous (i.e., backup) version of your file, you must either rename or copy the ".BAK" file to another directory before reopening the existing file!**



## Insert Start Template in New Document

If this option is checked (clicking on it will toggle the check mark) the section bounding tags (shown below) which are required in most HTML documents will be automatically inserted whenever a new document is started.

```
<HTML>
<HEAD>
<TITLE>Created With HTML Assistant Pro - 7/8/96</TITLE>
</HEAD>
<BODY>
</BODY>
</HTML>
```

This is equivalent to manually selecting the "Display standard document template" option in the "Command" menu.

## **Import Word Processor (RTF) File**

Selecting this option permits you to import and automatically convert formatted word processor text to HTML. Before importing it, the text must be saved as "Rich Text Format", an option that is provided by most Windows word processors.

## Automatic File Save

Selecting this option will reveal a sub-menu with two items. The first, "Automatically save when closing", when checked, will cause any work you have done to be automatically saved when the program is closed. If it is *not* checked, and if text has been modified, upon closing the file, a dialog box will be presented asking if you want to save the file.

The second option, "Automatically save every five minutes", will cause any files that have been modified between saves to be resaved automatically within five minutes.

Automatic file saving prevents excessive loss of work in the event of an abnormal program termination - due to a system crash or power failure, for example. Should abnormal termination occur, the next time the program is started, it will notify you so that you may retrieve any automatically saved work. *It is highly recommended that you work with this option selected* -- it is selected by default when HTML Assistant Pro is installed.

Files that already have file names will be saved in their existing locations, with their existing file names - that is, they will replace the files in the same way they would if they had been saved manually. New files that have not yet been manually saved will be given temporary names and saved with the suffix ".SAV" in *HTML Assistant Pro's* backup subdirectory, "proback", which resides in the *HTML Assistant Pro* program directory.

## Multiple File Replace

Selecting this option will bring up the "Multiple file replace" dialog window. (NOTE: this option is also available in the "Edit" menu.)

You can enter the text to be searched for, and the replacement text in the text boxes at the top of the dialog. You have the option of selecting a case sensitive search and/or searching for whole words only.

The left side of the dialog window represents the files on your disk drive(s). Using the list boxes, you can select the drive, subdirectory and file types (i.e., \*.htm, \*.txt, etc.) you wish to select from. The right side of the dialog has a list box which will hold the names of the files to be searched. File names may be quickly and easily transferred from your disk drive to the search list.

Clicking on a file name in the file list box (on the left side of the dialog near the bottom) will cause it to be selected. Double-clicking on the filename will cause it to be copied to the list of files to be searched.

You may select more than one file name to be copied by pressing the <Ctrl> key when clicking. Dragging the mouse across several file names (with the left button pressed) will also result in multiple selections.

Clicking on the "Transfer all" button will cause all files in the file list (which means all files in the selected subdirectory) to be copied. Clicking on the button with the arrow on it, just below the file list box, will also cause any selected file names to be copied to the search list.

Once the files to be searched have been added to the search list, and the search and replace text has been entered, clicking on the "Replace" button, in the upper right corner of the dialog, will cause every instance of the search text, in each file on the search list, to be replaced.

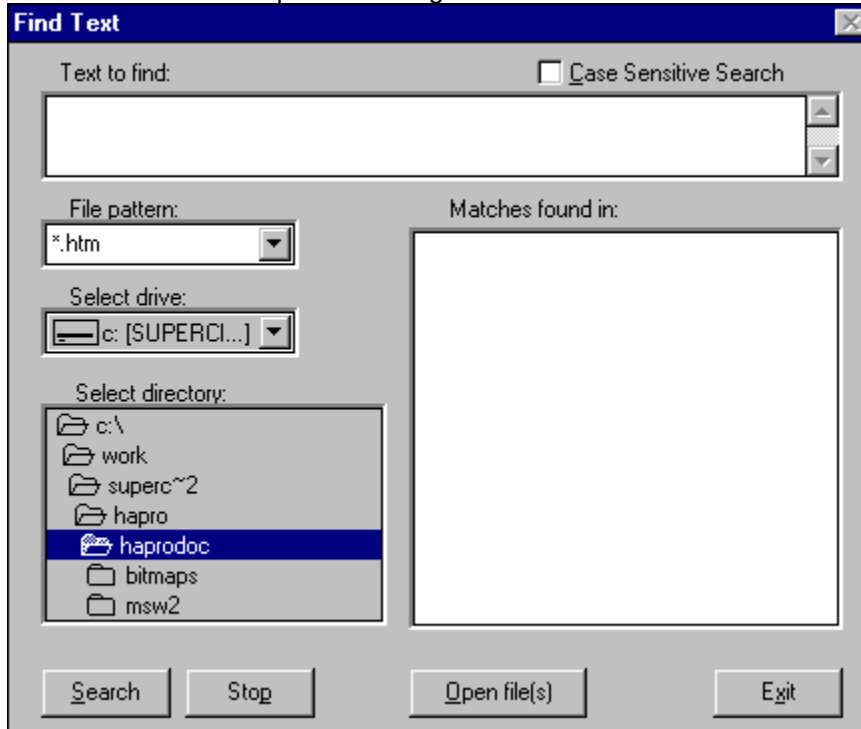
Backup files, if you have selected the "Create backups" option, will be created for each file. They will be in the same directory as the original file, and will have the suffix, ".BAK". When a backup file is created, it replaces any existing backup file with the same name and directory location.

You may halt the search at any time by using the "Stop search" button, or pressing the <Esc> key.

## Text Search

The "Text search..." option on the "File" menu permits you to search a *single* subdirectory (*not* the full subdirectory tree), for files containing a user selectable text string.

When this option is chosen, the "Text Search" dialog window will appear. Type the text to be searched for in the text box at the top of the dialog window.



Use the "File Pattern" combo box to let the search routines know what files to search. This works the same way that any MS-DOS or Windows directory listing file pattern works.

That is, use "\*" to search all files in the directory. To search all HTML files, use "\*.htm". To search all files starting with "K", enter "K\*", etc.

Use the "Select drive" and "Select directory" listboxes to specify the subdirectory that contains the files to be searched

Click the "Search" button to begin the search. You may interrupt the search at any time by clicking the "Stop" button.

Full path names of all the files containing the specified text will be inserted into the listbox on the right side of the dialog window.

If you want to edit or view one or more of the files that have been found, simply click on the file name in the listbox. You may select more than one file by pressing the control key (usually marked "Ctrl") while clicking additional file names. To select a continuous section of the list, click on the first file, then, while holding down a "Shift" key, select the last one in the section you are interested in.

Use the "Open file(s)" button to edit or view the selected files.

Use the "Exit" button to end the search without viewing files.

**Note on viewing multiple files**

You *are not* required to select *all* the files you want to view, before you view any one of them. That is, if you want to look at several files without opening too many windows, you can select just one before you click the "Open file(s)" button.

The next time you use the "Text search..." option from the "File" menu, the "Text Search" dialog will open *with the previous list already in place*, and ready for you to select other files as you need them.

You can do this is many times as you like.

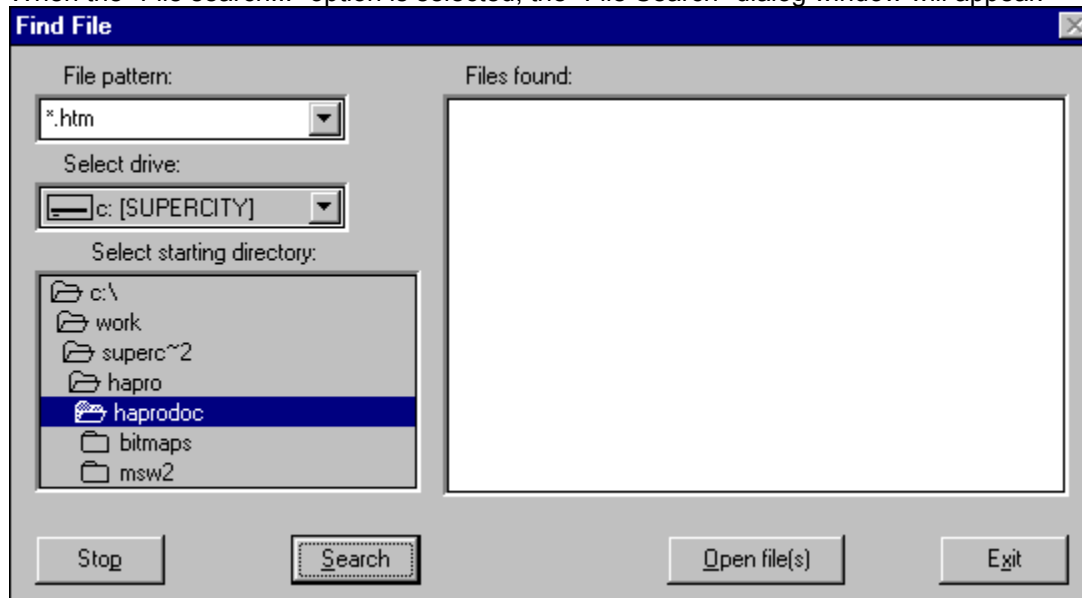
However, in order for the file list to be "remembered", *you must select and open at least one file!*

# File Search

The "File search..." option on the "File" menu permits you to search a *full subdirectory tree*, for files (rather than text within files) with a specified file name pattern.

This option is useful, for example, if you want to locate all HTML files anywhere on your "C" disk drive.

When the "File search..." option is selected, the "File Search" dialog window will appear.



Use the "File Pattern" combo box to let the search routines know what filename pattern to search for. This works the same way that any MS-DOS or Windows directory listing file pattern works.

That is, use "\*" to list all files in the directory. To search for all HTML files, use "\*.htm", etc.

Use the "Select drive" and "Select directory" list boxes to specify the subdirectory that is to be the outermost branch of the tree which is to be searched.

The routines will search that entire subdirectory and all its branches for files of the type specified.

For example, to search your entire "C" drive for all HTML files starting with "K", you would enter "k\*.htm" (file patterns for MS-DOS or Windows are *not* case sensitive) in the "File Pattern" combo box. Next, using the drive selection listbox, select the "C:" drive. Finally, use the directory list box to select the root directory, i.e., "C:\".

Click the "Search" button to begin the search. You may interrupt the search at any time by clicking the "Stop" button.

Full path names of all the files having the selected file specification will be inserted into the listbox on the right side of the dialog window.

If you want to edit or view one or more of the files that have been found, simply click on the file name in the listbox. You may select more than one file by pressing the control key (usually marked "Ctrl") while clicking additional file names. To select a continuous section of the list, click on the first file, then, while holding down a "Shift" key, select the last one in the section you are interested in.

Use the "Open file(s)" button to edit or view the selected files.

Use the "Exit" button to end the search without viewing files.

**Note on viewing multiple files**

You *are not* required to select *all* the files you want to view, before you view any one of them. That is, if you want to look at several files without opening too many windows, you can select just one before you click the "Open file(s)" button.

The next time you use the "File search..." option from the "File" menu, the "File Search" dialog will open *with the previous list already in place*, and ready for you to select other files as you need them.

You can do this is many times as you like.

However, in order for the file list to be "remembered", *you must select and open at least one file!*



## **Insert File at Cursor**

The "Insert file at cursor..." option in the "File" menu permits you to load and insert another file into your document at the cursor position.

This is most useful for inserting "boilerplate" text that appears in several documents. For example, you might want to insert information about how to contact you by Email, into a series of HTML pages.

When you select this option, a file select dialog box will appear. Simply select the file you want to insert and click the "OK" button to insert the file.

## Set Test Browser

If, as you are editing a document, you want to view or "test" your work with a World Wide Web browser without leaving *HTML Assistant Pro*, you need to let the software know where your test browser can be found.

You can tell HTML Assistant Pro which browser to use by selecting the option, "Set test program name...", from the "File" menu. The dialog boxes will lead you through the process. Using the appropriate list boxes, select the path and filename for your browser. If you want the software to "remember" where the browser can be found, so that you don't have to repeat the process each time you start the program, select the "Yes" button when asked if you want to save it as a "permanent test program name".

# The "Edit" Menu

Multiple File Replace

Undo/Redo

## Multiple File Replace

Selecting this option will bring up the "Multiple file replace" dialog window. (NOTE: this option is also available in the "File" menu.)

You can enter the text to be searched for, and the replacement text in the text boxes at the top of the dialog. You have the option of selecting a case sensitive search and/or searching for whole words only.

The left side of the dialog window represents the files on your disk drive(s). Using the list boxes, you can select the drive, subdirectory and file types (i.e., \*.htm, \*.txt, etc.) you wish to select from. The right side of the dialog has a list box which will hold the names of the files to be searched. File names may be quickly and easily transferred from your disk drive to the search list.

Clicking on a file name in the file list box (on the left side of the dialog near the bottom) will cause it to be selected. Double-clicking on the filename will cause it to be copied to the list of files to be searched.

You may select more than one file name to be copied by pressing the <Ctrl> key when clicking. Dragging the mouse across several file names (with the left button pressed) will also result in multiple selections.

Clicking on the "Transfer all" button will cause all files in the file list (which means all files in the selected subdirectory) to be copied. Clicking on the button with the arrow on it, just below the file list box, will also cause any selected file names to be copied to the search list.

Once the files to be searched have been added to the search list, and the search and replace text has been entered, clicking on the "Replace" button, in the upper right corner of the dialog, will cause every instance of the search text, in each file on the search list, to be replaced.

Backup files, if you have selected the "Create backups" option, will be created for each file. They will be in the same directory as the original file, and will have the suffix, ".BAK". When a backup file is created, it replaces any existing backup file with the same name and directory location.

You may halt the search at any time by using the "Stop search" button, or pressing the <Esc> key.

## Undo/Redo

*HTML Assistant Pro* offers 255 levels of undo and redo. This means that the editor will remember up to 255 edits to a document which can be "undone" by the user during a given editing session. This is useful to allow the user to make changes to a document to test alternative designs and be able to restore the document to its original form.

# The "Options" Menu

Items in the "Options" menu are generally concerned with the behavior of general program features that affect the users interaction with the software.

Options settings are "remembered" by the program between sessions, so you needn't reset them each time you run *HTML Assistant Pro*.

Save Text to UNIX File

Use DDE with ... [Browser Name]

Autosave File Before Test

Configure Paragraph Button

Autoconvert UNIX Files to DOS

Use Lower Case HTML

Default Font

Show ISO Characters List

Edit User Tools

Select Default User Tools Directory

Hide Tag Tools Palette

## **Save Text to UNIX File**

Files that can be read by editors and other software that runs under the UNIX operating system have different end-of-paragraph characters than MS Windows files do. Some UNIX editors cannot read MS Windows files correctly. This option permits you to save your files in a form that can be read by UNIX software.

## Use DDE with ... [Browser Name]

If you are using Netscape or Microsoft Internet Explorer as your test browser, *HTML Assistant Pro* gives you the option of using DDE (Dynamic Data Exchange) to send the browser the name of the HTML file you want to test. If the browser is not loaded, *HTML Assistant Pro* loads it. Once your browser is loaded and you do not close it, it will not have to be reopened whenever you want to test a file. Each time you select the item in the File menu for testing using DDE, or the "Test" button on a document window, the name of the file you are testing is sent to the browser and it becomes the active, topmost application.

When using DDE, your browser may not always refresh itself if your file has changed. That is, changes in your HTML text will not always be immediately seen when you invoke the browser. If that happens use the browsers "Reload document" (or equivalent) option to ensure that you are viewing the latest saved version of your HTML file.

Clicking on the small '*To editor*' box in the lower right corner of the browser's window will return you to *HTML Assistant Pro* (without closing the browser).



## **Autosave File Before Test**

Selecting the "Autosave file before test" item in the "Options" menu will cause any changes in your file to be saved automatically before it is tested. In order for you to give the file a name, new files must be saved manually before they can be tested the first time.

## Configure Paragraph Button

Selecting the "Configure paragraph button" option from the "Options" menu will display a sub-menu with several choices involving the placement of "newlines" in the text whenever the button that inserts a new paragraph tag is clicked.

This option affects only the way your text is displayed in the editor. It has no effect on its appearance when displayed by a browser.

The submenu will also permit you to select from various options involving bracketing text with paragraph tags and use of an "align" attribute. These options can, with some browsers, affect the appearance of your displayed text.

The best way to find out what these do is to experiment and try them with different browsers.

## **Autoconvert UNIX Files to DOS**

Text files created by UNIX systems, as many WWW HTML files have been, are not normally displayed in an easily readable form by most Windows or DOS based programs.

Selecting "Autoconvert UNIX files to DOS" in the "Options" menu will cause a file in UNIX format to be automatically converted to MS-DOS format text as it is loaded into a document window. A message box will popup to let you know when this happens. The original file is not affected, unless, of course, you resave the file with the same name. In that case, it will be saved in MS-DOS format.

Text that is already displayed in document windows may be converted to and from UNIX format, by using options in the "Filters" menu.

## **Use Lower Case HTML**

If this option is checked (i.e., selected) then all text in HTML markings will be displayed as lower case. Otherwise, they will be displayed as upper case.

HTML is not case sensitive, so choose lower case display if it suits you.

## **Default Font**

This option permits selection of the font and font characteristics (i.e., bold, size, etc.) which will be used as the default for all new text windows.

The font for text in an individual window that is already displayed, can be changed by using the "Font (current window only)" option in the "Edit" menu.

## Show ISO Characters List

Selecting and deselecting this option will toggle the display of a small window that permits selection of certain of characters not available on most North American keyboards. These include things like accented letters, umlauts, etc.

When a character is selected from the list displayed in the window, its HTML equivalent is inserted into the text at the cursor.

## **Edit User Tools**

Selecting this option will open the "Edit User Tools" dialog. This permits changes to the current "User Tools" list displayed in the User Tools dialog.

Clicking with the right mouse button on the main tool bar's "User Tools" button will also open the "Edit User Tools" dialog.

## Select Default User Tools Directory

When the program runs the first time, it uses the *HTML Assistant Pro* program directory as the location for automatically saving and retrieving the active User Tools file, called "default.ut." This option permits you to change the default User Tools directory.



## Hide Tag Tools Palette

Selecting and deselecting this option toggles the display of the floating tool palette that permits quick insertion of common HTML tags.

The small button, with a "T" on it, at the right hand end of the main tool bar has the same function.

# The "Filters" Menu

The filters menu is generally used for operations that transform the contents of the currently selected document window in some way.

These are all "Undoable". That is, click the "Undo" button after trying one of these options, and the text will be restored to its original state.

The best way to find out exactly what they do is to try them.

[Convert UNIX Text to DOS Text](#)

[Convert ISO ASCII Codes to HTML](#)

[Strip HTML](#)

[Reformat Paragraphs](#)

[Strip Extra Newlines](#)

[Extract & Display Link URLs](#)

## Convert UNIX Text to DOS Text

UNIX and DOS text files handle end-of-line characters differently. That is why, when a UNIX format file is loaded into *HTML Assistant Pro* (or *Notepad*, for that matter) it looks like one long paragraph, interspersed with strange rectangular bars. In this form, UNIX files are very difficult to read or to edit.

Using this option on the "Filters" menu will transform that unreadable UNIX text to DOS text so that you can work with it.

You can use the "Convert DOS text to UNIX" option if you want to resave your edited work in UNIX format.

## Convert ISO ASCII Codes to HTML

Some keyboards permit characters that are not available as part of the North American ASCII character set to be typed and entered into files. Accented characters, umlauts, and many others, fall into this category.

This filter will convert these characters to HTML.

## **Strip HTML**

Selecting this option will cause all the HTML code to be removed from the text in the current document window.

## Reformat Paragraphs

Using this option causes all newline characters in *every* paragraph in the current document window to be removed, *except* those found in empty lines.

This can make some documents, from which the HTML code has been removed, easier to use with a word processor for further reformatting.

## **Strip Extra Newlines**

Use of this option will remove extra blank lines, often left after HTML code has been removed from text. This affects all the text in the current document window.

## Extract & Display Link URLs

This option extracts, and displays, only the URLs and the live, clickable text found in "<A HREF..." links in the current document window.

This is useful, for example, if you want to work with all the links in a file, without being bothered by any of the other text.

Be aware though, that some of the links extracted may be "relative" and are not usable from a file that is accessed from any other disk drive than the one that is on the server from which the HTML file was retrieved. These should either be edited so that they are proper links, that is, have complete URLs including a server address and full file path, or deleted.

The URLs and associated "clickable" text are listed in "HTML Assistant" URL format. The list, as is, or edited, can be saved as an HTML Assistant URL file, using the ".url" suffix.

You can then use *HTML Assistant Pro's* Automatic Page Creator to very quickly create an HTML document that can be easily perused for links that are of interest.



## The "HTML" Menu

Selecting an option from the upper part of this menu will cause the HTML code, as displayed, to be inserted at the cursor. Most of these items are duplicated on *HTML Assistant Pro's* main window tool bar and floating tag palette. They can be handy when the tag palette is not visible.

The characters listed on the lower part of the menu require special HTML code in order to be displayed properly. Clicking on one of these special characters will cause its HTML code to be inserted at the cursor.

# The "Command" Menu

Repeat Last Command

Autoinsert Paragraph Markings

Autoinsert Line Breaks

Create Formatted Link

Insert Horizontal Line

Display Standard Document Template

Mark Selected Text as BODY

Mark Selected Text as HEAD

Mark Selected Text as HTML

## **Repeat Last Command**

This option causes the HTML code inserted as a result of the previous command to be redisplayed at the current cursor position.

This duplicates the function of the "Repeat" button on the document window tool bar.

## **Autoinsert Paragraph Markings**

This will insert new paragraph tags ("<P>") wherever a newline is found in the selected text. Tags will not be inserted on empty lines.

## **Autoinsert Line Breaks**

This will insert line break tags ("`<BR>`") wherever a newline is found in the selected text. Tags will not be inserted on empty lines.

## Create Formatted Link

This option converts a single line containing a URL, followed by its "Clickable" text, to a formatted "<A HREF...", link. The line to be converted must first be selected with the mouse.

For example, the line:

```
www.nsac.ns.ca/,Agricultural College Site
```

would be converted to:

```
<A HREF="www.nsac.ns.ca/">Agricultural College Site</A>
```

Note the comma (",") separating the URL from the clickable text in the original line. This is required, unless there is no clickable text.

## Insert Horizontal Line

Inserts a "Horizontal Rule" tag ("`<HR>`") at the cursor.

## Display Standard Document Template

Displays the HTML tag pairs that are required for delineating the structure of an HTML document.

The following code is displayed at the cursor position:

```
<HTML>  
<HEAD>  
<TITLE> </TITLE>  
</HEAD>  
<BODY>  
</BODY>  
</HTML>
```



## Mark Selected Text as BODY

Encloses selected text with the "body" tag pair, "<BODY></BODY>"

## Mark Selected Text as HEAD

Encloses selected text with the "head" tag pair, "<HEAD></HEAD>"

## Mark Selected Text as HTML

Encloses selected text with the "HTML" tag pair, "<HTML></HTML>"

# The "Special" Menu

Background AssistantTM

QuickFramesTM

URL AssistantTM

Table AssistantTM

Form Tool Bar

## **Background AssistantTM**

Background Assistant is a tool for creating background and text colors in HTML documents. See the section "Background Assistant" for a complete description of this feature.

## QuickFrames™

This feature permits you to easily create pages that are divided into separate windows or "Frames". Simply follow the instructions in the dialogs that appear. When the "Create Frame Set" button is "clicked", a new edit window will be opened with the frame page text on it.

Frame page documents *are different* from other HTML documents - they *do not* have `<BODY>...</BODY>` tags, and each frame tag contains only an attribute that references another HTML document. The frame itself has no content and no closing tag.

Be aware that not all browsers support frames.

## **URL Assistant<sup>TM</sup>**

URL Assistant is a tool for organizing and selecting URLs to be used in your documents. You can use it to combine URLs from various sources, select them by subject or other criteria, and quickly insert them singly, or as groups, into your documents. See the section "URL Assistant" for a complete discussion of this tool.

## **Table Assistant™**

The Table Assistant is a dialog designed to automate the task of creating tables from the HTML table tags. For a complete discussion, see the section "Table Assistant". The tutorial in "Beginners' Guide to HTML" also provides information on working with tables.



## Form Tool Bar

The Form tool bar consists of a set of buttons used to create the individual components of HTML forms. This tool bar is described in detail in the section "Form tool bar". Forms are also discussed in "Beginners' Guide to HTML".

## The "URL" Menu

Items on this menu relate to editing, extracting and generally working with URLs.

Create Local File Reference

Autoconvert File to HTML

Edit/Build HTML Assistant URL files

Automatically Create a Browser Page

## Create Local File Reference

Selecting this option brings up a file select dialog. Selecting a file, causes a local file URL to be inserted at the cursor.

For example, selecting the file, "d:\images\picture.gif" will cause the following text to be inserted at the cursor:

```
file://localhost/d:/images/picture.gif
```

## **Autoconvert File to HTML**

Selecting this option will bring up a sub menu with options for selecting file types from which URLs and their associated "clickable" text will be automatically converted to HTML text. The HTML text will be displayed in a document window.

You may convert URLs from Cello Bookmark files (Cello was the first MS Windows based browser available), NCSA Mosaic's "Mosaic.ini" file, or HTML Assistant format ".url" files.

NOTE: Netscape bookmark files are already HTML files.

## **Edit/Build HTML Assistant URL files**

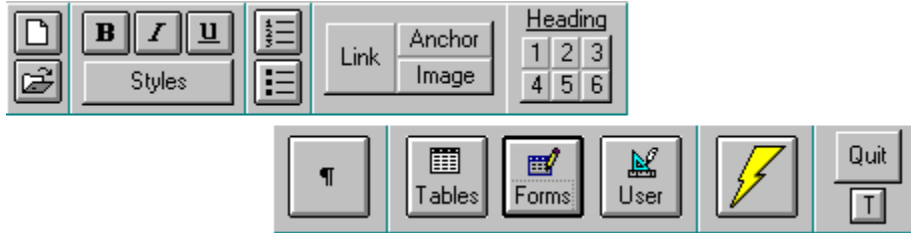
Selecting this item from the 'URL' menu brings up a dialog window that permits you to edit and combine commented URLs from different sources.

## Automatically Create a Browser Page

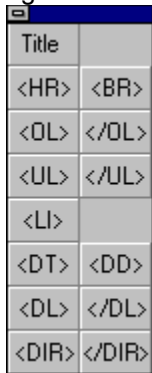
Selecting this option brings up *HTML Assistant Pro's* "Automatic Page Creator" window.

## Tools Overview

*HTML Assistant's* tools can be accessed by means of a tool bar at the top of the program's main window.



There is also menu item labeled "HTML" which permits the insertion of HTML tags, as well as a floating tool palette that contains commonly used HTML tags. The tool palette can be hidden by clicking the small button marked "T" in the lower right hand corner of the main tool bar. The button is a toggle. Clicking again will restore the tool palette.



The tools are basically of two types:

Tools that normally involve the selection of text, and which mark the selected text by *bracketing* it (Type 1 tools.)

Tools that insert HTML elements at the cursor (Type 2 tools.) NOTE: Insertion tools will REPLACE any selected text.

The tools accessible on the tool bar are almost all Type 1 tools that involve the selection of text.

In addition, the tool bar has three special buttons, labelled "Tables," "Forms", and "User", and a button with a picture of lightening bolt on it. These involve more complex operations and are described elsewhere.

To use the Type 1 tools, highlight the text you wish to select by 'dragging' the text cursor across it. Selecting a tool bar item (by 'clicking') will cause the appropriate HTML markings to bracket the selected text.

The tools listed on the "HTML" menu are all Type 2 tools which insert HTML markings at the cursor. 'Clicking' on a menu item will cause the displayed HTML text to be inserted at the text cursor. Any selected text will be *replaced* by the tags.

There is also a special group of tools called 'User Tools' that can be used for inserting tags or bracketing selected text. These are described below.

**See also:**

## The Auto Repeat Function



## Using Bracketing (Type 1) Tools

To use the Type 1 tools, highlight the text you wish to select by 'dragging' the text cursor across it. Selecting a tool bar button (by 'clicking') will cause the appropriate HTML markings to bracket the selected text.

The 'List' tool items work with several lines of text. Select a few lines of text and try them.

A few Type 1 Tools are a bit more complicated. They ask for additional text to be added by the user as part the HTML element that they apply. The 'URL Link' tool (associated with the "Link" button) is one of these. So are the 'Anchor' and 'Image' tools. Try them to see how they work.

To keep up with changes in HTML and to permit you to configure the editor to suit your style of working, *HTML Assistant Pro* provides a special "User Tools" list for your own custom tools - type 1 and type 2.

## The 'Link' Tool

When the "Link" button is selected, a window pops up asking for selection or entry of a 'URL'. URLs (Uniform Resource Locators) are instructions for fetching data. Each URL type has a prefix (as shown in the URL Prefix List box) indicating the kind of data that will be fetched.

As you enter and use URLs in your documents they are saved in memory and appear in the URL list box whenever the URL data entry window pops up. You may save the contents of this list box to a file, and retrieve the contents of a saved file to the list box by accessing the 'URL File' menu on the URL pop up window.

See Edit/Build URL Files.

## Using Insertion (Type 2) Tools

The tools listed on the "HTML" menu are all Type 2 tools which insert HTML markings at the cursor. The tool bar button labeled "<P>" is also a Type 2 tool. 'Clicking' on a menu item, or the button, will cause the displayed HTML text to be inserted.

**Insertion tools will REPLACE any selected text.**

To keep up with changes in HTML and to permit you to configure the editor to suit your style of working, *HTML Assistant Pro* provides a special "User Tools" list for your own custom tools - type 1 and type 2.

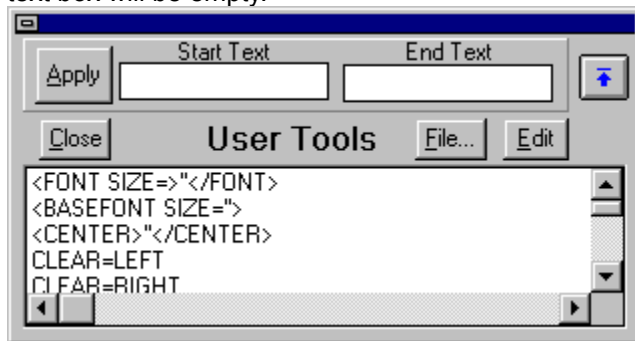
# User Tools

User Tools permit the use of HTML's latest features by means of tools customized with text provided by the user.

This permits users to create new tools, as new HTML tags become available, that can quickly be inserted at the cursor position or be used to bracket text that has been selected with the mouse.

An unlimited number (well...almost) of User Tools can be created. They can be deleted or modified at any time, are automatically saved when exiting the program, and are reloaded when the program starts.

Clicking on an item in the list, will cause the list item's text to be inserted at the cursor, or to bracket text that has been selected by dragging with the mouse. The text will also appear in the text boxes near the top of the User Tools dialog window. Tags that bracket selected text will appear in both text boxes. If the text is to be inserted, rather than bracket, it will appear only in the left hand text box and the right hand text box will be empty.



You can also use the button labelled "Apply" to insert or bracket text in the edit window. In this case, the text in the text boxes will be either inserted or used to bracket any selected text. If the right hand text box is empty, the text will be inserted and any selected text will be replaced. If the right hand text box is not empty, selected text will be bracketed by the text in the text boxes. Note that you can automatically enter text into the text boxes and use the "Apply" button to enter it into the document. However, this text will *not* be part of the saved User Tools file.

When you exit *HTML Assistant Pro*, the User Tools in the list will automatically be saved for use in future editing sessions.

User Tools for "Bracketing" or "Surrounding" Text

## User Tools for "Bracketing" or "Surrounding" Text

User Tools can be used to bracket selected text, as well as for inserting text at the cursor. When using the User Tools edit window, place the opening and closing text in the left and right hand text boxes, respectively.

The text will be saved with two *single* quotation marks (with NO space between them) in a User Tool to represent the selected text.

For example, to create a user tool to mark selected text as "centered", you would use the following text in the Edit User Tools dialog:

Left text box: "<center>"

Right text box: "</center>"

The User Tool would be saved as

<center>"</center>

Note the *two single quotation marks* (not a double quote!) between the tags.

### See also:

Creating Tools that Insert an EOL (Carriage Return/Line feed)

## Editing User Tools

The User Tools list can be changed by clicking on the 'Edit User Tools...' item in the "Options" menu, clicking with the right mouse button on the tool bar button marked "User", or by selecting the "Edit" button on the User Tools dialog window.

Any of these actions will cause an "Edit User Tools" dialog box to appear.

The buttons on the dialog box, along with the contents of the text boxes at the top, are used to modify the contents of the User Tools list. Use the buttons to delete, insert, append and replace list items.

Click on the "OK" button to copy the edited list to the User Tools list on the User Tools dialog window.

Selecting the "Close" button will cause the edit dialog box to close *without* altering the User Tools list on the User Tools dialog.

**Edit User Tools**

File

Start Text      Edit      End Text

☐ Autoset HTML Tags                 

Double click on a list item to edit it

<FONT SIZE=>"</FONT>  
<BASEFONT SIZE=>  
<CENTER>"</CENTER>  
CLEAR=LEFT  
CLEAR=RIGHT  
CLEAR=ALL  
<HR ALIGN= SIZE= WIDTH=%>  
NOSHADE  
<NOBR>

Copy edited list to User Tools window and exit User Tools editor

## User Tools Files

The "File" menu on the "Edit User Tools" dialog permits you to save your lists of User Tools as files, and to reload them for later use. In this way, you can build sets of special purpose user tools as the need arises. For example, you might have a set of tools for creating forms, another for a specialized application that involves tables, etc. An option in the "File" menu permits you to combine different User Tools files by appending to the list box.

User Tools files have the suffix, ".UT". A file called "DEFAULT.UT" is automatically loaded to the User Tools list when HTML Assistant starts. The User Tools list is saved to this file when the program ends.

## Creating Tools that Insert an End-of-Line

To create a user defined tool that inserts a carriage-return/linefeed pair (End-Of-Line marker) enter "\n" -- without the quotation marks -- in the list box item where you want the EOL to be created.

For example to create a tool that inserts an EOL followed by a paragraph mark followed by another EOL you would use the following text in the tool's list box text:

```
\n<P>\n
```



## The Auto Repeat Function

To repeat the action of the tool that was last used, simply press press CTL-R. This function is also available from the 'Command' menu and as a command button in each file display window.

## **Autoinsert Paragraph Markings**

Found in the 'Command' menu this item will insert new paragraph marks ("`<P>`") at every End of Line (carriage return-line feed pair) in the selected text. You must select text for this command to work.

## Printing

The contents of the current edit window may be printed by selecting the 'Print' option in the file menu.

This is a simple print routine that prints in a monospaced font on the default printer.

For more elaborate print formatting you can use any word processor that will accept standard ASCII text files.

## What is *Background Assistant*?

*Background Assistant* is a WYSIWYG (What-You-See-Is-What-You-Get) aid for creating background and text colors in HTML documents.

This version of *Background Assistant* is part of the *HTML Assistant Pro* hypertext editor. *Background Assistant* is also available as a stand alone application for use in conjunction with other editors.

The color and background image tags created with *Background Assistant* can be used by browsers that support "BGCOLOR" and "BACKGROUND" attributes in the starting <BODY> tags of HTML documents. *Netscape* (Version 1.1 and higher) introduced these elements and other browsers may also incorporate them.

Colors for background and text are edited by moving on-screen sliders with the computer's mouse. You can see what the colors will look like when displayed by your browser while you are creating them.

If you wish, instead of using a background color, you can use a small "gif" or "jpg" image, which is automatically "tiled" by a browser, as a background.

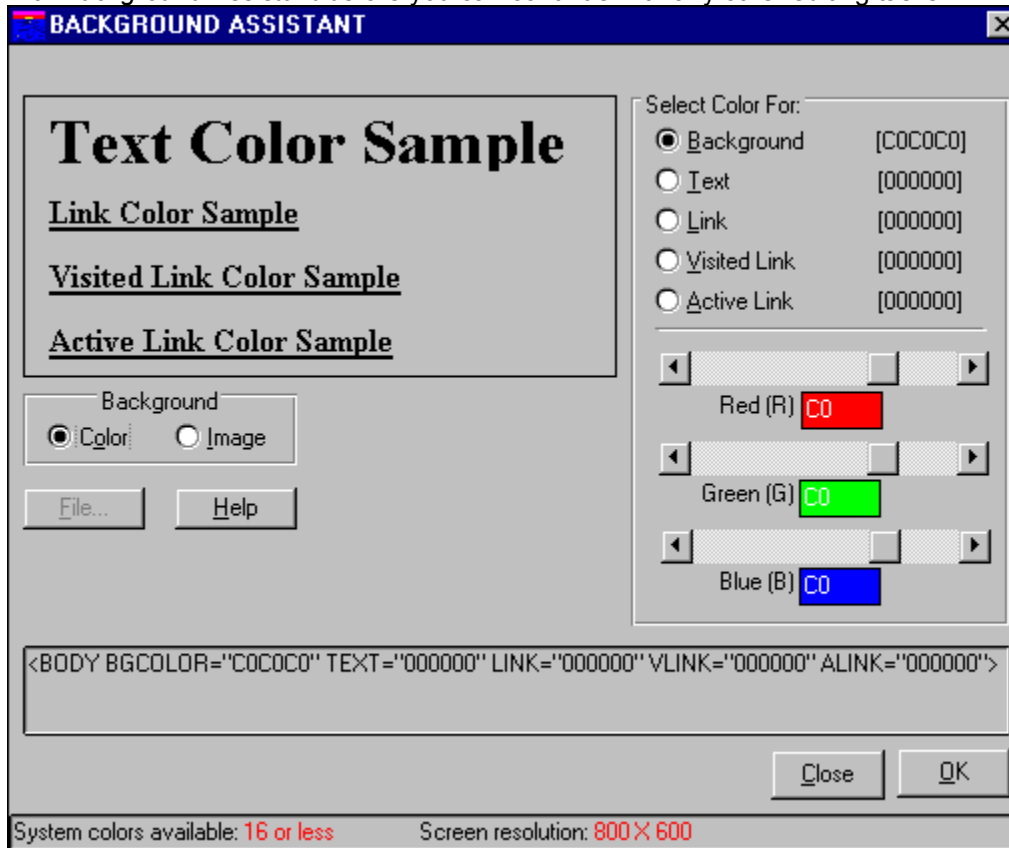
Once you are satisfied with the background color (or tiled image) and text colors, with a mouse click you can insert the <BODY> tag, with its background and text color attributes, into your HTML document.

In a simple and intuitive manner, *Background Assistant* permits you to add interest and excitement to your WWW pages.

## Starting Background Assistant

From *HTML Assistant Pro*, you can start *Background Assistant* by selecting the "Background Assistant" item from the "Special" menu, or you can use the <F5> "shortcut key."

*Background Assistant* opens as a "modal" dialog window. "Modal" means that you must finish working with *Background Assistant* before you can continue with any other editing tasks.



During an editing session, the first time you start *Background Assistant*, it opens in "Color" background mode. Using this mode will result in a plain single color background for your document. You can also choose to use "Image" background mode. In that case, your document's background will be patterned, or textured, with a "tiled" image.

Selecting the "Background" option button in the box labeled "Select color for:", located in the upper right portion of the dialogue window makes the *background* of the image display box (in the upper left portion of the window), the object of color changes. The first time that the dialog window appears, this option is already selected.

Moving any of the three (Red, Green and Blue) color sliders, will result in a change to the background color of the display.

The manner in which colors change as you move the sliders depends upon the number of colors available on your computer. If you are using 16 or 256 color (4 or 8 bit) graphics, then the colors may change suddenly when a slider reaches a particular position. With 64K or 16 million colors (16 or 24 bit graphics), the color changes will be continuous and gradual.

**Note: If you are using 256 or 16 color graphics mode, the colors used as backgrounds will be**

**limited to those found in the Windows default palette. Colors not part of this palette will be "dithered" by Windows, using combinations of existing colors - resulting in the appearance of gridlike effects on the image display box background. Text colors will not be dithered, but will be limited to the Windows default color set.**

In addition to the background, four text items in your document may have their colors modified:

Text - headings, lists, plain text, etc.

Link - "clickable" text used for accessing other files, documents or images

Visited Link - a link that has previously been used or "visited", and which is marked as such by the browser's use of an alternate color

Active Link - the link which is currently selected by placing the mouse cursor over it and pressing the left mouse button

You can select any of these text options for color modification, either by clicking on the appropriate text in the image display box, or, by clicking on its option button.

Note that the <BODY> tag with its currently selected attributes is displayed in the gray inset box near the bottom of the *Background Assistant* window.

# Using a Tiled Image as a Background

You can use a graphic as a background for your Web page.

Instead of using a simple colored background, *Background Assistant* permits you to preview a picture file which the browser displays repeatedly in adjacent rows and columns so that it covers the entire background. That is, the image is "tiled" to form the background of your document.

Usually, the files used for HTML document backgrounds are textures or patterns that provide a pleasing surface upon which the text is displayed.

Because the image is tiled, even though it fills the entire document background, the actual image file can be quite small. This permits the use in your documents of attractive, large scale graphic designs without requiring the user to download large files.

These background textures or designs can be created, and saved in "gif" format, with any one of a number of graphics packages such as Corel Draw, Adobe Photoshop, or a number of shareware products. Texture files in "gif" or "jpg" format are also available on the Internet.

If you don't already have the software or the images that are required, you can use one of the "search" sites on the Web to find what you need. Start your search by using the word "graphic" or "image" as the search string.

To use a tiled image as the background for your document, select the "Image" option button in the box labeled "Background", immediately below the image display box.

Selecting the "Image" option will cause the "File..." button, immediately below the "Background" option buttons to become enabled, and a small white "picture box" will appear to the right of the "File..." button.

Click the "File..." button, and choose the "Load image..." option from the popup menu that appears. Next, using the file select dialog box, select the image file that you want to use as your background. Browsers can usually display files in "GIF" or "JPEG" format, but you can display an image in another format and have *Background Assistant* convert it for you.

Assuming that you have selected an appropriate image file, the image itself will appear in the small picture box. Almost simultaneously, the tiled image, with the colored text superimposed on it, will appear in the image display box above.

The <BODY> tag with the appropriate attributes will be displayed in the gray inset box near the bottom of the *Background Assistant* window.

At any time, you can select another image file and it will become the tiled image in the image display box, with your colored text superimposed.

You can also switch back and forth between a plain color background and a tiled image by selecting the appropriate option button in the "Background" options box.

Possible Problem Displaying Some "GIF" Files

## Possible Problem Displaying Some "GIF" Files

Some "gif" files may not be able to be displayed by *Background Assistant*. Instead an error message will appear to the effect that the file is of the wrong type. The third party custom control we use for loading images cannot, at this time, display "gif" files that have been saved in the "interlaced" format. *Interlaced* format files are the ones that are displayed in alternate sections by your browser. Most graphics packages *do not* save files in this format. However, these files are available on the Internet. Be aware that if you want to use an interlaced format "gif" file with *Background Assistant*, you must first load it into an appropriate graphics software package and resave it in *noninterlaced* format - the default format in which most graphics packages save "gif" files.



## **File Conversion: Use of Files Other Than GIF or JPG**

In addition to "gif" or "jpg" files, *Background Assistant* will permit you to load other graphics file formats.

This version of *Background Assistant* will permit you to *convert* these files to "gif" or "jpg" files that can be displayed by browsers.

Allowable formats include "raster image" or "bit mapped" files with the suffixes ".tif", ".bmp", ".dib", ".pcx" and ".wpg" files that have been saved as raster images. You can also load some "vector image" files (with suffixes ".wmf", ".eps" and ".wpg" files that are not saved as raster images), but they cannot be tiled or converted.

You can use the "Save as..." option in the "File..." button popup menu to convert and save a copy of the file as a "gif" or "jpg" image. The converted image can be used by your browser.

## Transferring the <BODY ... > Tag to your File

When you are satisfied that the background and text colors are to your liking, simply click the "OK" button. This will insert the <BODY> tag with its attributes into your HTML document, at the place where the cursor was when you opened the *Background Assistant* dialogue window. If you had selected any text -- a pre-existing body tag, for example -- it will be replaced.

Upon returning to the editor, you can reverse any changes made by *Background Assistant*, by using the editor's "Undo" feature.

You can exit *Background Assistant* without inserting a tag, or replacing any selected text, by using the "Close" button. The existing settings will be "remembered" by 'Background Assistant' and you will be returned to the editor, with your document unchanged.

## Preselecting an Existing <BODY ....> Tag

Before opening *Background Assistant*, you may select, by dragging the mouse cursor across it, an existing body tag with its attributes for background and text colors.

This will cause *Background Assistant* to attempt to *preselect* the background mode and color combinations used in that tag. If this is successful, the dialog window will appear with the tag's colors and image, if any, already displayed.

## Where to Insert the <BODY ... > Tag

A properly formatted HTML document will incorporate the following tags, which serve as boundaries for the formal sections of the text:

The <HTML></HTML> tag pair brackets the entire document

The <HEAD></HEAD> tag pair brackets the title and, optionally, certain other descriptive information that is not part of the document text

The <TITLE></TITLE> tag pair brackets the document title

The <BODY [with attributes, if any]></BODY> tag pair brackets the document text, links, images, etc. This is what most people normally think of as the contents of their page.

Diagrammatically it looks like this:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Your Document's Descriptive Title</TITLE>
```

```
</HEAD>
```

```
<BODY BGCOLOR="FFFFFF" TEXT="FF0000" LINK="0000FF" VLINK="FF00FF" ALINK="000000">
```

Your document -- text, links and images -- should be here

```
</BODY>
```

```
</HTML>
```

**Note that the starting <BODY ... > tag, with its attributes, must be the *first* item in your document following the closing </HEAD> tag, and only one <BODY ... ></BODY> tag pair may be used.**

While most browsers will format your document properly if you do not include all the bounding tags, it is good HTML practice to use them.

If you choose not to include the <HTML> and <HEAD> tags, be sure that the starting <BODY ... > tag is the first tag following the closing </TITLE> tag. If you don't use the <TITLE></TITLE> tag pair, ***be sure that the starting <BODY ... > tag precedes any other text or headings in your document.***

## Overview

QuickFrames is a tool designed to automate the creation of pages divided into frames. Frames are separate windows displayed on a single HTML page.

Each frame can contain a separate HTML document, permitting several different documents to be displayed at once. Moreover, individual frames may optionally be scrolled, sized and displayed independently.

Frame page documents *are different* from other HTML documents - they *do not* have `<BODY>...</BODY>` tags, and each frame tag contains only an attribute that references an HTML document. The frame itself has no content and no closing tag.

QuickFrames makes the process of creating frames easy and automatic.

## Using QuickFrames

To access QuickFrames, a document edit window must be open.

To start QuickFrames click on the 'Special' menu and select 'QuickFrames' or press the Shift and F5 keys (Shift-F5) simultaneously.

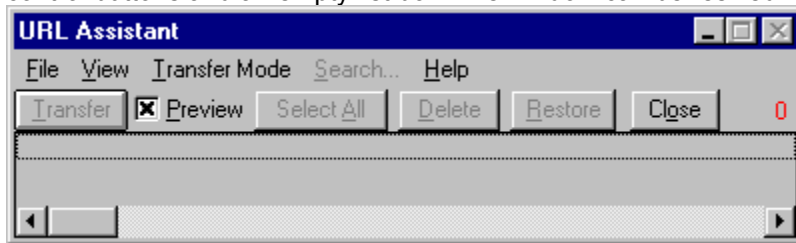
Then, simply follow the instructions in the dialogs that appear. When the "Create Frame Set" button is "clicked", a new edit window will be opened with the frame page text on it.

## URL Assistant Overview

URL Assistant is a tool for organizing and selecting URLs to be used in your documents. You can use it to combine URLs from various sources, select them by subject or other criteria, and quickly insert them singly, or as groups, into your documents.

To access the URL Assistant, a document edit window must be open.

Selecting the "URL Assistant" option from the "Special" menu, or pressing the F2 key, will cause the URL Assistant window to appear on your screen. It consists of a floating window with a menu bar, some control buttons and an empty list box. The window can be resized.



Using the various options in URL Assistant's "File" menu, the list box can be filled with URLs and their associated 'clickable' text or comments. The URLs that fill the list box can come from HTML Assistant URL files, or can be extracted from HTML documents (Netscape Bookmark files are HTML documents), Cello (an early browser) Bookmarks or a Mosaic.ini file.

By checking the "Append" option in the file menu, URLs from several sources may be combined.

Double-clicking on a list item, or selecting the "Transfer" button, will cause the URL and its associated comment to be transferred to the active text window at the cursor position.

# Transferring URLs from the URL Assistant List

URL Assistant's list box is used to hold URLs which may be selected for transfer to your HTML files.

The URLs that fill the list box can come from HTML Assistant URL files, or can be extracted from HTML documents (Netscape Bookmark files are HTML documents), Cello Bookmarks (Cello is an early browser) or a Mosaic.ini file.

By checking the "Append" option in the file menu, URLs from several sources may be combined.

The URLs in the list box may be saved as an HTML Assistant URL file, or they can be automatically reformatted in HTML and saved as an HTML file.

By default, when the list box is filled, you will see both the URL and its associated comment or clickable text. You can choose an option from the "View" menu to see only the URLs, only the comments, or both.

Double-clicking on a list item, or selecting the "Transfer" button, will cause the URL and/or its associated comment to be transferred to the active text window at the cursor position.

By selecting the appropriate option from the "Transfer Mode" menu, you can cause only the URL, only the comment, or both to be transferred.

You also have the option to automatically convert the URL and its comment to HTML code as it is transferred.

Other options in the "Transfer Mode" menu can cause the transferred item to be removed from the list, and have a newline added when it is transferred.

**Note that the "Transfer Mode" and "View" options are *independent*.** That is, you can choose to display only comments, and can transfer URLs *and* comments.

The Preview Option

User Selected Text as Comment

Deleting Items From the List

Multiple Selections



## The Preview Option

If the "Preview" check box, next to the "Transfer" button, is selected, you will have an option to preview or edit any text to be transferred before the transfer actually takes place. After previewing the text, you will be able to cancel the transfer if you wish.

## User Selected Text as Comment

The "Transfer Mode" menu has an option for "User selected text as comment". If this is selected (checked), *regardless of what other options in the "Transfer Mode" menu are selected*, the URL will be converted to HTML code with the user selected (highlighted) text in the active document as the clickable text.

## **Deleting Items From the List**

Selection one or more items from the list and clicking the "Delete" button will cause them to be deleted from the list.

An option in the "File" menu will permit clearing the entire list.

Clicking the "Restore" key will cause the original list box contents to be restored.

## Multiple Selections

You can select multiple items from the list for transfer, or deletion, as follows:

Clicking the "Select All" button will cause all list items to be selected.

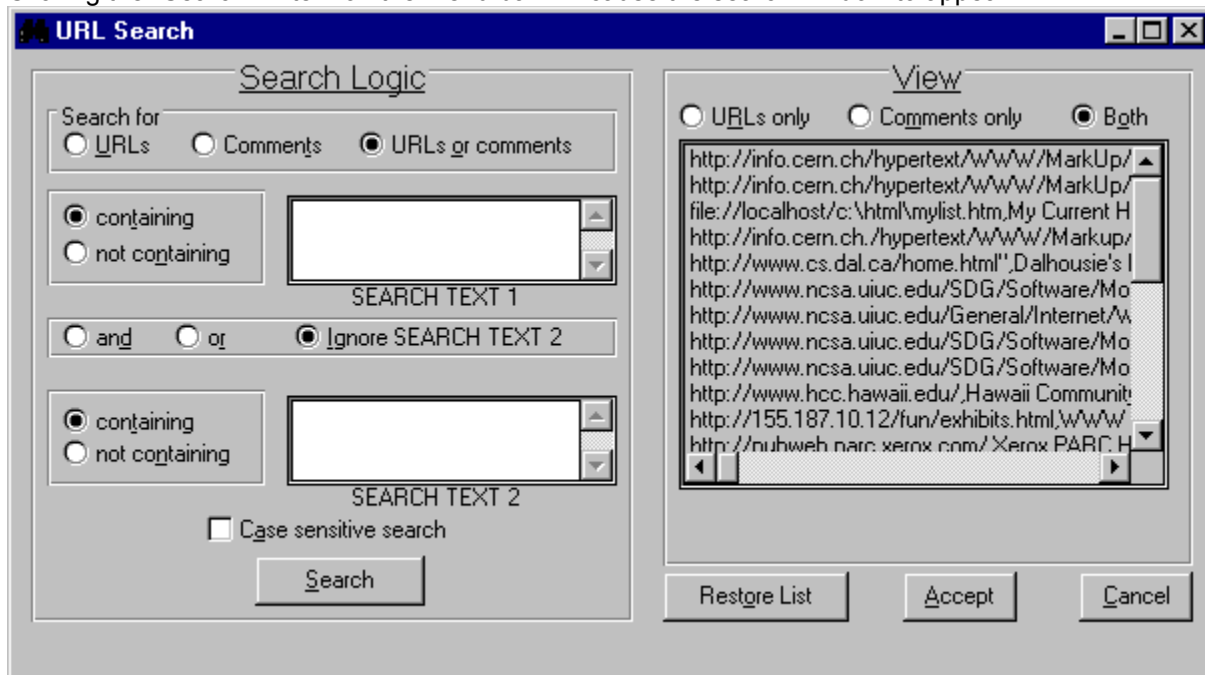
Dragging the mouse up or down the list will select a continuous group of items.

Clicking the mouse on an item, followed by holding the "Shift" key while clicking on another item, will cause the two items and all the list items between them to be selected.

Clicking while holding the "Control" key will permit multiple, but discontinuous, items to be selected.

## Searching the List

Clicking the "Search..." item on the menu bar will cause the search window to appear.



The search window, like the URL Assistant window, permits various options for viewing and searching list items.

You can select key words to search for in the list, and the resulting selections will appear in the search window's list box.

The view and search options are independent. For example, you can search for key words in the comments field even though the comments field might not be displayed.

Clicking on the "Accept" button will close the search window and cause the list items in the search window's list to be transferred to the URL Assistant list, *replacing the items that are already there*. URL Assistant's full list can be brought back by selecting the "Restore" button.

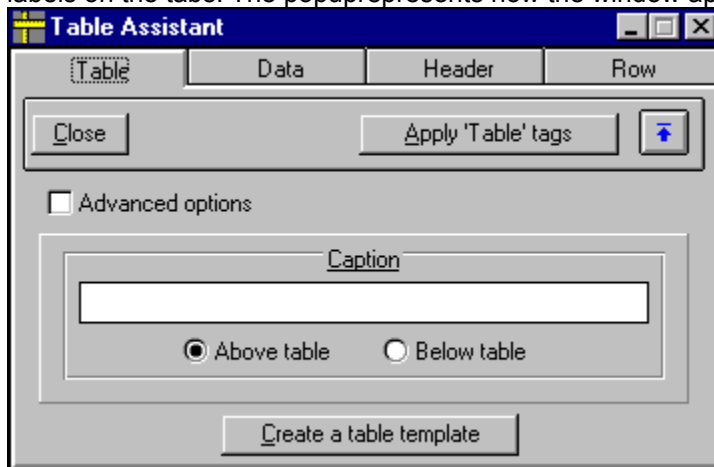
Using the "Cancel" button will close the search window without transferring its list.

## Table Assistant Overview

Table Assistant is a tool to aid the user in working with the HTML 3 tags which have been proposed for creating tables and are supported by Netscape and other browsers.

To access the Table Assistant, a document edit window must be open.

Selecting the "Table Assistant" option from the "Special" menu, pressing the F11 key, or clicking the Tables button on the main tool bar, will cause the Table Assistant window to appear on your screen. This window has four "tabs" which when clicked causes the contents of the window to change according to the labels on the tabs. The popup represents how the window appears if the "Table" tab is selected.



Each tab selection provides options which allows the user to supply information about the various components which make up HTML tables, namely information which applies to the table as a whole, as well as information defining the formats of the data, headers, and rows of the table. Once the user has filled in the information as needed, the "Apply" buttons can be used to cause the corresponding HTML tags to be inserted into the current document.

If the user plans to create several tables, the button with the arrow on it can be clicked to cause the Table Assistant window to collapse to display only the Close and Apply buttons. This allows the user to keep the Table Assistant window active without taking up too much screen space. The collapse button is a toggle and clicking it a second time will cause the window to return to its former size.

As an alternative to filling in all these options manually, if the "Table" tab is selected the user can select the "Create a table template" button. This displays a new window with a few minimal options to enable the user to create a basic table with n rows and m columns without having to go through the lengthy process of filling in all the information normally required by the Table Assistant. Once the template has been created and inserted in the user's document, it can be edited to better suit the user's needs.

The Close button closes the Table Assistant window. Any options which are set are remembered the next time the window is opened.

# The Table Assistant Tabs

This section describes the four main components of the Table Assistant window.

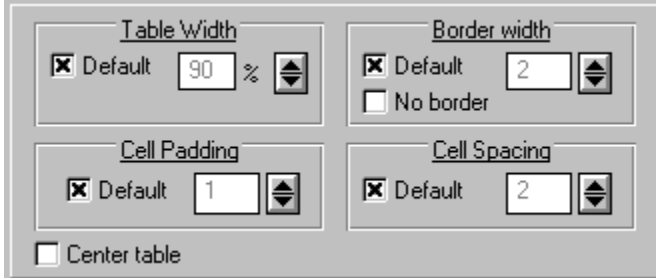
Table Tab

Row Tab

Data and Header Tabs

## Table Tab

If the user chooses not to create a simple template, the user must fill in the individual fields defined by each of the four tabs. The Table tab of the Table Assistant window lists a set of options which applies to the table as a whole. By default, a field for the table "caption" is the only option visible to the user. If the "Advanced options" check box is clicked, additional options are made available, such as the width of the table.



The screenshot shows the 'Table' tab of the 'Table Assistant' window. It contains four sub-sections for configuring table appearance:

- Table Width:** A checked checkbox labeled 'Default' followed by a text box containing '90' and a '%' symbol, and a vertical spinner button.
- Border width:** A checked checkbox labeled 'Default' followed by a text box containing '2' and a vertical spinner button, and an unchecked checkbox labeled 'No border'.
- Cell Padding:** A checked checkbox labeled 'Default' followed by a text box containing '1' and a vertical spinner button.
- Cell Spacing:** A checked checkbox labeled 'Default' followed by a text box containing '2' and a vertical spinner button.

At the bottom left, there is an unchecked checkbox labeled 'Center table'.

*Figure 1*

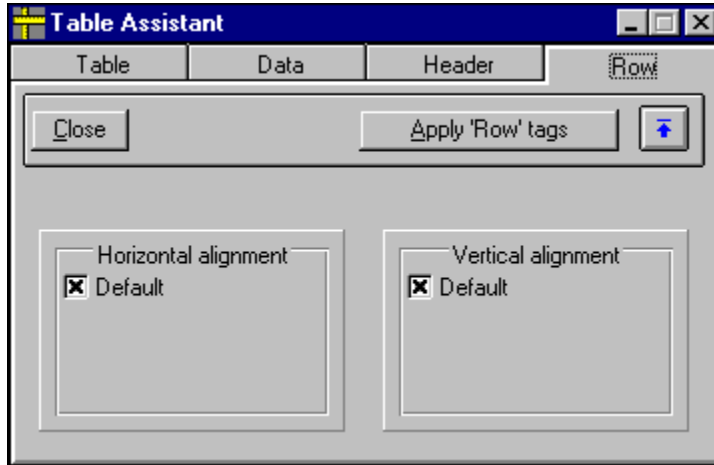
These additional options allow the user to provide more detailed information about the appearance of the table, including its width on a page, the size of the borders, as well as how the cells are spaced. These fields all have default values so it is not necessary for the user to modify them unless a different appearance is desired.

Once the user has filled in whatever fields are needed in the Table portion of the Table Assistant, the HTML tags can be inserted into the current document by clicking the "Apply Table tags" button. The tags are inserted at the cursor position, so the user should be sure the cursor is at the desired location.



## Row Tab

Each HTML table consists of a number of rows and the options in the Row tab portion of the Table Assistant determine the characteristics of the rows.



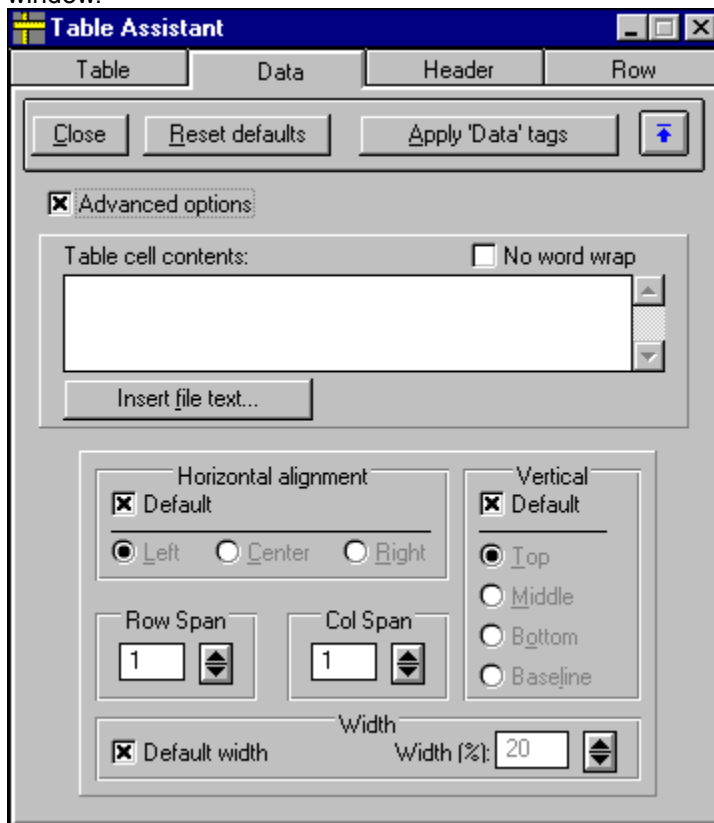
All of the options deal with how the rows are aligned horizontally and vertically. Once the user has set these options as desired, the "Apply Row tags" buttons is used to cause the corresponding `<TR> ... </TR>` HTML tags to be inserted into the current document. The tags are inserted at the cursor position, so the user should be sure that the cursor is at the desired location. Note that a `<TR>` tag can only appear within a `<TABLE> ... </TABLE>` tag pair.

## Data and Header Tabs

Each row of a table is made up of some number of cells to hold the table's data. The Data tab is used to define these cells. Header cells are similar to data cells except that the text is usually highlighted in some manner. Header cells are intended to appear at the top of columns or to the left of rows, although this is not required.

Selecting the Data or Header tab causes the Table Assistant to display a set of options geared to formatting data cells. There is no difference between the options displayed for data and headers. As with the Table tab, there is a check box which can be turned on to display a set of "advanced" options.

All of these options are used to specify the characteristics of a data cell, such as the alignment of the text within the cell, the number of columns and rows it spans, and its contents. After the user has set the options as desired, the "Apply" button is used to insert HTML tags defining a single data cell with these characteristics. These tags are `<TD> ... </TD>` for normal cells and `<TH> ... </TH>` for header cells. They are inserted at the current cursor position and the user should insure that the tags are inserted within a proper `<TR> ... <TR>` pair. The Apply button can be pressed as many times as needed; each time it is pressed another TD or TH tag set is inserted with the characteristics as defined in the Table Assistant window.



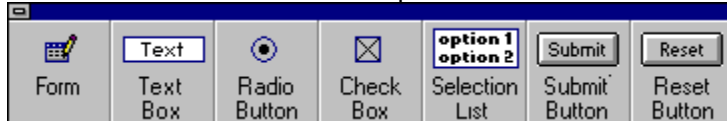
For more information on how to create tables, see the tutorial in "Beginners' Guide to HTML". For a detailed description of the HTML table tags, see the "HTML Reference" section.

## Form Tool Bar Overview

The form tool bar provides a set of tools to aid the user in working with the HTML 2 tags defined for creating forms.

In order to use the tools provided on the form tool bar, a document edit window must be open.

Selecting the "Form tool bar" option from the "Special" menu, pressing the F8 key, or clicking the Forms button on the main tool bar, brings up the form tool bar. This is floating palette with a set of seven buttons used to create different form components.



Each button brings a window in which the user can supply information about a particular form element. Once the user has filled in the information as needed, the corresponding HTML tags can be inserted into the current document.

# The Form Tools

This sections describes each of the tools provided on the Form tool bar.

Form

Text Box

Radio Button

Check Box

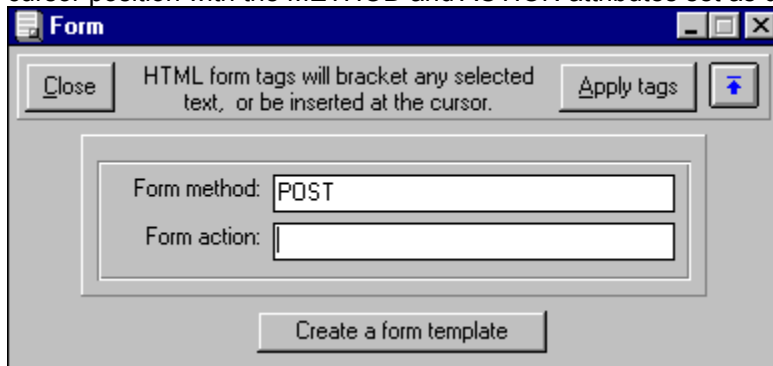
Selection List

Submit and Reset

# Form

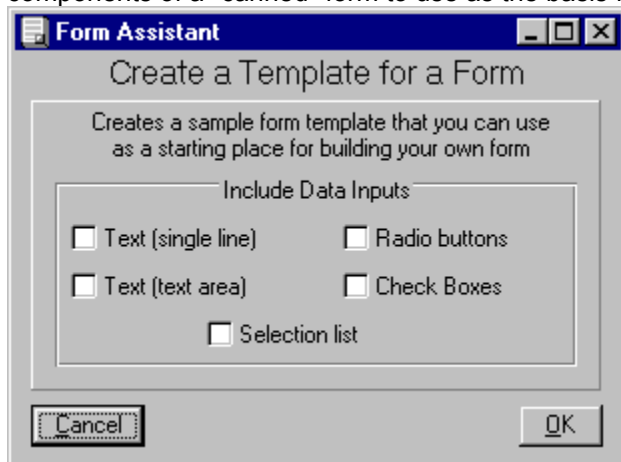
All form elements are enclosed in <FORM> ...</FORM> tags. Clicking the Form button on the Form tool bar brings up a window which is used to define the characteristics of these enclosing tags.

The two text fields in this window are used to define the METHOD and ACTION attributes of a <FORM> tag. Clicking on the Apply button inserts a set of <FORM> ... </FORM> into the document at the current cursor position with the METHOD and ACTION attributes set as defined in the window.



The button with arrow on it is used to collapse the window so that only the Close and Apply buttons are visible. This allows the user to leave the Form window active without taking up too much screen space.

The "Create a form template" button brings up a separate window in which the user can easily specify the components of a "canned" form to use as the basis for designing a form.



The user can select which of the six types of form elements to appear in this form template. Clicking OK inserts a <FORM> tag pair along with any other tags needed to represent the selected form elements.

## Text Box

Clicking the Text Box button brings up the "Text Box" window.

This window is used to specify the characteristics of a text box form element. This type of element is the same as that used in typical dialogs, such as the print dialog. It is a field in which the user can enter a line or lines of text. The user can optionally specify a caption to be associated with the form and if the advanced options are displayed, the placement of the caption can be specified as well. The advanced settings give the user control over other characteristics of the text box as well, such as the size of the text box and how the text is formatted within the box. If "Single line" is selected, the Apply button inserts `<INPUT TYPE="TEXT">` tag along with attributes representing the options that are selected in the Text Box window. If "Multiple line" is selected, the Apply button inserts a `<TEXTAREA>` ... `<TEXTAREA>` tag pair. See the HTML Reference section for a complete description of the various tags and attributes for creating text fields in forms.

**Text Box**

Text Box Input

Close Text input tags will be inserted at the cursor. Apply tags

☒ Advanced

Style

☒ Single line ☐ Multiple line

Displayed caption Variable name

Caption Position

☒ Left ☐ Right  
☐ Above ☐ Below

Use caption as variable name: ☐

Add end-of-line

☐ None  
☒ Line break (<BR>)  
☐ Paragraph (<P>)

Style - Single line text box

Maximum length (characters): 0 (0 = No limit)

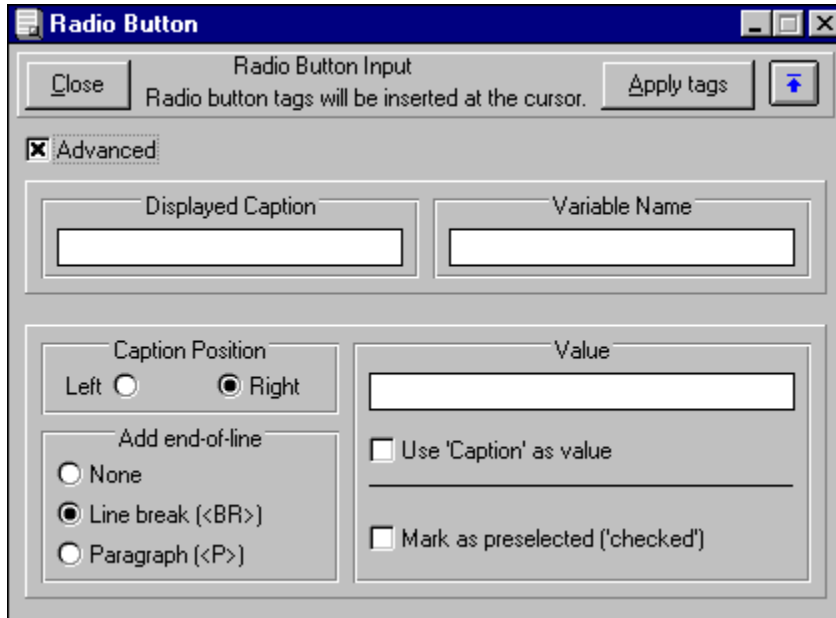
☐ Password field ☐ Hidden field

Default text (if any):

Textbox width (characters): 20

## Radio Button

The Radio button tool opens a window used to define characteristics of a typical radio button. The advanced options, if visible, allow the user more flexibility in how the radio button is formatted, such as the position of its caption and whether or not the button is initially selected. The Apply tag inserts a `<INPUT TYPE="RADIO">` tag into the current document, along with attributes reflecting the options in the window.



The image shows a dialog box titled "Radio Button" with a standard Windows-style title bar. Inside the dialog, there is a "Close" button on the left and an "Apply tags" button on the right, with a small icon of a blue arrow pointing up. Below these buttons, a text label reads "Radio Button Input" and "Radio button tags will be inserted at the cursor." Below this, there is a checkbox labeled "Advanced" which is checked. The main area of the dialog is divided into several sections. The first section contains two text input fields: "Displayed Caption" and "Variable Name". The second section, titled "Caption Position", has two radio buttons: "Left" (unselected) and "Right" (selected). The third section, titled "Add end-of-line", has three radio buttons: "None" (unselected), "Line break (<BR>)" (selected), and "Paragraph (<P>)" (unselected). The fourth section, titled "Value", contains a text input field and two checkboxes: "Use 'Caption' as value" (unselected) and "Mark as preselected ('checked')" (unselected).

Radio Button

Close Radio Button Input Apply tags

Radio button tags will be inserted at the cursor.

☒ Advanced

Displayed Caption Variable Name

Caption Position  
Left ☐ Right ☒

Add end-of-line  
☐ None  
☒ Line break (<BR>)  
☐ Paragraph (<P>)

Value  
  
☐ Use 'Caption' as value  
☐ Mark as preselected ('checked')

## Check Box

The Check Box button opens a window used to define characteristics of a typical check box. The advanced options, if visible, allow the user more flexibility in how the check box is formatted, such as the position of its caption and whether or not the check box is initially checked. The Apply tag inserts a `<INPUT TYPE="CHECKBOX">` tag into the current document, along with attributes reflecting the options in the window.

**Check Box**

Check Box Input

Check box tags will be inserted at the cursor.

☒ Advanced

Displayed Caption:

Variable Name:

Caption Position:  
Left ☐ Right ☒

Add end-of-line:  
☐ None  
☒ Line break (<BR>)  
☐ Paragraph (<P>)

Value when selected:

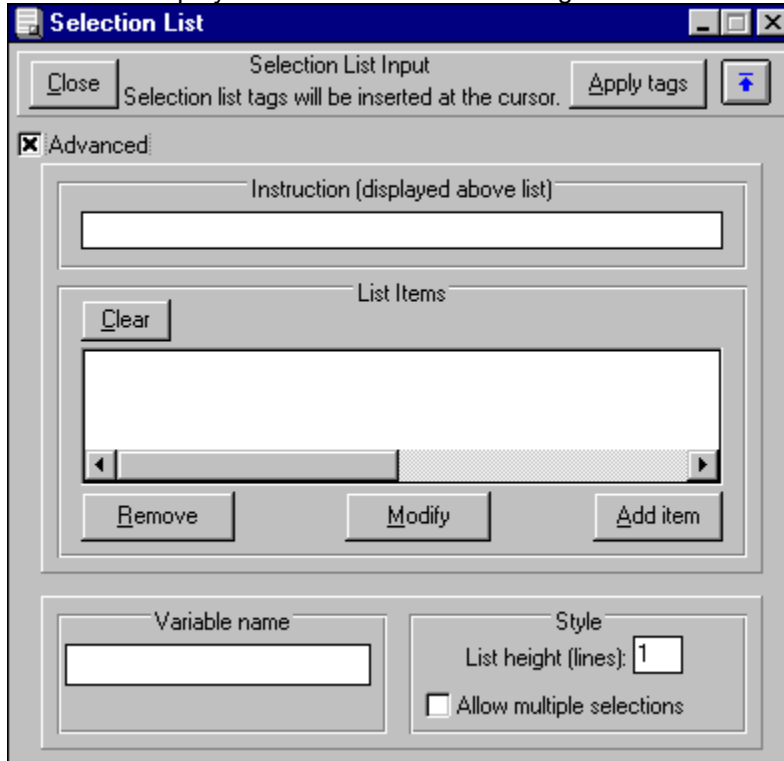
☐ Use 'Caption' as value

☐ Mark as preselected ('checked')



## Selection List

A selection list is a typical scrolling list, such as the list of files display in a file open dialog. The selection list window displays a set of controls for building such a list.



The most important area of this window is the section labelled "List Items". This displays the items the user wants to appear in the selection list. There are buttons for adding items to this list, modifying them, and deleting them. The list can be given an optional caption or "instruction". If the advanced options are turned on, additional characteristics can be defined, including the height of the list and whether or not the user can select multiple items from the list at one time.

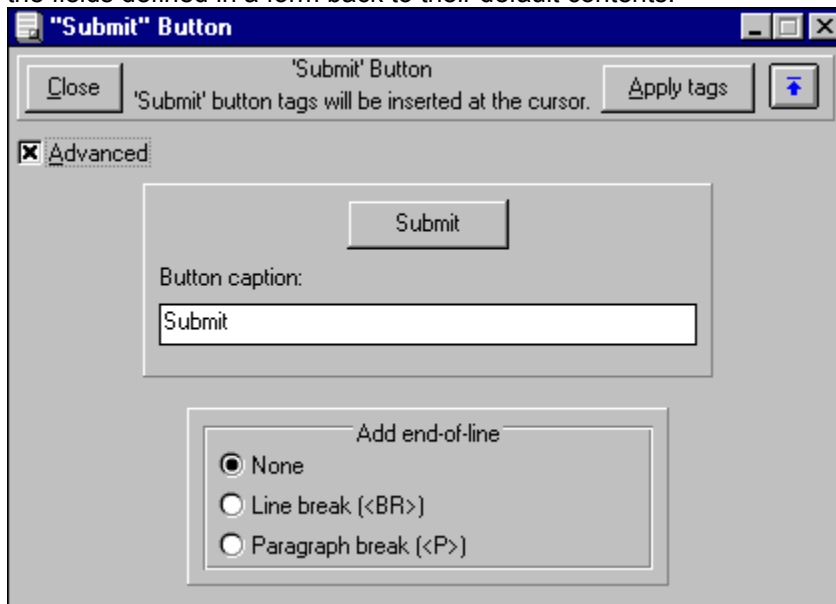
Once the list has been defined, clicking on the Apply button causes a `<SELECT> ... </SELECT>` set of tags to be inserted into the current document, bracketing `<OPTION>` tags corresponding to each list item which has been defined.

## Submit and Reset

Every form a Web page designer creates will most likely have a "submit" button defined for it. The submit button is used to submit the form data for processing. The "Submit Button" tool on the form tool bar opens a window containing options to define a form submit button, namely the name assigned to the submit button and whether or not a line or paragraph break should be inserted after the button. This later option is available only if "Advanced" is selected.

Clicking Apply causes a `<INPUT TYPE="SUBMIT">` tag to be inserted at the current cursor position.

The "Reset Button" tool opens a window which is identical to the "Submit Button" window except the default caption is "Reset" instead of "Submit". This is used to insert a Reset form element which resets the fields defined in a form back to their default contents.



For more information on how to create forms, see the tutorial in "Beginners' Guide to HTML." For a detailed description of the HTML form tags, see the "HTML Reference" section.

## Edit/Build URL Files

Selecting this item from the 'URL' menu permits you to edit and combine commented URLs (URL is an abbreviation for Uniform Resource Locator) from different sources.

For example, you may quickly combine selected URLs from HTML Assistant URL files with other URLs automatically extracted from Mosaic menus and Netscape bookmarks.

While perusing news groups, mailing lists, etc., you might cut and paste URLs into text files (with Notepad or *HTML Assistant Pro*) and later combine these with other files.

Of course, once the URLs and comments from various sources are loaded into the editor's list boxes they may be edited and supplemented with new URLs that are manually typed in. If they are saved as HTML Assistant URL files (the default type) they can later be loaded into an edit window for more extensive editing and reorganization.

You may also convert and save any combination of URLs from any of these sources into a Cello Bookmark file (Cello was one of the first browsers available).

### **See also:**

Operation of the URL Editor

## Operation of the URL Editor

When the 'Edit/Build URL Files' option is chosen, a window with two list boxes and two text boxes, and some control buttons pops up.

When a file is loaded, any necessary conversions or extractions are performed, and the list boxes are filled with the URLs from the file (the files themselves remain unchanged). One list box contains a list of all the URLs found in the selected file. The other list box, in corresponding positions in the list, contains comments or descriptions of the URLs in the first box.

You may create your own files or add to the contents of existing files, by clicking the 'Add New URL' button and entering URLs and Comments into the corresponding text boxes. When an entry is complete use the 'Accept New URL' control button to save the new data into the list boxes.

Clicking on a data item in either list box causes the URL and its associated comment (if any) to be copied to corresponding text boxes. The contents of the text boxes may be edited and the changes can be saved. Clicking the 'Accept Edited Text' button makes the changes part of the file. Saving the file will make them permanent. You may add and delete URLs and comments by clicking the appropriate control buttons.

Combining Files

## About Netscape (and other browsers') "Bookmark" files

Netscape bookmark files are ordinary HTML files that contain additional information for use by Netscape, such as the date that the bookmark was saved.

Other browsers may also use the same, or a similar scheme (look for a file in the browser's directory hierarchy with a name like "bookmark.htm"). In those cases, the remarks about Netscape bookmark files, below, *may* apply to them as well.

You can use the *HTML Assistant Pro* URL Editor to extract the URLs from your Netscape (or, possibly, another browser's) Bookmark file(s) so that you can edit and combine them with URLs from other sources.

Simply use the URL Editor's File menu's "Extract URLs from HTML file..." option. This will bring up a file selector dialog. Simply find and select the Netscape Bookmark file, usually called "bookmark.htm", that you want to edit.

This will bring the URLs and associated text or comments into the URL Editor's list boxes, so that you may work with them.

## Combining Files

Files may be combined by selecting the "Append " option button (displayed between the two list boxes) when a file is to be opened and there is data already in the list boxes. The new data will be appended to the existing data.

Files are saved as HTML Assistant URL Files unless the 'Save as Cello Bookmark File...' option in the 'File' menu is selected.

## Direct Conversion of Files to HTML Text

By selecting the 'Autoconvert File to HTML' option in the 'URL' menu, *HTML Assistant Pro* permits you to automatically convert certain files that contain URLs (along with optional comments or descriptions) directly to HTML text in a new *HTML Assistant Pro* text editing window. This is a quick and easy way to build HTML files for editing, testing or browsing.

When a file type is selected (from the popout sub menu) you will be presented with a file selector box from which you may select a file to convert (or you may cancel the operation).

As soon as a file of the appropriate type is selected, the conversion process begins. In a few seconds when conversion is complete the URLs that were found in the file, now properly converted to HTML text separated by paragraph marks, are automatically inserted into a new text window for further editing or testing.

You can elect to convert data in the following types of files to HTML :

Cello Bookmark Files (\*.bmk files)

HTML Assistant URL Files (\*.url files)

Mosaic Initialization Files (\*.ini files)

NOTE: Netscape bookmark files are already HTML files. This may also be true of the bookmark files created by some other browsers.

## **Cello Bookmark files (\*.bmk files)**

Cello Bookmark files are used by the Cello browser for saving URLs so that browsed documents may be revisited. They have their own format and their file names usually have the suffix ".bmk".



## HTML Assistant URL Files (\*.url files)

HTML Assistant URL files are text files containing URLs and, possibly, descriptions or comments. Each record, or line, in the text file contains a URL optionally followed by a comma (,) and a description.

An uncommented line looks like this:

```
http://info.cern.ch/example.html
```

A commented line would look like:

```
http://info.cern.ch/example.html, Sample Page
```

Because *HTML Assistant Pro* can automatically convert these files into an HTML text document, they make an excellent way to store or 'capture' interesting looking URLs for later perusal.

If, for example, while browsing news groups, mailing lists or the World Wide Web itself, the URL for an interesting looking home page is presented, you can simply 'Copy' the URL from the document and 'Paste' it into a text file using Notepad (or *HTML Assistant Pro*). The URL should be saved WITHOUT quotation marks.

If you wish, you may enter a comma (,) after the URL (but before the "Enter" key is pressed), followed by a description or comment which will appear as the 'live' or 'clickable' item in the HTML file when viewed by a browser. If you don't enter a comment, *HTML Assistant Pro* will add a copy of the URL (for display as the 'live' selectable item) to the HTML file at conversion time.

Just be sure you press the "Enter" key at the end of each line. Make sure that the URL and its comment (separated by a comma) appear on one logical line. It may look like two or more lines if the editor window is narrow, but as long as the "Enter" key was pressed only at the end of the URL-Comment combination, the item is a single record and will be processed properly.

## Mosaic Initialization Files (Mosaic.ini)

The early Mosaic WWW browsers used an initialization file with the name, "Mosaic.ini" for storage of setup parameters.

Included in this file are lists of menu items that contain descriptions and URLs that can be executed by Mosaic when the descriptions on the menus are selected (or 'clicked').

Also included in the file is a 'Hot List' of user selected URLs gathered during browsing expeditions on the Web. These do not have descriptors associated with them.

The URLs in Mosaic initialization files are what is converted to HTML text by *HTML Assistant Pro*. If the URLs come from menu items their associated menu captions are displayed as the selectable on screen text. 'Hot List' items are displayed with a copy of the stored URL as the on screen text.

## What is the Automatic Page Creator?

The Automatic Page Creator supplied with *HTML Assistant Pro* permits very rapid creation of HTML text pages for use with WWW browser programs. Minimal knowledge of HTML is required, and an attractive screen with your own selected hypertext links can be created in a matter of seconds.

For those who have access to HTML servers, this makes it quite easy to create pages for others to view.

For active collectors of information on the World Wide Web, the Automatic Page Creator (as well as *HTML Assistant Pro's* other features) makes it possible (and easy) to organize links to information in ways that make sense to you. You can select links from different pages and files and combine them into a single page with headings and descriptions of your choice.

You can create HTML pages on your local drive and access them from your own 'Home Page' (which is simply another HTML page you created and keep on your hard drive). By making effective use of the File Editor/Builder, you can organize links to WWW data by subject, date or just about any way you like, because it is so easy to create a page containing the information that you want on it. This makes it easy to create presentation pages on any topic for which you have collected links.

Starting the Automatic Page Creator

Using the Automatic Page Creator

Building HTML Displays by Appending New Pages

## Starting the Automatic Page Creator

The easiest way to start *HTML Assistant Pro's* Automatic Page Creator is to 'click' on the button with the picture of the lightning bolt that is on the right hand end of the tool bar.

The Automatic Page Creator may also be started from the 'URL' menu by selecting the 'Automatically Create a Browser Page...' item.

Using either option will cause the Automatic Page Creator's screen to appear.

If you start the Automatic Page Creator by accident, simply 'click' the 'Exit' button to close it.

Using the Automatic Page Creator

## Using the Automatic Page Creator

The Automatic Page Creator's screen includes a box with a few instructions, some text boxes into which you may type information that you want to appear on your new browser page, and some button controls and check boxes.

The screenshot shows the 'HTML Assistant - WWW Page Creator' window. It has a title bar with a lightning bolt icon and standard window controls. The main area is divided into several sections: 1. 'HTML Document Title:' with a text box containing 'Created with HTML Assistant Pro, October 2, 1995' and a 'Help' button. 2. An 'Image...' button next to a text box. 3. 'Main Heading:' and 'Subheading:' text boxes, each with a 'Horizontal Line' checkbox to its right. 4. 'Descriptive Text:' with a large text area, a 'Use text from File ...' button, and a 'Horizontal Line' checkbox. 5. 'URL File:' with a text box, a 'Select URL File...' button, a 'Create URL File...' button, and radio buttons for 'Unnumbered List' (selected) and 'Numbered List'. A checked checkbox for 'Double Space' is also present. 6. A bottom section with a checkbox for 'Append to previous page', an 'Undo Last Append' button, a 'Create Browser Page' button with a lightning bolt icon, and an 'Exit' button.

For the most part the function of these controls is made clear by their adjacent descriptive labels.

The Page Creator uses the information that you enter to create a browser screen with a title, an in-line image, a heading, a subheading, some text (as much as you like), followed by a list of 'clickable' URL s (Uniform Resource Locators that serve as links to more information or other browser screens).

The list of URLs in the HTML page produced by Page Creator comes from a URL file -- either a previously created file, or one that can be created while you are using the Automatic Page Creator. If you want to have URL links automatically inserted into your file you must provide an HTML Assistant URL file for that purpose.

See "Edit/Build URL Files"

'Clicking' the large button with the lightning bolt on it will cause the HTML text for your screen to be created in an empty text box, and the Page Creator screen will disappear.

It's a good idea to create your own HTML pages and test them with a browser until you understand (and

see) what each textbox and button on the Page Creator screen does. You will have to save the text as a file before you can test it.

If you don't want to keep it, close the window with the HTML text in it and try again -- saving the new text to the same file as you did before (usually something like 'TEST.HTM').

Building HTML Displays by Appending New Pages

## Building HTML Displays by Appending New Pages

More complex HTML pages can be built by taking advantage of the Page Creator's "Append to previous page" feature.

The Page Creator keeps an internal copy of the last page it created. When the 'Append...' check box is selected it creates a new page which consists of the *stored* last page plus the new one. It copies the combined result into a *new* text window.

This means that if you edit the text box with the previous page in it, and then choose to append a new page, the edited contents of the text box *will not appear* when the new page is added to the old. The original, unedited text will be used, and the new page will be appended to that. This is necessary to ensure that the HTML markings on the original text remain syntactically correct so the new text can be properly inserted.

Because it ignores blank (empty) text boxes, Page Creator permits you to create your own page design by adding new material to the contents of the page it previously created. (Text in the 'Title' text box is ignored when a page is appended -- the original title is always used.) This means, for example, that you could create a page with headers and a list of URLs, and by appending a page with only descriptive text, end your browser page with that text. You can append pages as many times as you like, as long as the resulting text occupies less than 32K bytes.

If you don't like the result of an append page operation you may bring up the Page Creator window and use the 'Undo last append' command button to remove it.

Remember that each Create, Append or Undo operation opens a new text window. For that reason you should periodically delete the windows that are no longer useful. This keeps the screen uncluttered and conserves Windows resources. While this is sometimes inconvenient, it ensures that you have an opportunity to review and save your work at any stage.

## Using a WWW Browser for Testing

*HTML Assistant Pro* permits you to display your HTML annotated file (without leaving *HTML Assistant Pro*) with any browser that will permit the use of the name of a file on a local disk as a startup file.

If you click on the 'Test' button in an edit window or select the 'Test' option from the 'File' menu, you will be led through a process which includes selecting your test program and saving its name for future use.

The 'Test' function uses the last saved version of your file. If your file has changed since you saved it last, you will be given an opportunity to save it before testing.

Alternatively, you may select the 'Autosave Before Testing' feature in the 'Options' menu, and your file will be saved automatically.

The 'Test' button or menu option will be disabled if the active file is a new file and you have not yet saved it. You must always save a new file at least once before you can test it.

To test the program, *HTML Assistant Pro* simply runs the program you designate as your test program, with the full path and file name of the active edit file in the command line. Therefore, for the test to work, the browser you use must be able to use a local file specified on the command line as its startup file.

You also need to be concerned with where your test program expects to find configuration files, initialization data, etc. We have sometimes found it necessary to copy some or all of these files to *HTML Assistant Pro*'s home directory or to the directory where the files to be tested reside.

Microsoft Internet Explorer and Netscape as well as most other browsers, can be used as test programs. You can easily try new browsers as they become available.



## Using Dynamic Data Exchange (DDE) with a Browser

Use of DDE (Dynamic Data Exchange) with a test browser permits relatively smooth back and forth transitions from *HTML Assistant Pro* to the browser and vice versa. It eliminates the need to close and restart the test browser each time you want to test a file.

If you are using Netscape or Microsoft Internet Explorer as your test browser, *HTML Assistant Pro* gives you the option of using DDE (Dynamic Data Exchange) to send the browser the name of the HTML file you want to test. If the browser is not loaded, *HTML Assistant Pro* loads it. Once your browser is loaded and you do not close it, it will not have to be reopened whenever you want to test a file. Each time you select the item in the File menu for testing using DDE, or the "Test" button on a document window, the name of the file you are testing is sent to the browser and it becomes the active, topmost application.

With some browsers, when using DDE, your browser may not always refresh itself if your file has changed. That is, changes in your HTML text may not always be immediately seen when you invoke the browser. If that happens use the browsers "Reload document" (or equivalent) option to ensure that you are viewing the latest saved version of your HTML file.

Clicking on the small '*To editor*' box in the lower right corner of the screen will return you to *HTML Assistant Pro* (without closing the browser).

To set the DDE option on, go to the 'Options' menu and click on the use DDE option. You can tell it has been selected if it is checked. That's all there is to it. If you run into problems with DDE, go back to the Options menu and click it again to deselect it and return to the test mode without DDE.

As other browsers permit DDE, *HTML Assistant Pro*'s DDE capability will be updated to work with them.

# Overview

This section provides a complete reference to the HTML tags supported by HTML Assistant Pro. This includes the HTML extensions which have been introduced by Netscape and Microsoft, as well as HTML 3.0 tables.

Since HTML is hierarchical in nature, with elements being nested inside other elements, it is important to identify where an element can be used, and not just how it is used. This is provided in the descriptions below.

# Comments

**<!-- ... -->**

Comments can be included in an HTML document by surrounding the text with <!-- and -->. All text occurring between these two delimiters will be ignored by the HTML browser. Comments can appear anywhere text would normally be allowed.

*Example:*

```
<!-- Created using HTML Assistant Pro 2 -->
```

# Document Structure Elements

<HTML> ... </HTML> -- The HTML Document

<HEAD> ... </HEAD> -- Document Information

<TITLE> .. </TITLE> -- Document Title

<BASE...> -- Base URL

<ISINDEX> -- Searchable Document

<NEXTID> -- Counter for Automated Editors

<LINK> -- Relationship to Other Documents

<BODY> ... </BODY> -- The Displayed Text Body

## **<HTML> ... </HTML> -- The HTML Document**

*Can contain:* HEAD, BODY

Although not strictly required at present, an HTML document should always be enclosed by this pair of tags.

*Example:*

```
<HTML>
<!--The rest of the document, including all elements, appears here -->
...
</HTML>
```

## **<HEAD> ... </HEAD> -- Document Information**

*Can contain:* TITLE, BASE, ISINDEX, NEXTID, LINK, META

*Can be inside:* HTML

This element encloses the "head" of the HTML document. The head contains general information about the document and is not displayed when the page is rendered. A number of elements can appear in the head, in any order, but only the TITLE element is required.

*Example:*

```
<HTML>
<HEAD>
<!-- HEAD elements go here -->
...
</HEAD>
<!-- The rest of the document goes here -->
...
</HTML>
```

## **<TITLE> .. </TITLE> -- Document Title**

*Can contain:*    characters only

*Can be inside:* HEAD

This element specifies the title of the document. Every HTML document must have exactly one of these elements, accurately and concisely identifying the document. It does not effect how the document is rendered, but instead is normally used as the title of the window in which the document is displayed.

*Example:*

```
<HTML>
<HEAD>
<TITLE>Herbert D. Langley's Home Page</TITLE>
<!-- Other HEAD elements go here -->
...
</HEAD>
<!-- The rest of the document goes here -->
...
</HTML>
```

## <BASE...> -- Base URL

*Attributes:*        HREF="..."

*Can be inside:* HEAD

An optional element which specifies the base URL of the current document. It is used to help evaluate any relative URLs which are referenced in the document. This allows a document to be on one machine and the documents which it references using relative URLs to be on another. If the BASE element is absent, the browser assumes relative URLs identify documents on the same machine as the main document.

*Example:*

```
<HEAD>
```

```
...
```

```
<BASE HREF= "HTTP://host.somewhere.org/~herb/home.html">
```

```
...
```

```
</HEAD>
```



## <ISINDEX> -- Searchable Document

*Can be inside:* HEAD

This optional element identifies the parent document as a gateway script that allows searches. Its presence informs the browser program that the document can be examined using a keyword search, and that the browser should query the user for a search or query string.

*Example:*

```
<HEAD>
<ISINDEX>
...
</HEAD>
```

## **<NEXTID> -- Counter for Automated Editors**

*Attributes:*        N=number

*Can be inside:* HEAD

An optional element which indicates the "next" document to this one (as might be defined by a tool to manage HTML documents in series). It is not intended to be used directly by a user.

*Example:*

```
<HEAD>
```

```
...
```

```
<NEXTID N=50>
```

```
...
```

```
</HEAD>
```

## **<LINK> -- Relationship to Other Documents**

*Can be inside:* HEAD

*Attributes:* Same as A (anchor) element (see below)

Indicates the relationship between this document and other documents or objects. It is generally used only by HTML generating tools and at present is not fully supported by any browser.

## **<BODY> ... </BODY> -- The Displayed Text Body**

*Can contain:* Hn, P, HR, DIR, DL, MENU, OL, UL, ADDRESS, BLOCKQUOTE, FORM, PRE

*Can be inside:* HTML

Encloses the text and tags comprising the main body of the document. Whereas the tags contained in the header provide information about the document, the elements in the body are the document. These directly control how the document appears when rendered by a browser. Strictly speaking, text cannot appear by itself in the body; it is always part of some tag which determines how the text will appear.

*Example:*

```
<HTML>
```

```
<HEAD>
```

```
...
```

```
</HEAD>
```

```
<BODY>
```

```
<!-- The complete document body goes here -->
```

```
...
```

```
</BODY>
```

```
</HTML>
```

# Block Formatting

Block formatting elements typically apply to whole blocks of text rather than individual characters. In addition to possibly changing font size and style, these elements usually cause line breaks to be inserted or new paragraphs to start.

<Hn> -- Headings

<P> ... </P> -- Paragraph

<BR> -- Line Break

<ADDRESS> ... </ADDRESS> -- Addresses

<BLOCKQUOTE> ... </BLOCKQUOTE> -- Quotes

<PRE> ... </PRE> -- Preformatted Text

<HR> -- Horizontal Rule

## <Hn> -- Headings

*Can contain:* A, IMG, BR, EM, STRONG, CODE, SAMP, KBD, VAR, CITE, TT, B, I

*Can be inside:* BLOCKQUOTE, BODY, FORM

Six levels of heading are defined in HTML, from <H1> to <H6>. A Heading element implies all the font changes, paragraph breaks before and after, and white space necessary to render the heading. This information is defined in the browser itself and can be changed by the user, although typically the default values are used. The H1 heading is usually a large bold-faced font, representing the top-level heading, and each subsequent heading is normally an increasingly smaller font.

*Example:*

```
<H1>Top-level heading</H1>
```

Here is some text.

```
<H2>Second level heading</H2>
```

Here is some more text.

## **<P> ... </P> -- Paragraph**

*Can contain:* characters, character formatting, A, BR, IMG

*Can be inside:* BODY, BLOCKQUOTE, FORM, DD, LI, TD

This element is used to delineate text representing a paragraph. The closing element </P> is optional and is rarely used since a <P> implies the end of the current paragraph and the start of a new paragraph. The rendering of a paragraph is determined by settings in the browser and can be changed by the user, although the default is normally used.

*Example:*

<P>

This is the start of a new paragraph.

<P>This is another new paragraph.

## **<BR> -- Line Break**

*Can be inside:* A, ADDRESS, B, CITE, CODE, DD, DT, EM, Hn, I, KBD, LI, P, PRE, SAMP, STRONG, TT, VAR, TD

This tags causes a line break to be inserted in the displayed text.

*Example:*

This line would break<BR>right here.



## **<ADDRESS> ... </ADDRESS> -- Addresses**

*Can contain:* characters, character formatting, A, BR, IMG

*Can be inside:* BLOCKQUOTE, BODY, FORM, TD

The ADDRESS element is used to specify information such as addresses or e-mail signatures. The rendering of address text is defined by the browser and can often be changed by the user. Typically, it is italic and may possibly be indented. A new paragraph is implied by this element.

*Example:*

<P>You can reach me at <ADDRESS>herb@host.somewhere.org</ADDRESS>

## **<BLOCKQUOTE> ... </BLOCKQUOTE> -- Quotes**

*Can contain:* BLOCKQUOTE, Hn, UL, OL, DIR, MENU, DL, PRE, FORM, ISINDEX, HR, ADDRESS

*Can be inside:* BLOCKQUOTE, BODY, DD, TD, FORM, LI

The BLOCKQUOTE element is used to designate a block of text as a quote or citation. Browsers can be configured to render such text in various ways, such as indenting and/or italicizing the BLOCKQUOTE contents, and so on. An automatic paragraph is generated by this element. Note that BLOCKQUOTES can be nested.

## **<PRE> ... </PRE> -- Preformatted Text**

*Can contain:* Characters, character formatting, A

*Can be inside:* BODY, BLOCKQUOTE, FORM, DD, LI

This element presents blocks of text in a fixed-width (non-proportional) font, and so is suitable for text that has been formatted on screen. This element is the only element which preserves line breaks and character spacing in the original text. It can take an optional WIDTH attribute which specifies the maximum number of characters for a line. If it is not included, a width of 80 characters is assumed.

*Example:*

```
<PRE>
```

```
The following text will look  
    exactly like  
    this  
        when rendered.
```

```
</PRE>
```

## **<HR> -- Horizontal Rule**

*Can be inside:* BLOCKQUOTE, BODY, FORM, PRE

In addition to generating a line break, this element causes a horizontal line to be rendered on the screen, usually the full width of the screen. Netscape has enhanced this element by adding additional attributes to affect the appearance of the rule. See "Netscape Extensions" below.

# Character Formatting

Unlike block formatting elements, character formatting elements apply to individual characters. They are used to specify either the logical meaning or the physical appearance of marked text without causing a paragraph break and many render the marked text using the same format. Like most other elements, character-level elements include both opening and closing elements. The following characteristics apply to all character formatting elements.

*Can contain:* A, IMG, BR, EM, STRONG, CODE, SAMP, KBD, VAR, CITE, TT, B, I

*Can be inside:* A, ADDRESS, B, CITE, CODE, DD, DT, TD, EM, Hn, I, KBD, LI, P, PRE, SAMP, STRONG, TT, VAR

<B> ... </B> -- Boldface

<I> ... </I> -- Italics

<TT> ... </TT> -- Teletype

<EM> ... </EM> -- Emphasis

<STRONG> ... </STRONG> -- Strong Emphasis

<CODE> ... </CODE> -- Typed Code

<KBD> ... </KBD> -- Keyboard Input

<VAR> ... </VAR> -- Variable

<SAMP> ... </SAMP> -- Sample Text

<CITE> ... </CITE> -- Citation

## **<B> ... </B> -- Boldface**

This causes the marked text to be rendered in bold highlighting.

*Example:*

HTML is <B>really, really</B> easy.

This would be rendered as:

HTML is **really, really** easy.

**<|> ... </|> -- Italics**

This causes the marked text to be rendered in an italic font.

## **<TT> ... </TT> -- Teletype**

The TELETYPE element specifies that the marked text should be rendered in fixed-width (non-proportional) typewriter style font.



## **<EM> ... </EM> -- Emphasis**

The EMPHASIS element indicates typographic emphasis, typically rendered as italics.

## **<STRONG> ... </STRONG> -- Strong Emphasis**

This element marks a block of text for strong emphasis, typically rendered in boldface.

## **<CODE> ... </CODE> -- Typed Code**

The Code element indicates an example of code and is typically rendered in a fixed-width font. Unlike the PRE element, this does not preserve line breaks and character spacing. Consequently, for large code fragments, the PRE element should be used instead of CODE.

## **<KBD> ... </KBD> -- Keyboard Input**

The KBD element marks a block of text as keyboard input and is typically rendered in a fixed-width typewriter font.

## **<VAR> ... </VAR> -- Variable**

Text marked with this element is intended to represent a program variable name. It is usually rendered as italic.

## **<SAMP> ... </SAMP> -- Sample Text**

The SAMPLE element marks a block of text as a sequence of literal or *sample* characters. It is typically rendered in a fixed-width typewriter font.

## **<CITE> ... </CITE> -- Citation**

This element specifies a citation and is typically rendered as italics.

# List Elements

<LI> ... </LI> -- List Item

<OL> ... </OL> -- Ordered List

<UL> ... </UL> -- Unordered List

<DIR> ... </DIR> -- Directory List

<MENU> ... </MENU> -- Menu List

<DL> ... </DL> -- Definition List

<DT> ... </DT> -- Definition Term

<DD> ... </DD> -- Definition Description



## **<LI> ... </LI> -- List Item**

*Can contain:* characters, character formatting, A, BR, IMG, DIR, DL, MENU OL, UL, BLOCKQUOTE, FORM, P, PRE

*Can be inside:* DIR, MENU, OL, UL

The LI tag marks an element in a list, of which there are several varieties, described below. A list element can contain characters and character formatting tags, as well as hypertext anchors. Lists can also be nested.

## **<OL> ... </OL> -- Ordered List**

*Can contain:* LI

*Can be inside:* BLOCKQUOTE, BODY, DD, FORM, LI

The Ordered List element is used to present a numbered list of items. A browser indicates ordering by numbering the items, assigning them ascending numbers, letters, and so on. Each item in a list is identified with an LI element, described below. Netscape has enhanced this element by adding several attributes; see the section on "Netscape Extensions" for more details.

*Example:*

```
<OL>
<LI>This is the first item of the list. It
automatically wraps and indents when rendered.
<LI>This is the second item of the list. Note
how its number is automatically increased.
<LI>This is the third item in the list.
</OL>
```

This would appear similar to the following in most browsers:

1. This is the first item of the list. It automatically wraps and indents when rendered.
2. This is the second item of the list. Note how its number is automatically increased.
3. This is the third item in the list.

## <UL> ... </UL> -- Unordered List

*Can contain:* LI

*Can be inside:* BLOCKQUOTE, BODY, DD, FORM, LI

This element defines an unordered list, where each element of the list is marked by a special symbol such as a bullet. Netscape has enhanced this element by adding several attributes; see the section on "Netscape Extensions" for more details.

*Example:*

```
<UL>
<LI>This is the first item of the list. It
automatically wraps and indents when rendered.
<LI>This is the second item of the list. Note
how it is also marked with a bullet.
<LI>This is the third item in the list.
</UL>
```

This would appear similar to the following in most browsers:

This is the first item of the list. It automatically wraps and indents when rendered.

This is the second item of the list. Note how it is also marked with a bullet.

This is the third item in the list.

## **<DIR> ... </DIR> -- Directory List**

*Can contain:* LI

*Can be inside:* BLOCKQUOTE, BODY, DD, FORM, LI

This element is used to present a list of short items containing up to 20 characters each. Some browsers may arrange the items in the list as columns across the screen instead of one on top of the other.

*Example:*

```
<H2>HTML List Types</H2>
<DIR>
<LI>Ordered<LI>Unordered
<LI>Directory<LI>Definition<LI>Menu
</DIR>
```

This would appear similar to the following in most browsers:

### **HTML List Types**

Ordered	Unordered	Directory
Definition	Menu	

## **<MENU> ... </MENU> -- Menu List**

*Can contain:* LI

*Can be inside:* BLOCKQUOTE, BODY, DD, FORM, LI

A menu list is a list of short items, each preferably less than one line in length. The menu list style is often rendered similar to an unordered list but uses a more compact style.

## **<DL> ... </DL> -- Definition List**

*Can contain:* DT, DD

*Can be inside:* BLOCKQUOTE, BODY, DD, FORM, LI

*Attributes:* COMPACT

This list type is used to create lists such as would be used for glossaries, consisting of terms and accompanying descriptions. It takes a single optional attribute, COMPACT, to specify that the list should be rendered in a more compact style than normal. For an example, see the description of the DD tag below.

## **<DT> ... </DT> -- Definition Term**

*Can contain:* characters, character formatting, A, BR, IMG

Can be inside: DL

The term part of an entry in a definition list is identified by this tag. The text supplied should be short, consisting of a only few words. The terminating </DT> is optional since it is either implied by the start of another <DT> or <DD> element or by the </DL> ending the definition list.

## <DD> ... </DD> -- Definition Description

*Can contain:* characters, character formatting, A, BR, IMG, DIR, DL, MENU OL, UL, BLOCKQUOTE, FORM, P, PRE

*Can be inside:* DL

The description part of an entry in a definition list is identified by this tag. It should always be preceded by DT element supplying the term being described. The text supplied can be long, consisting of multiple paragraphs if needed. The terminating </DD> is optional since it is either implied by a <DT> element starting another list entry or by the </DL> ending the definition list.

*Example:*

```
<DL>
<DT>Indoor soccer
    <DD>A form of soccer played indoors,
        usually in an arena. It is popular in
        the United States and Canada.
<DT>Jai alai
    <DD>A Latin-American game like handball,
        played with a curved basket fastened to
        the arm, for catching and hurling the
        ball.
</DL>
```

This example would be rendered similar to the following:

Indoor soccer

A form of soccer played indoors, usually in an arena. It is popular in the United States and Canada.

Jai alai

A Latin-American game like handball, played with a curved basket fastened to the arm, for catching and hurling the ball.



# Links

## <A...> ... </A> -- Hypertext Anchors

*Attributes:*        HREF, NAME

*Can contain:*     characters, character formatting BR, IMG

*Can be inside:*   ADDRESS, Hn, P, PRE, DT, DD, LI, character formatting

This element marks text that is the start and/or destination of a hypertext link. It accepts two attributes, HREF and NAME, one of which must be present. (Other attributes are officially defined for this element but few browsers support them and are not described here.) The text is usually highlighted in some way by browsers to identify it as hypertext link, typically underscored and in a different color. The HREF and NAME attributes are described below:

### **HREF="URL"**

If the HREF attribute is present, the text between the <A> and </A> tags becomes hypertext and becomes associated with the URL provided with this attribute. If this hypertext is selected by users in a browser, they are moved to another document, or to a different location in the current document, as defined by the associated URL.

*Example:*

Our home page

```
<A HREF="http://www.herb.com/home.html">Herbert Associates</A> provides  
complete information about our services.
```

This might be rendered as follows:

Our home page **Herbert Associates** provides complete information about our services.

In this example, selecting "Herbert Associates" takes the user to a document called home.html located at http://www.herb.com. If the HREF refers only to a file without a corresponding network address, the file is normally located on the same host as the current document. It is also possible to transfer to a specific location within a document using HREF strings of the form "location#identifier", where *identifier* refers to a named anchor in the file specified by *location*. If the location portion is blank, the link refers to a point in the same document.

*Example:*

```
Click <A HREF="#details">here </A> for more details.
```

When the user clicks on the string "here", the browser jumps to the anchor named "details". Anchors are named using the NAME attribute.

### **NAME="string"**

If the NAME attribute is used, an anchor can act as the target of a link. This is typically used to transfer the user to a different location within the current document. The string supplied with the NAME attribute effectively gives the anchor a name or identifier. Identifiers are arbitrary strings but must be unique within a given HTML document. These identifiers can be referred to by other anchors to indicate the destination of the link.

*Example:*

Click `<A HREF="#details">here</A>` for more details.

...

`<A NAME=details>Additional Details</A>`

In this case, clicking on "here" would cause the browser to transfer its focus to the anchor called "details".

# Images

## <IMG...> -- In-line Images

*Attributes:* ALIGN, ALT, ISMAP, SRC

*Can be inside:* A, ADDRESS, B, CITE, CODE, EM, I, KBD, SAMP, STRONG, TT, VAR, Hn, P, PRE, DT, DD, LI, TD, character formatting

This element is used to insert a graphics image, such as a ".GIF" file, into a document. Several attributes are defined, as described below:

### **SRC="URL"**

This is a mandatory attribute specifying the URL of the image file.

### **ALIGN=*alignment***

This optional attribute specifies the alignment of the image with respect to the neighboring text. Possible values are TOP, MIDDLE, and BOTTOM.

### **ALT="text alternative"**

The ALT attribute is optional and is used to specify a text string which is displayed on the screen in browsers that do not support graphics or can be configured to delay the displaying of graphics.

### **ISMAP**

Another optional element which identifies the image as a clickable map. This allows the user to click on the image and have different regions of the image cause different actions. When the user clicks on a map, the browser transmits the mouse coordinates where the click occurred to the server which uses them to determine how to respond.

# Forms

<FORM...> ... </FORM> -- Fill-In Forms

<INPUT...> -- Form Field Type

<TEXTAREA...> ... </TEXTAREA> -- Text Block Input

<SELECT...> ... </SELECT> -- Multiple Options

<OPTION...> ... </OPTION> -- Options for SELECT

## <FORM...> ... </FORM> -- Fill-In Forms

*Attributes:* ACTION, ENCTYPE, METHOD

*Can contain:* INPUT, SELECT, TEXTAREA, Hn, p, HR, DIR, DL, MENU, OL, UL, ADDRESS

*Can be inside:* BLOCKQUOTE, BODY, DD, LI, TD

The FORM element is used to delimit a data input form. It is used to create forms containing check boxes, radio buttons, text input fields, and buttons. The browser is used solely to collect information for the form; it must be sent to a program on the server to be processed. The attributes defined for the form tag are:

### **ACTION="URL"**

A required attribute used to specify the URL to which the forms contents are sent when submitted for processing. This is usually either a program on an HTTP server or an e-mail address specified using a "mailto" URL.

### **METHOD=GET or POST**

This is an optional attribute. It is used to specify how the form data will be sent to the HTTP server when it is submitted for processing.

## <INPUT...> -- Form Field Type

*Attributes:* ALIGN, CHECKED, NAME, MAXLENGTH, SIZE, SRC, TYPE, VALUE

*Can be inside:* FORM, any nonempty element allowed inside a form

This tag is used to specify a field in a form whose contents may be edited by the user. The type of field and its characteristics are specified through various attributes:

### **TYPE=***field-type*

This defines the type of the input field. Possible values for *field-type* are CHECKBOX, RADIO, TEXT, SUBMIT, RESET, IMAGE, or HIDDEN.

### **VALUE=***"field-value"*

This attribute specifies different things for different field types. For a text or hidden field, it represents the value of the field; for a check box or radio button, it specifies the value to be submitted with the form if the field is checked; for a RESET or SUBMIT button, it represents the label for the button.

### **SRC=***"source-file"*

Specifies the location of the image file when TYPE=IMAGE is used.

### **CHECKED**

For use with check boxes and radio buttons to set the initial value of the field to checked (on).

### **SIZE=***n*

Indicates the size of a text field in characters.

### **MAXLENGTH=***n*

Specifies the maximum number of characters that can be entered into a text field.

### **ALIGN=***alignment*

This attribute specifies the alignment of the images within forms with respect to the neighboring text. Possible values are TOP, MIDDLE, and BOTTOM.

## **<TEXTAREA...> ... </TEXTAREA> -- Text Block Input**

*Attributes:*       ROWS, COLS, NAME

*Can contain:*     characters only

Can be inside:    FORM

Used to create a multi-line text input field. It has three required attributes which provide the following information:

### **ROWS=*n***

Specifies the height of the text area in characters.

### **COLS=*n***

Specifies the width of the text area in characters.

### **NAME=*name***

Specifies the variable name associated with the text area contents. It is used by the script which processes the form when it is submitted to the server.

## **<SELECT...> ... </SELECT> -- Multiple Options**

*Attributes:*        SIZE, MULTIPLE, NAME

Can contain:    OPTION

Can be inside:  FORM

This form element is used to create a menu or scrolling list of multiple items from which the user can choose when the form is presented. Attributes determine the characteristics of the SELECT item:

### **SIZE=*n***

Specifies the displayed size of the list or menu in lines. If omitted, a value of 1 is assumed.

### **MULTIPLE**

Optional attribute to specify if the user can select multiple items from the SELECT list at one time or a single item.

### **NAME=*name***

Specifies the variable name associated with the SELECT element. It is used by the script which processes the form when it is submitted to the server.



## <OPTION...> ... </OPTION> -- Options for SELECT

*Attributes:* VALUE, SELECTED

*Can contain:* characters only

*Can be inside:* SELECT

Specifies an entry in a SELECT element. The trailing </OPTION> tag is optional since the end of an option is implied by another option element or by the end of the select element. A series of option elements defines the list which is presented by the SELECT element when the form is rendered. An option has two optional attributes:

### **SELECTED**

Indicates that this option is initially selected.

### **VALUE="value"**

When present, indicates the value to be returned if this option is chosen. The returned value defaults to the contents of the OPTION element.

## **Frames**

Most current browsers now support a new type of HTML document called a *frame* document. The tags associated with this kind of document are described in this section. This is an area OF HTML that is still evolving and some of this discussion may be inaccurate or incomplete by the time you read this.

## <FRAMESET ...> ... </FRAMESET> -- Defining Framesets

*Attributes:* COLS="col-def", ROWS="row-def"

*Can contain:* FRAME, FRAMESET, NOFRAMES

*Can be inside:* HTML

In a frame document, the BODY element is replaced by a FRAMESET element. Its purpose is to define the layout of window frames within the browser window. Each frame is a distinct area of the main viewing window and each can contain its own separate HTML document. FRAME elements are used to define the content of a frame region. FRAMESETS can be nested to allow a viewing window to be divided into an arbitrary number of rows and columns.

### **COLS="col-def" ROWS="row-def"**

These attributes are used to specify the number of columns/rows into which the indicated frame is divided. Its value is a comma-separated list defining the number and size of the columns/rows making up this frame. Each value in the list takes on one of these forms:

*n* Specifies the absolute frame size, in pixels.

*n%* Specifies the frame size as a percentage of the total available width/height.

*n\** Specifies the size as a free-floating, relative value. All space remaining after allocating fixed width or percentage frames is divided among all the relative-sized frames. An optional integer preceding the asterisk defines the weight given to a column. For example, the string "2\*, 3\*, \*" would allocate 2/6 of the remaining space to the first frame, 3/6 to the second frame, and 1/6 to the third frame.

*Example:*

```
<FRAMESET ROWS="90, 3*, *, 100">
```

This defines four frame rows, with the top row being 90 pixels high and the bottom (fourth) row 100 pixels high. The second row is defined to be three-quarters of the remaining height and the third frame is one-quarter of the remaining height.

*Example:*

```
<FRAMESET COLS="80%, 20%">
```

This defines two frame columns., the first one 80 percent of the width of the window and the second 2 percent.

## **<FRAME ...> -- A Frame within a Framesets**

*Attributes:*      MARGINWIDTH, MARGINHEIGHT, NAME, NORESIZE, SCROLLING, SRC

*Can be inside:* FRAMESET

This element defines the properties of each of the frames declared by the FRAMESET element.

**MARGINWIDTH="n"** (optional)

Specifies the width in pixels of the left and right margins for the frame. Must be greater than 1. If omitted, the browser will determine an appropriate margin width.

**MARGINHEIGHT="n"** (optional)

Specifies the width in pixels of the top and bottom margins for the frame. Must be greater than 1. If omitted, the browser will determine an appropriate margin width.

**NAME="name-string"** (optional)

Used to assign a name to a frame, which can be referenced by the TARGET attribute of the A and BASE elements to direct documents or data to this named frame. If NAME is omitted, the frame name is undefined.

**NORESIZ**E (optional)

Informs the browser that the frame size is fixed and cannot be modified by the user.

**SCROLLING="yes", "no", "auto"** (optional)

Specifies what kind of scrolling, if any, is defined for the frame. "Yes" means that the frame must always have scrollbars and "no" means the frame should never have scrollbars. If "auto" is specified, the browser decides automatically if scrollbars are needed based on the size of the document being displayed and the size of the frame in which it is being displayed. The default value is "auto".

**SRC="URL"** (optional)

Specifies the URL of the HTML document to be displayed within the frame. If omitted, the frame is initially left blank.

## **<NOFRAMES> ... </NOFRAMES>**

*Can contain:*    BODY

*Can be inside:* FRAMESET

The NOFRAMES element is used to support browsers which do not recognize the FRAMESET or FRAME tags. Such browsers will ignore the FRAME tags and instead display the code defined within the NOFRAMES tag (which is also not recognized by the browser). FRAMES aware browsers will process the frames tags as expected and will ignore anything defined within the NOFRAMES tag pair.

**Example:**

```
<HTML>
<HEAD>
<TITLE>Sample Frames Document</TITLE>
</HEAD>
<FRAMESET ROWS="50%, 50%">
  <NOFRAMES>
    <BODY BACKGROUND="brooknorth.gif">
      <H1 ALIGN="center">Frames Not Supported!</H1>
      <P ALIGN="center"><EM>If you are reading this text,
your current browser does not support the new FRAMES
tags which have been used to create this page. You
will need to upgrade your browser to properly view
this page. For a non-frames version of this page,
click <A HREF="brooknorth2.html">here</A>.
    </NOFRAMES>
    <FRAME SRC="products.html" NAME="frame1">
    <FRAME SRC="info.html" NAME="frame2">
  </FRAMESET>
```

# Tables

This section describes the proposed TABLE element of HTML 3.0. Some of the features and attributes may not be supported by all browsers.

<TABLE...> ... </TABLE> -- Table Wrapper

<CAPTION...> ... </CAPTION> -- Table Caption

<TH...> ... </TH> -- Table Heading

<TR...> ... </TR> -- Table Row

<TD...> ... </TD> -- Table Data

## **<TABLE...> ... </TABLE> -- Table Wrapper**

*Attributes:*        BORDER, CELSPACING, CELLPADDING, WIDTH

*Can contain:*     CAPTION, TH, TD, TR

*Can be inside:*   BLOCKQUOTE, BODY, FORM DD, LI

This element is used to encapsulate a table definition. Other table elements will be ignored if they don't appear inside this pair of tags. A number of attributes define how the table will appear:

### **BORDER or BORDER=*size***

Specifying this attribute causes borders to be drawn around all cells of the table. By default, tables have no borders. If the optional size field is included, the width of the border, in pixels, can be specified. Otherwise, the browser's default border size is used.

### **CELLSPACING=*value***

This specifies the amount of space inserted between individual cells in a table. Netscape uses a default cell spacing of two pixels.

### **CELLPADDING=*value***

Specifies the amount of space between the border of a cell and its contents.

### **WIDTH=*value or percent***

Used to specify the width of the table as either an absolute width in pixels or a percentage of the document (window) width.

## **<CAPTION...> ... </CAPTION> -- Table Caption**

Attributes:      ALIGN

*Can contain:*    characters, character formatting, A, BR, IMG

Can be inside: TABLE

The CAPTION element specifies the table caption or title. It should appear inside a <TABLE> definition but not inside row or cell definitions. Titles are always centered with respect to the table. The ALIGN attribute specifies whether the title should appear above the table or below the table, using ALIGN=TOP or ALIGN=BOTTOM respectively. It defaults to TOP if not specified.

## **<TH...> ... </TH> -- Table Heading**

*Attributes:* COLSPAN, ROWSPAN, ALIGN, VALIGN, NOWRAP

*Can contain:* characters, character formatting, A, BR, IMG, TABLE

*Can be inside:* TABLE

This element is used to specify a heading for an individual row or col of a table. A number of attributes can be used to define how the heading is presented:

### **ROWSPAN=*n***

Specifies how many rows of the table this heading cell should span. The default is one.

### **COLSPAN=*n***

Specifies how many columns of the table this heading cell should span. The default is one.

### **ALIGN=left, right, or center**

Controls how the heading text is aligned in the cell in which it appears.

### **VALIGN=top, middle, bottom, or baseline**

Controls whether the header text is aligned to the top of the cell, the bottom of the cell, or vertically centered within the cell.

### **NOWRAP**

Prevents the header text from being broken into multiple lines to fit in the cell. Instead, the width of the column expanded so that the header text can fit in one line.

## **<TR...> ... </TR> -- Table Row**

*Attributes:*      ALIGN, VALIGN

Can contain:    TD

Can be inside: TABLE

Specifies a row definition in a table. The number of rows in a table is exactly specified by how many <TR> elements are contained within it, irregardless of cells that may attempt to use the <ROWSPAN> attribute to span into nonspecified rows. A number of attributes are defined to control how the row is presented:

**ALIGN=left, right, or center**

Controls how the text in each cell of the entire row is aligned horizontally.

**VALIGN=top, middle, bottom, or baseline**

Controls whether the text in each cell of the entire row is aligned to the top of a cell, the bottom of the a cell, or vertically centered within a cell.



## **<TD...> ... </TD> -- Table Data**

*Attributes:* COLSPAN, ROWSPAN, ALIGN, VALIGN, NOWRAP

*Can contain:* characters, character formatting, A, BR, IMG, TABLE

*Can be inside:* TR

This element is used to specify a heading for an individual row or col of a table. A number of attributes can be used to define how the heading is presented:

### **ROWSPAN=*n***

Specifies how many rows of the table this cell should span. The default is one.

### **COLSPAN=*n***

Specifies how many columns of the table this cell should span. The default is one.

### **ALIGN=left, right, or center**

Controls how the text in each cell of the entire column is aligned horizontally.

### **VALIGN=top, middle, bottom, or baseline**

Controls whether the text in each cell of the entire column is aligned to the top of a cell, the bottom of the a cell, or vertically centered within a cell.

### **NOWRAP**

Prevents the lines within this cell from being broken into multiple lines. Instead, the width of the column expanded so everything can fit in one line.

# Netscape Extensions

Netscape has introduced a number of enhancements to HTML, mostly based on those proposed in HTML 3.0, but some features are unique to Netscape. The extensions are in the form of new tags as well as additional attributes to existing tags. These are described below.

<CENTER> ... </CENTER> -- Centering Text

Centering Paragraphs and Headers

In-line Images

Horizontal Rules

Font Sizes

Lists

Line Breaks

Backgrounds and Text Colors

Miscellaneous

## **<CENTER> ... </CENTER> -- Centering Text**

Used to center a block of text on the page. It can include paragraphs, headings, and so on.

## Centering Paragraphs and Headers

The <P> and <Hn> tags have been extended to recognize an additional attribute:

**ALIGN=CENTER**

This is the equivalent to the <CENTER> tag on individual paragraphs and headings.

## In-line Images

The <IMG> tag has had the following new attributes defined:

**VSPACE=*n***

Specifies the space in pixels between the image and the text above or below it.

**HSPACE=*n***

Specifies the space between the image and text to its left or right.

**WIDTH=*n* and HEIGHT=*n***

Specifies the width and height of the image in pixels. If the width and height values specified do not match the image's actual size, the image is scaled to fit.

**BORDER=*n***

Causes the image to be rendered with a surrounding border *n* pixels in width.

**LOWSRC="*URL*"**

This attribute can be used to load a smaller or lower resolution image compared to that specified by the SRC attribute. These alternate images are loaded when the page is first rendered. The SRC images are only loaded after all LOWSRC images have been loaded.

## Horizontal Rules

The horizontal rule tag <HR> has been given the following new attributes:

**SIZE=*n***

Specifies the thickness of the rule line, in pixels.

**WIDTH=*n***

Specifies the width of the rule, in pixels.

**ALIGN=left, right, or center**

Indicates how the rule will be aligned on the page.

**NOSHADE**

Netscape normally renders rules so that they have a drop shadow appearance. This attribute causes a rule to be displayed as a solid black line.

## Font Sizes

Two tags for specifying the font of rendered text have been defined:

**<FONT...> ... </FONT>**

Specifies the size of the font for the enclosed text via the SIZE attribute:

**SIZE=*n***

*n* is a value from 1 to 7 indicating the size of the font. 3 is the default value. It can also be specified as a value relative to the current size using a + or -, for example <FONT SIZE=+2>.

**<BASEFONT SIZE=*n*>**

Changes the default size of the font. The initial default is 3. *n* can be a value from 1 to 7.

# Lists

## **<UL...> ... </UL> -- Unordered lists**

A type attribute has been added to the unordered list tag:

### **TYPE=disc, circle, or square**

This controls the type of bullet used to mark list items.

## **<OL...> ... </OL> -- Ordered lists**

Two additional attributes have been defined for ordered lists:

### **TYPE=A, a, I, I, or 1**

This controls the type of numerals used to mark list items.

### **START=*n***

Specifies the number from which to start the list. The default is 1.

## **<LI> -- List item**

Two new attributes have been defined for list items:

### **TYPE=disc, circle, square, A, a, I, I, or 1**

Specifies the type of bullet used to mark this one list item.

### **VALUE=*n***

The numeric value this list item should have. It affects this item and all items below it in OL lists.



## Line Breaks

### **<BR CLEAR=left, right, or all>**

The CLEAR attribute has been added to the <BR> line break attribute to control how text which is aligned with images behaves when a line break tag is encountered. Without any value, the CLEAR attribute causes a new line to start below the image, back to the original margin. A value of **left** means to break to a clear left margin, **right** to a clear right margin, and **all** to a full clear line.

### **<NOBR> ... </NOBR>**

Prevents the text within the <NOBR> tag from being wrapped to the next line.

### **<WBR>**

Causes the text to be wrapped at this point only if necessary.

# Backgrounds and Text Colors

The following attributes have been added to the <BODY> tag:

## **BACKGROUND="URL"**

The purpose of this attribute is to specify a URL pointing to an image that is to be used as a background for the document. In Netscape, this background image is used to tile the full background of the document-viewing area.

## **BGCOLOR="#rrggbb"**

Changes the color of the background without having to specify a separate image that requires another network access to load. The "#rrggbb" value is a hexadecimal red-green-blue triplet a color as derived from standard RGB color tables.

## **TEXT="#rrggbb"**

Changes the color of all the normal text in a document (anything that isn't a link).

## **LINK="#rrggbb"; VLINK="#rrggbb"; ALINK="#rrggbb"**

Used to control the coloring of link text. VLINK stands for visited link, and ALINK stands for active link. The default coloring of these is: LINK=blue, VLINK=purple, and ALINK=red.

## Miscellaneous

**<BLINK> ... </BLINK>**

Causes the enclosed text to blink. In some version of Netscape this feature can be turned off as some users find blinking text irritating.

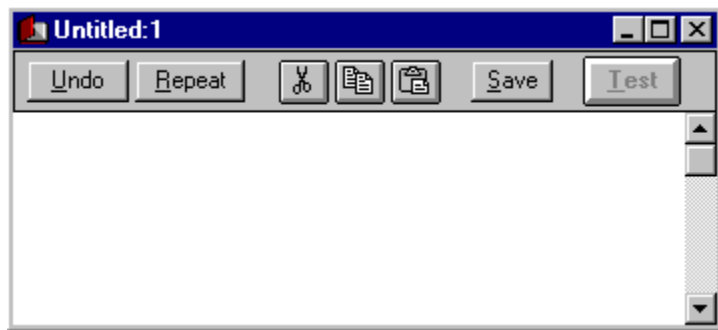
**&reg**

The registered trademark symbol ®.

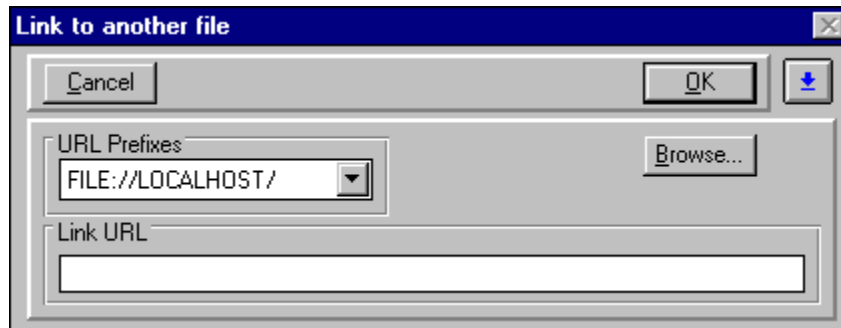
**&copy**

The copyright symbol ©.

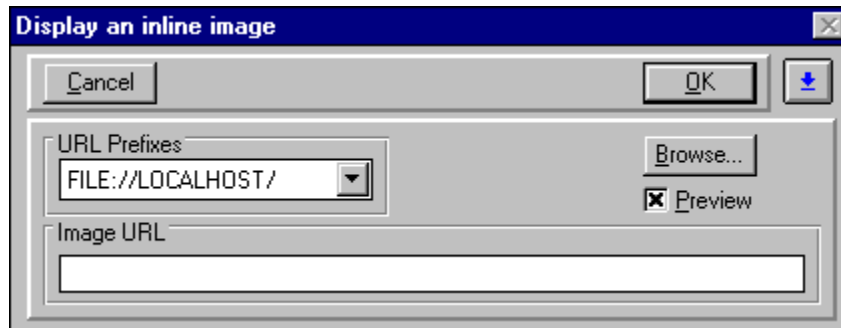
## Initial document window



## Link to file dialog



## Inline image dialog



## Sample form

### Product Ordering Form

Products to order:

Milk

Butter

Eggs

Juice

Apples

Oranges

Shipping Address:

Payment Method: ☐ VISA ☐ Master Card ☐ COD

Send Order

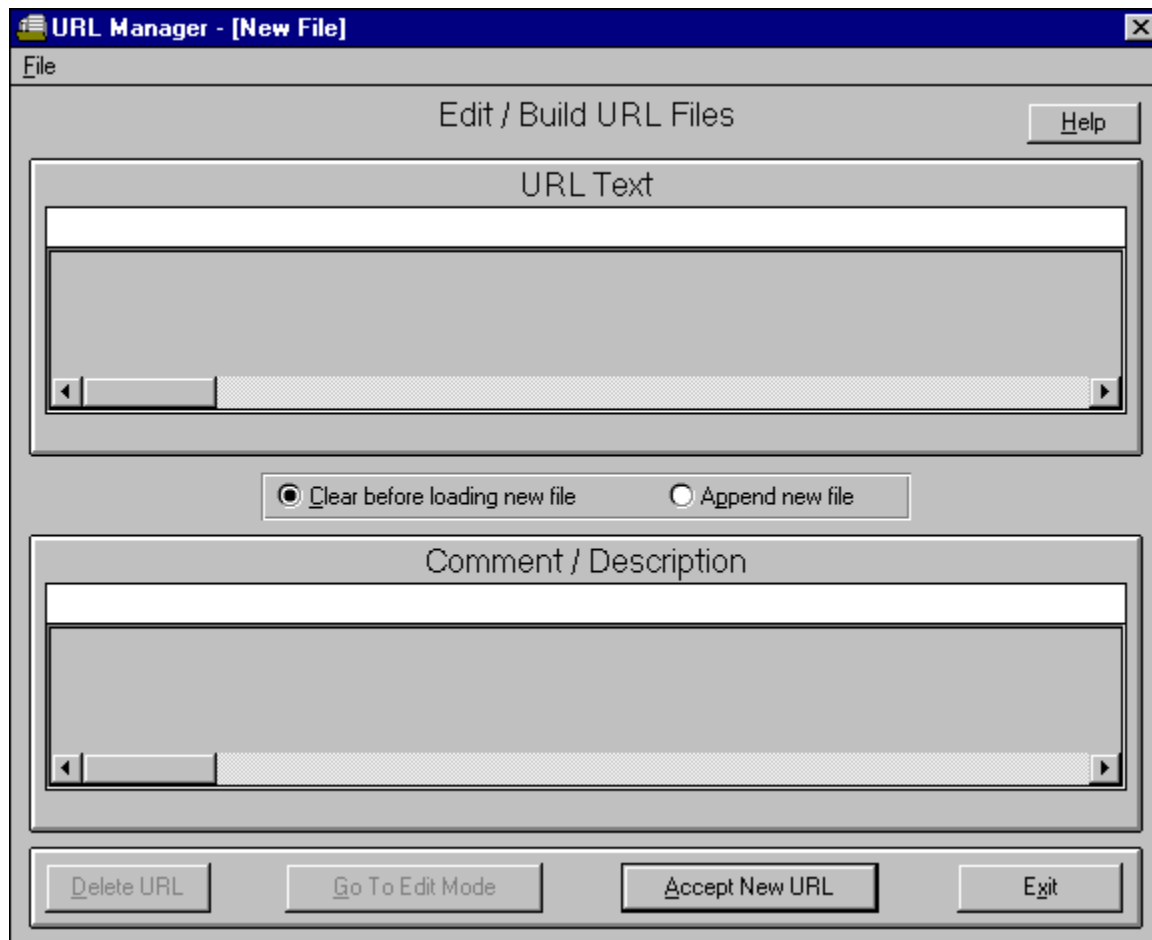
Reset

Sample table

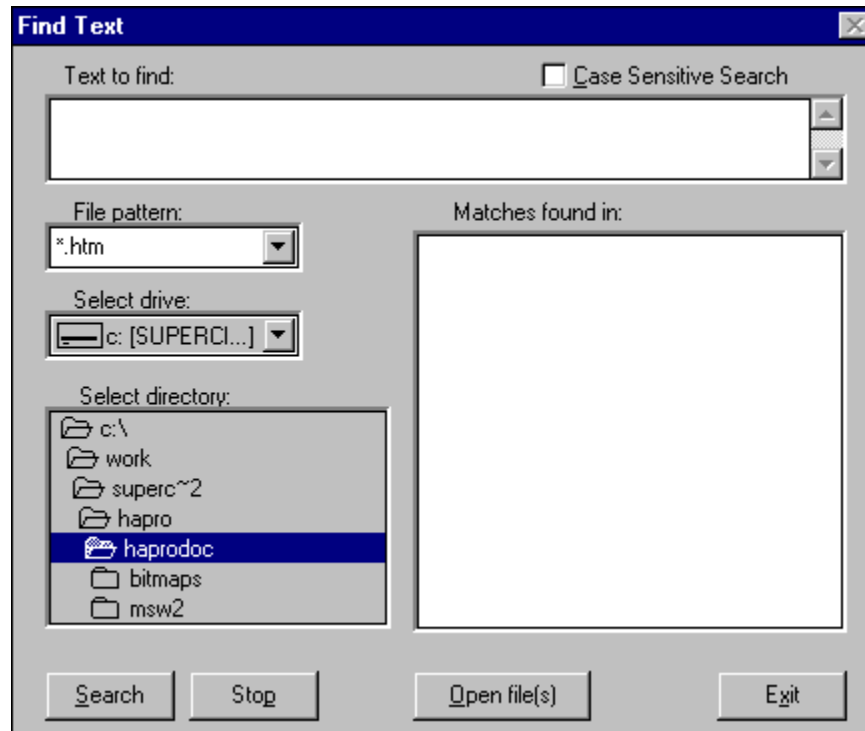
Product Inventory		
Product	Warehouse 1	Warehouse 2
Milk	8	12
Butter	19	17
Juice	23	20
Total	50	49
Grand Total	99	



## URL Manager window



## Text Search dialog



The image shows a 'Find Text' dialog box with a blue title bar and a close button. It contains several input fields and controls for searching text in files.

**Find Text**

Text to find:  ☐ Case Sensitive Search

File pattern:  ▼

Select drive:  ▼

Select directory:

- c:\
- work
- superc~2
- hapro
- haprodoc**
- bitmaps
- msw2

Matches found in:

## File Search dialog

