

Arachnophilia

[Definition of Arachnophilia](#)

-- P. Lutus, Port Hadlock, WA --

For a general overview of the Arachnophilia documentation, press the "Contents" button at the upper right of this window.

To find information about a specific topic, it is most efficient to use the "Find" feature. Click "Contents" above, choose the Find tab, and type in one or more words describing your interest.

Arachnophilia is the descendant of WebThing, which is the descendant of Apple Writer, but I digress. I learned a lot writing and distributing WebThing, the most important being "Don't write programs in Visual Basic." WebThing had a number of shortcomings, all of which are addressed in Arachnophilia.

Arachnophilia's purpose is to create Web pages. It does this in one of two general ways. **The easy way** is to drop a Rich Text Format (RTF) document onto the Arachnophilia program window and watch Arachnophilia turn it into a web page for you. **The not so easy way** is to write the HTML code yourself, which, although more work, produces the most professional-looking results.

Arachnophilia will help you create Web pages, no matter what method you choose. And, by just typing, *you can create new Arachnophilia commands, even entire toolbars, for HTML tags I haven't thought of* -- Arachnophilia's suite of commands is entirely under your control. You can load hundreds of documents at once (memory permitting) and search through them at once for particular words. You can preview your work on up to six browsers, thus assuring your pages will look good no matter what browser your visitors own.

I hope you like Arachnophilia. I also hope you will read about [CareWare](#), which is how you "pay" for Arachnophilia.

Web Resource Sites

[Frequently Asked Questions](#)

[www.arachnoid.com/
arachnophilia](http://www.arachnoid.com/arachnophilia)
www.microsoft.com,
www.netscape.com

This web site contains answers to the most often asked questions about Arachnophilia.

Use this link to get the most recent version of Arachnophilia.

Click these links to get the most recent versions of popular Web browsers.

Be sure to visit www.arachnoid.com, the home of [Arachnophilia](#).

Arachnophilia © Copyright 1996-1998 P. Lutus. All Rights Reserved.

Note: Function key F1 provides context-sensitive help throughout Arachnophilia.

Introduction

Arachnophilia is a full-featured HTML editor and workshop. With Arachnophilia, you can:

- Create Web pages and preview them on all available browsers.
- Interactively edit Web page resources and appearance.
- Import full-featured text, tables and outlines into your Web page, with table structure, indentation, font color, size and face preserved.
- Create interactive forms that launch programs to do things not possible in HTML alone.
- Organize a set of Web pages and associated graphics, sound files, and other resources.
- Maintain Web pages by examining their mutual references.

Arachnophilia will help you get started if you are a beginner, and it will help you organize larger projects as you acquire more experience. Arachnophilia will automatically copy resource files to your working directory as you select them, and will alert you if there are resources that are no longer being used.

Arachnophilia supports many of the HTML tags currently in use, and can be customized to include any specialized tags that the user desires, and any new tags that come into use in the future.

These help pages cannot teach you all you need to know about HTML to be a successful Web page writer, but there are many sources of detailed information that you should access. One very effective way to learn about HTML and Web pages in general is to browse the Web. There you will see any number of well-designed Web pages.

Another way to get more information is to search the Web itself for information about HTML. Example: Get online and access the Yahoo search engine at www.yahoo.com. Type HTML as your search string. Another good search engine is www.lycos.com. Also, bookstores and libraries have many excellent references on HTML.

These pages will help you with the particulars of Arachnophilia and its special features, and contain instructions on the use of the most common HTML tags.

The Internet and the Web

Computer networking has had a short and spectacular history. From the first effort to connect two computers together in the 1950s (rumor has it they deadlocked each other with "syntax error in line 12" messages) to the present, the value of sharing computer resources has become obvious. Regional networks became connected to each other, and eventually the Internet came into being.

There was always some way to navigate from place to place on networks. Earlier systems required you to type something cryptic at a keyboard, and to know something about UNIX. As time passed and more people tried to use the Internet for more kinds of things, the old ways were replaced with more intuitive graphic interfaces.

The most popular present method to interact with the Internet is by way of a Web Browser, a program that can read Web Pages that have been posted on the Internet. A modern Web page is a multimedia resource, containing text, images, sounds, and animations. Soon Web pages will contain links to objects that will offer you access to resources such as programs and information in ways not presently possible.

The "Web" in "Web page" refers to The World Wide Web, which is not so much a network as a way to interact with a computer network. The Web is in fact a combination of communication protocols that include a language called HTML, several kinds of programs called Browsers, and the Internet to connect it all together.

"Web pages" are written in HTML using a program like this one, generically known as a "Web Page Editor." After being written, Web pages are usually posted on a computer that is attached to the Internet. Then people who have network access to that computer can "read" the web page using their browser program.

Some Web pages contain graphics, sounds, and other things to increase their information content (translate: fun). But the most important single thing about a Web page is that it can contain links to other pages, resources, and locations. The behavior of these links is referred to as Hypertext, a method by which one can jump from place to place by activating the links.

In the original embodiment of Hypertext, one would move from paragraph to paragraph in a document using links, as in this help page. Now, using HTML, one may move from country to country by clicking a pointer device.

Individuals and institutions are therefore very interested in creating Web pages in the easiest possible way, and that is why Arachnophilia was written -- to make it easy to create your own Web page, add to it, and maintain it.

A good way to start is to browse the Web and see some of the pages that have been written by others, to get ideas about how you would like your page to look. Most browsers allow you to examine the original HTML code to see how a particular effect was created. This is a good way to learn advanced techniques.

The Basics of Web Pages

In the simplest terms, the World Wide Web is a collection of web pages and other resources located on Internet server computers connected to web page browsers located on individual machines also connected to the Internet. What makes it a Web is that the Web pages communicate the information desired by their authors, and also they connect readers to other resources by way of links. The links in Web pages can refer to another Web page, a program, or some other kind of file, or another way of interacting with the Web such as Telnet.

Web pages have two aspects. One aspect is how the page looks to a web browser, colorful and interactive. The other aspect is the actual HTML code that tells the browser what to do. The fun part of browsing is seeing the result aspect, but the other part, the code, has to come first.

Fortunately, HTML was designed to be as obvious as possible, consistent with its purpose. You will be writing HTML in a very short time from starting out. Writing *great* HTML code will take longer.

The overall process for creating a Web page is to get the resources together -- a web editor (you've already taken care of that) and a browser. If you don't already have a Web browser, I recommend Microsoft Internet Explorer. It is available on the Internet (possibly a catch-22 if you don't have any browser right now) and, like Arachnophilia, it is free (but see the CareWare section for a better explanation of "free").

Once you have these basic tools, you can follow this strategy:

- Make a plan. Perhaps you want to have just one page that says "Hi, world, this is me!", or perhaps you want to create something more ambitious, several pages, lots of graphics. A broad outline of your intentions is a good starting point.
- Select a data directory to store your work as it proceeds. Arachnophilia will prompt you to do this the first time you create a link between two file resources.
- Create the first page (if you are planning more than one). This page is usually the first thing a visitor sees, and is often no more than a table of contents. Usually this page is named "index.htm" to be consistent with how Internet servers locate and activate pages.
- Create the child pages, and link them to the index page. As you work, launch your Web browser (Arachnophilia allows you to do this while you work) to see the result of your efforts.
- Finally, after testing all the pages and jumping between them and to any outside resources you reference, you can upload your page(s) to the Internet. This is in some ways the most specialized part of the process, and you may have to ask your Internet Service Provider for help.

Arachnophilia has many features to simplify this process. Arachnophilia will remind you to set

up a working directory, then it will ask whether you want to automatically copy resource files to this directory as you work. Using this feature, you will automatically assemble all the resources for your Web page in one place, and you will also use what are called "relative links" to them, meaning the entire package will work as expected after being moved onto the Internet server. And, using a cross-reference feature, Arachnophilia will find and list resources files that are no longer being used by your pages, for easy removal.

With Arachnophilia and a browser installed on your system, you can develop web pages without actually being connected to the Internet. You can write, test, debug, and modify your pages before posting them on the Internet. And, perhaps best of all, you can add your favorite tags to Arachnophilia, tags you use a lot, or tags that don't exist as this is being written.

The Basics of HTML

The most important thing to realize about HTML is that it is changing with time, and that different browsers support different features. Therefore, if you want to be a serious Web page developer, you should collect enough browsers to be sure your page looks acceptable on all of them. Arachnophilia supports up to six browsers simultaneously, allowing you to verify changes as you work.

The next most important thing to know about HTML is that it was designed to be as easy and transparent as possible. After a short time, you will be able to look at an HTML script and predict more or less how it will look when viewed with a browser. Naturally, Arachnophilia does not require you to guess -- you can just press a button to move between the HTML code display and the browser display.

The most basic element of HTML is called a "Tag." HTML tags usually, but not always, come in pairs, an opening tag and a closing tag.

Tags are enclosed in the greater than - less than symbols "<" ">."

The *opening tag* of a pair of tags appears this way: <TAG>.

The *closing tag* of a pair appears this way: </TAG>. Notice the slash.

Tags can contain modifiers within the "<" ">" symbols. Example:
The information *inside* the tags tell the browser to do something. Example: <BGSOUND SRC="music.mid"> tells the browser to play some music (if the browser in question is Microsoft Internet Explorer).

The information outside the tags is printed as text. Example: the HTML code "This should be emphasized more than usual." will be shown by the browser as "This should be **emphasized** more than usual.

In this example, the browser understood the tag to mean "start printing in **boldface**." The browser then took the tag to mean "go back to the previous state." Most HTML tags are nested, which means they add to the formatting of any prior tags, even if the prior tags specify the same thing. This example of HTML:

```
"This is <B>an <B>important</B> point</B> to make."
```

Would most likely be rendered as:

"This is **an important point** to make."

because each pair of tags independently affect the text, and when the closing tag appears, the browser simply returns to whatever formatting had existed previously. This is important to understand and can save you from mysterious behavior in your Web pages.

NOTE: These structural issues are important to understand, but Arachnophilia will automatically convert text with many different styles, fonts, and colors for you, either by dragging text from another application using the Arachnophilia File Import methods, or by saving files in Rich Text Format, then simply loading them into Arachnophilia. Arachnophilia will automatically convert formatted text, tables, and outlines (indented structures) into the appropriate HTML code, as much as possible preserving the original appearance of your text (within the constraints imposed by HTML).

The most basic HTML document looks like this:

```
<HTML>

<Head>
<Title></Title>
</Head>

<Body>
|
</Body>

</HTML>
```

These pairs of tags are regarded as the basic structure of an HTML page, and all existing browsers support it. Arachnophilia will automatically create this structure for you when you open an HTML document from the File menu, or use the Structure/Lists toolbar.

The `<HTML></HTML>` tags simply identify this as an HTML document. The `<Head><Title></Title></Head>` tags basically allow you to place a title inside your document. This title is normally not displayed by the browser as part of the page, but appears at the top of the browser display on its title bar, and is also accessed by Internet search engines for descriptive information about your page.

Most Web Page development takes place within the `<Body></Body>` tag pair. For example, if you typed "This is my page" between the `<Body>` and `</Body>` tags, your browser would display that phrase.

There are many structural elements in HTML, such as tables, lists, and provisions for graphic images and other resources, but by far the most important to understand is the Hyperlink or link.

The most common HTML link looks like this:

```
<A href="destination">An interesting place</A>
```

The browser will show this link as [An interesting place](#) and if you click on it (in a browser), the browser will locate and load the thing described by "destination". In creating a link, the text you type between the `<A>` and `` tags tells the *reader* what the link is, and the

`href="destination"` part of the link tells the *browser* where to look for the resource.

Links are the way you navigate around the World Wide Web. A link might contain another of your own pages:

```
<A href="mystory.htm">Click here to read my story</A>
```

A link might refer to another location in the same Web page

```
<A href="#somebookmark">Click here to move to the bookmark</A>
```

Elsewhere in the Web page would be the name to which this bookmark refers, and to which the browser would move:

```
<A name="somebookmark">Thanks for coming around!</A>
```

A link can refer to a file you want to allow access to:

```
<A href="myprogram.exe">Click here to run my program</A>
```

Or:

```
<A href="database.mdb">Click here to download my data file</A>
```

And (actually a very common example) you can refer to something else on the World Wide Web:

```
<A href="http://www.microsoft.com">Click here to visit Microsoft</A>
```

You can also place a graphic image on your page:

```
<IMG SRC="myhouse.gif">
```

You can even use a graphic image as the clickable area for a hypertext link:

```
<A href="http://www.beachstuff.com"><IMG SRC="sandybeach.gif"></A>
```

In this example, the visible part of the tag is a picture instead of text. When the user clicks on the picture "sandybeach.gif", the browser jumps to the Web location identified by "http://www.beachstuff.com".

Hypertext links can refer to many kinds of things, including files, Web sites, locations in the same document, other pages you have written, they can even launch applications and send E-mail.

This is just the barest outline of what HTML can do. Once again, there are many resources available that can provide a full description and use of HTML, including resources on the Web itself.

How to make your own page (the easy way)

You always have the option of creating a page from scratch, by entering the HTML tags on your own. But, even with a lot of predefined tags, this takes a long time, and you have to learn a lot about HTML as you go along.

But there is an easier way -- Arachnophilia will allow you to import a word processing document that has been saved in Rich Text Format (RTF) and will then convert it automatically into a Web page.

Some word processing programs will export their documents in the RTF format, others won't, but you can still export them using one of the methods described here. Here are the ways to get your document from your word processor/spreadsheet/database to Arachnophilia:

- Simply save the document in the Rich Text Format (RTF) document type. Then open it in Arachnophilia (File ... Open File ... RTF File). When you do this, Arachnophilia will ask whether it should convert it into HTML format. Choose "Yes" and you are done.
- Open Windows Explorer and drag an RTF document icon onto the Arachnophilia program window (not onto an open document, but onto the background). In the same way as method (1), Arachnophilia will ask whether it should convert it into HTML format.
- For programs that do not support the RTF file format, open your source program and Arachnophilia at the same time. Open a new, blank RTF document in Arachnophilia (File ... New File ... RTF File). Move to your source program and select the block of text you are interested in. Drag this block over to the open RTF document window in Arachnophilia, and drop it. Then select the menu option Tools ... Convert RTF to HTML.
- Use the Windows clipboard and proceed as in method (3) above.

These methods apply to data sources such as spreadsheets and databases -- just use method 3 or 4 above if the program does not export RTF files.

Here are some suggestions to help Arachnophilia convert your document:

Always use *real* bulleted lists and *real* numbered lists, available in most word processing programs, instead of manually numbering a list of items. If you use the real versions of these features, Arachnophilia will create the HTML equivalent of these structures, which look great. If you simply manually number a list, it will not look nearly as good when viewed on a browser.

Avoid "outdenting" paragraphs in your document formatting -- Arachnophilia will interpret this as an outline. Instead, if you create a real outline, even one with multiple levels, Arachnophilia will translate it into HTML for you.

Here are some restrictions for this automatic conversion method:

Arachnophilia will automatically import tables and outlines, but cannot import any pictures that are included in your original document. These parts of your document have to be imported separately, using the HTML tags that are designed for that purpose.

Sometimes Arachnophilia will misinterpret a line with tab characters in it as a table row. If this is not what you intended and you want to prevent this behavior, choose Tools ... Options and select "Convert Tabs into Spaces." Remember later that you made this choice, because Arachnophilia won't create tables until the "Convert Tabbed Lines into Tables" option is enabled again.

In general, avoid the use of tabs, because this character is used in the RTF document format to identify table rows, and Arachnophilia relies on them for this purpose.

Be sure to review the [Arachnophilia File Import methods](#) for more information on these techniques.

How to make your own page (the ambitious, heroic way :))

Remember: you can just import many types of documents directly into Arachnophilia. The methods described here are for finer control over the outcome than the HTML translator can provide. See the [Arachnophilia File Import methods](#) for more information.

Now we will create a small Web Page, just to get your feet wet. At this point you should have installed Arachnophilia on your computer, have an icon available to run Arachnophilia, and be in a somewhat adventurous frame of mind.

(If this is hard to follow the first time, you might want to print this help screen so you don't have to keep referring to it on the computer screen.)

Start Arachnophilia (be sure to close any open browsers before you do), choose the internal browser if it is not already selected (Preview .. Select & Launch Browser ... Internal Browser), then select File ... New HTML Document (or press the leftmost toolbar button). A new document will appear on the screen, and a dialog box will appear with some choices. Just to keep this simple, for now just press "OK" in the dialog to move on.

If you have chosen the internal browser, you may want to arrange the windows in the program. I recommend horizontal tiling. Use the menu command Window ... Tile Horizontally. See how you like it.

This document outline will appear on your screen:

```
<HTML>

<Head>
<Title></Title>
</Head>

<Body Background="" BgColor=#FFFFFF Text=#000000 Link=#0000FF VLink=#800080
ALink=#FF0000>

|

</Body>
</HTML>
```

The vertical bar '|' above represents the location of the cursor in the Arachnophilia editor screen. Now type "This is my first Web Page!" and then press the *preview button*, the toolbar button that looks like this:



If you are using the internal browser and your system is relatively fast, you may want to try Instant View on the Preview menu. This allows you to see a preview on every keystroke. If this feature makes your system slow down too much, you may want to disable it. And it can only be

used on relatively small, simple files – on large files, especially those with graphics, the browser just can't keep up with the keyboard

(Those using the internal browser may skip the next three paragraphs)

If you don't see your Web Page displayed, then a word of explanation is in order. When Arachnophilia runs for the first time, it looks around in your system for useful information, like whether Windows knows the whereabouts of a browser. If there is such a browser, Arachnophilia will use it to show your page.

If Windows doesn't know the location of a browser, Arachnophilia will ask you to locate one for it. If you actually don't have a browser on your system, you should stop this exercise, acquire one, and continue (or you may have skipped the step where you enabled the internal browser – please go back up this page). There are several good web browsers posted on the web (a possible catch-22 situation if you don't already have a browser with which to look for it) and some are free.

If you do have a browser that Windows doesn't know about, simply tell Arachnophilia its location, and our exercise can continue.

When you are done admiring your handiwork as displayed by your browser, just move the browser out of the way and Arachnophilia will reappear. Again, if you are using the internal browser, you may want to choose Window ... Tile Horizontally so you can see your document and the preview display at once.

Now let's experiment. I can't know what your background is or how much you know about computers, so you may choose to take each of these steps, or skip ahead if you wish.

Drag the mouse across the word "first" to select it, so the display looks like this:

This is my **first** Web Page!

(note: the text shown in **color** here will look white-on-black on the Arachnophilia editor screen, i.e. selected)

Now press the toolbar button that looks like this:



This button will make the selection **bold**, just like in a normal word processor. The Arachnophilia display will look like this:

This is my **first** Web Page!

Remember: all formatting in the HTML language is by way of tags. You won't see the bold effect until you click the preview button on the toolbar and let the Browser display the result of your

changes.

Now let's choose some more interesting colors for the display. Use your mouse to position the cursor as shown below:

```
<Body Background="" BgColor=#FFF|FFF Text=#000000 Link=#0000FF VLink=#800080 ALink=#FF0000>
```

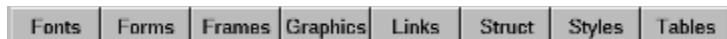
Notice the vertical bar in the middle of " BgColor=#FFF|FFF". This time, don't select anything, just position the cursor as shown. Now press the *right-hand mouse button*. A dialog box will appear asking you to select a color. Choose a color you like for a background. Now position the cursor in the middle of " Text=#000|000" and make a selection in the same way.

Now press the preview button again and look at the colors you have chosen. You can experiment until you are satisfied with your colors. You will notice that, if you have selected any of the text in the document and then click the right mouse button, a menu appears with some of the formatting tags that also appear on the toolbar, but if you just position the cursor in a color definition without selecting anything, the color selection dialog appears.

Now let's add an interesting graphic image for your page. Web Browsers use the background graphic as sort of wallpaper, i.e. they make repeated copies of the graphic to fill up the screen. Position the cursor as shown below:

```
<Body Background="|" BgColor=#FFFFFF Text=#000000 Link=#0000FF VLink=#800080 ALink=#FF0000>
```

Now look at the bottom of the screen. You will see a row of tabs that looks like this:



Each of these tabs launches a separate toolbar that can be placed in any convenient location. For this example, click "Graphics" and then select EditImg from the toolbar that will appear.

Arachnophilia will now ask you whether you want to save your file. This is a very good idea for several reasons. One reason is that you might otherwise lose your work in the event of a system crash or power failure. Another is that, if you locate your Web page in a particular directory, Arachnophilia can automatically move all your resource choices to that directory also. This makes it very easy to move your completed Web page (or pages) to the Internet later on.

After you have responded to the Save-File dialog box, hopefully by saving your file, Arachnophilia will ask you whether you want to automatically copy any files you select to this same directory. This is a good idea also, because (1) all your work will be located in one place, and (2) you are then allowed to use "relative links" to identify your resources, links that will work the same way no matter where you move your Web page.

These two dialog boxes are intended to guide you to a method that will seem desirable and obvious when you have had more experience.

Now navigate to a directory that contains some graphic files and select one. The two most-often used Web page file types are Graphic Image Format with the suffix .GIF, and Joint Photographic Experts Group (wow -- what a title!) files with the suffix .JPG. There are some others, but these are the best choices. If you don't have any graphic images with these formats, perhaps you have a paint program (Corel Photo-Paint is one example) that can convert some of your existing images to one of these two formats.

Having selected a graphic image, you may once again see the result of your work by previewing.

By the way, if you make a change you don't like, you can undo it by pressing this toolbar button:



Now let's do some rudimentary formatting to improve the appearance of our page. Let's make two horizontal bars, one above and one below our typing. Position the cursor as shown below:

```
<Body Background="" BgColor=#FFFFFF Text=#000000 Link=#0000FF VLink=#800080  
ALink=#FF0000>
```

|

This is my **first** web page!

```
</Body>
```

Now press the right-hand mouse button and select "Horizontal Rule" from the menu that will appear. Now the display should look like this:

```
<Body Background="" BgColor=#FFFFFF Text=#000000 Link=#0000FF VLink=#800080  
ALink=#FF0000>
```

```
<Hr Width=95% Align=Center>
```

This is my **first** web page!

```
</Body>
```

Make another horizontal rule below your typing, so things look like this:

```
<Body Background="" BgColor=#FFFFFF Text=#000000 Link=#0000FF VLink=#800080  
ALink=#FF0000>
```

```
<Hr Width=95% Align=Center>
```

This is my **first** web page!

```
<Hr Width=95% Align=Center>
```

```
</Body>
```

Now select the entire phrase "This is my first web page!", press the right mouse button, and select "Center" (there is also a toolbar button for this). Now look at your work in the browser.

Add paragraph breaks at the end of each line (some browsers will require this, some won't, but it is good HTML style). These are available from the right-click menu, or you can just type Ctrl-Enter:

```
<Body Background="" BgColor=#FFFFFF Text=#000000 Link=#0000FF VLink=#800080  
ALink=#FF0000>
```

```
<Hr Width=95% Align=Center><P>
```

```
<Center>This is my <B>first</B> web page!</Center><P>
```

```
<Hr Width=95% Align=Center><P>
```

```
</Body>
```

This is important to understand: Normal line breaks don't mean anything to HTML -- there has to be an explicit tag that specifies a line break, such as <P>. There are two varieties of line break -- <P> (Ctrl-Enter), which separates paragraphs with a blank line (two line feeds), and
 (Shift-Enter), which provides a single break between lines (one line feed). Experiment with these and don't expect normal blank lines (with no tags) to appear in the Browser view.

Advanced Topic: Right-Clicking

When you press the right mouse button, what happens depends on where the text editing cursor '|' is positioned. If the cursor is pointing at a color definition (example: "Text=#000000"), you will see a color selection dialog. If your cursor is positioned over a file name, you will see the appropriate file dialog box.

If you have selected some text and press the right mouse button, a menu will appear with editing choices. If you simply position the cursor without selecting any text, the Right-Click Wizard will decide what action to take.

Once again, let's change the color of the page's background. This is quite easy -- just position the editing cursor as shown below:

```
<Body Background="" BgColor=#FFF|FFF Text=#000000 Link=#0000FF VLink=#800080  
ALink=#FF0000>
```

The vertical bar '|' shows the cursor location. Now press the right mouse button -- you will see a color selection dialog. Choose a different color and press OK. In order to see your color change, you will need to press the Preview button.

You can change the text color just as easily. Place the editing cursor here “`Text=#000|000`”, press the right mouse button, choose a color, and press OK.

For more on selecting colors and other right-click functions, read [The Right-Click Functions and Menu](#).

Using Arachnophilia

This section describes the general use of Arachnophilia, and some of the ways you can make Web page creation easier.

Before you run Arachnophilia, always be sure to close any open browsers. This allows Arachnophilia to control the behavior of your browsers during Preview.

To create a blank document, not necessarily an HTML document, press File ... New Blank Document or press the corresponding toolbar button.

To create an HTML document that includes the basic structure of HTML, press File ... New HTML Document, or use the leftmost toolbar button, or use the right-click menu in an unoccupied part of the Arachnophilia program window. You will see the "Create HTML Document" dialog, which allows you to choose text and background colors and an optional background graphic as well as some other less-often-used options. You may also change these options after you have created the document by placing the cursor on the item of interest and pressing the right mouse button. You may then change the name of the background graphic or your color choices, as well as several other things.

To open an existing HTML document, select File ... Open, press the corresponding toolbar button, or use the right-click menu.

To save a document, choose either File ... Save or File ... Save as, or press the corresponding toolbar button, or use the right-click menu within the document display.

You may find and/or replace any text string you wish using either Edit ... Find or Edit ... Replace if you want to replace one text string with another, or the corresponding toolbar buttons. You can specify two non-printing characters using special symbols. The special symbols are \p for line ending (the division between lines in a text editing document), and \t for tab character.

You can Cut, Copy and Paste within or between Arachnophilia documents and between Arachnophilia and other Windows applications, although advanced features such as drag-and-drop are often easier to use.

You can test the result of your work at any time with up to six user-identified Web browser programs.

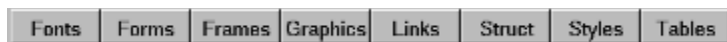
Beyond these normal behaviors, be sure to see Advanced Features of Arachnophilia for some spectacular capabilities that will really make your Web Pages stand out from the crowd!

Advanced features of Arachnophilia

Arachnophilia has many ways to make your work easier. You may drag a selection from a Arachnophilia document into another Arachnophilia document and drop it at a chosen location, or you may drag a Arachnophilia selection into another Windows application, or the reverse. But that is just the beginning.

Toolbar Customization

You can use the toolbar selections at the bottom of the main display to launch many independent toolbars that can be positioned in any convenient way. The default toolbar selector bar looks like this:



Each toolbar contains a group of related functions. You should explore these toolbars to gain an understanding of their functions.

After you have examined the toolbars, you can change them if you wish! Simply select Tools ... Toolbars/Macros ... Edit Toolbars to enter the Toolbar Editor. You can change any command's contents, you can add and remove commands, you can even create complete new toolbars! For more on this subject, read [The Toolbar Editor](#) .

For those who want to create entire sets of toolbar definitions, all the commands on these toolbars come from special Toolbar Definition Files (with the suffix TBD), located in the "toolbars" subdirectory of your Arachnophilia program directory. You can change the commands on the toolbars, or create entirely new toolbars, by editing the contents of these files.

Also, there is an Excel 7.0 spreadsheet named "arachnophilia command sets.xls" located in the "toolbars" subdirectory that will create the entire set of TBD files for you. The most secure and efficient way to use this sheet is to make a backup copy of the original, and edit the contents of a copy.

NOTE: if you use the Excel spreadsheet to change commands, any changes made with the Toolbar Editor will be lost – these two methods try to control the file contents independently. You should choose one or the other of these methods, but not both.

If you make a copy of one of the worksheets within the Excel spreadsheet and give it a different name, the spreadsheet will create a new TBD file for you with the name you have chosen. If you just add new commands and edit existing commands, these changes will appear the next time you run Arachnophilia.

If you don't have a copy of Excel, you may want to try loading the spreadsheet into another spreadsheet or database program, or you may edit the TBD file directly in a text editor, although this is more difficult.

IMPORTANT NOTE: If you have created your own custom commands, be sure to back up the .TBD files to a safe, second location. This way, if you update Arachnophilia, you will not lose your changes.

Power Undo

Most of the commands related to the actions of Arachnophilia are reversible. If you don't like the outcome of your action, you may always undo it with the menu Power Undo command or toolbar button:



If you use this button, and you don't like the outcome, you can press the Redo button to go back to the original version. Arachnophilia will save 32 versions (32 Undo and 32 Redo) of each of your pages in this way, so you can recover from experiments.

Smart Right-click

Some of the toolbar buttons allow you to edit a file, link, or color. To use these buttons, position the cursor in the area on interest and press the appropriate button. Another, faster way to accomplish this is to position the cursor on the link or color of interest and press the right mouse button. In most cases Arachnophilia will identify the target and launch the correct selection dialog. If the link contains the name of a graphic image, the most recent graphic directory is displayed and you can choose a replacement for the current graphic. If the link contains the name of Uniform Resource Link, you will see the directory that contains this file type, and so forth.

In general, if you prefer to right-click the mouse, first position the cursor without selecting anything to let Arachnophilia decide which action is appropriate, or select one or more characters to launch a menu of choices, or just use a toolbar function.

Keyboard Macros

You can define and use 26 keyboard macros, using Arachnophilia's built-in macro editor. For more on this subject, read the [Keyboard Macros](#) section.

See also:

[The Right-Click Functions and Menu](#)

[Arachnophilia File Import Methods](#)

Arachnophilia File Import Methods

There are three principal ways to import a fully-formatted text document from another application:

- Drag a Rich text Format (RTF) file icon from Windows Explorer to the Arachnophilia main window and “drop” it. Arachnophilia will then ask whether it should convert the document into HTML code.
- Use the File ... Open File ... RTF File function to import Rich-Text formatted files which contain the formatted text of interest.
- Drag a section of a document between the source program and a new, blank RTF document in Arachnophilia.

Arachnophilia will automatically convert your text into HTML code, keeping fonts and font colors, sizes, styles such as bold, italic and other formatting, centered titles, and many other characteristics. It will create HTML tables from your tables, preserving styles as above. It will create multilevel indented outline-format lists, both numbered and bulleted, also preserving the formatting of the original document.

This means you can work in a full-featured word processing program of your choice, and, when you are satisfied with the appearance of your text, tables and outlines, just move them into Arachnophilia using one of the methods above.

A special note about table formatting: In the RTF file format, the positioning of text within table cells is not preserved. Therefore, if you want to control the position of text *within* the table's cells, you must either:

- Convert the table into a tabbed list in the word processing program and then select left, center, or right positioning on a line-by-line basis before importing the table, or if this is not possible,
- Set the positioning within Arachnophilia manually, after the table has been imported.

In this example you want only the text in the middle cell of the row to be centered (you add the text in **red**):

```
<Tr><Td>data</Td><Td align=center>data</Td><Td>data</Td></Tr>
```

In this example you want the text of the entire row to be centered within the cells (you add the text in **red**):

```
<Tr align=center><Td>data</Td><Td>data</Td><Td>data</Td></Tr>
```

This is a feature that is supported in most modern browsers.

Custom Definitions

You have complete control over the content of the toolbar structure in Arachnophilia. When you run Arachnophilia, the toolbars are created from simple text files located in the “toolbars” subdirectory of the Arachnophilia program directory. These files have the suffix .TBD, which stands for Tool Bar Definition.

NOTE: In recent versions of Arachnophilia, you may edit the contents of the toolbars while running the program. For more on this new feature, see [The Toolbar Editor](#) .

I have provided a spreadsheet program compatible with Excel 7.0, also located in the “toolbars” subdirectory, that contains the content of these files and is an easy way to add new toolbars or edit the contents of those already defined.

The file is named “arachnophilia command sets.xls,” and, if you own Excel 7.0, you may simply click on the program icon to run the program. You may add any tags you wish, or change those that exist to suit your personal preferences. If you copy one of the worksheets within the workbook and give it a new name, this will become a new toolbar in Arachnophilia.

The user-command entries may contain some special tokens -- the token “\p” which results in a line ending when the command is carried out, the token “\t” which results in a tab (you cannot use literal tabs because tabs are used in the .TBD file format to separate fields), and the vertical bar “|” which identifies the user's selection before the command is issued. There are more special characters described in the [Keyboard Macros](#) section.

For example, let us say that a new HTML tag is invented called the SHOUT and it has this syntax: <SHOUT>this phrase</SHOUT>. To define a command to handle this new tag, you would enter:

Button name: “Shout” (this should be relatively short).

Command: <SHOUT>|</SHOUT>

Tool Tip Text: “Shout Tag”

Status Bar: "This is the new SHOUT Tag"

After you have entered this definition, a new command button will appear on the specified toolbar, and if you select a phrase and press the button, the tags <SHOUT> and </SHOUT> will enclose you selection.

Beyond these formatting procedures, there are some special commands, enclosed in braces, that execute Arachnophilia program commands. Here is an example:

One of the standard tags (from the “Graphics-Misc” toolbar) creates a new graphic image tag, and (by launching the graphic selection dialog) fills it with something you have chosen. Here is the tag:


```
<IMG src="[FullTagGraphic]" alt="">
```

This tag creates an editing entry just like the others, but, because of the special entry [FullTagGraphic], causes a file selection dialog to appear as the tag is being constructed. If you choose a graphic named “mygraphic.jpg,” the resultant tag would look like this:

```
<IMG src="mygraphic.jpg" alt="">
```

After you have defined this tag, you may simply right-click the graphic name and Arachnophilia will know what to do, but the first time -- when there is no filename to offer guidance -- the [FullTagGraphic] command is a way to automatically launch a file selection dialog.

The special string “[FullTagGraphic]” is called a “system command.” There is more on the subject of system commands in the [Keyboard Macros](#) section, and you can see a list of all the system commands by selecting Tools ... List System Commands.

One of the worksheets in the Excel workbook doesn’t become a toolbar, instead it becomes the right-click menu. This sheet has the internal name “RightClickMenu,” which is how Arachnophilia identifies it. The commands you see when you press the right-click menu are located in this sheet.

If you do not own Excel 7.0, you can still add to the toolbars and edit their contents using a word processor or even a simple text editor -- this is just less convenient. In this case, if you want to add a new toolbar, simply copy one of the TBD files in the “toolbars” subdirectory, change its file name and its internal name, and create the commands that meet your requirements.

The Right-Click Functions and Menu

When you are editing text, you may press the right mouse button to gain access to some special functions and a shortcut menu. Here is how:

To let Arachnophilia choose the appropriate action, simply place the editing cursor at the desired location, without selecting anything, and press the right mouse button. At this point, the Rick-Click Wizard will choose the appropriate action. If Arachnophilia finds itself pointing at a color definition, the color dialog will appear. If Arachnophilia detects a file name, a link, a resource, or one of many other things, the appropriate dialog box will appear, allowing you to make a new choice.

Example: if you position the cursor this way (the cursor is marked with the vertical bar character "|"):

```
This is a <Font Color=#FFF|F80>Color Block</Font> to emphasize it.
```

Arachnophilia will detect the color definition and display the color dialog box. If you point at this:

```
<IMG src="book|case.jpg" alt="">
```

Arachnophilia will open a "Change Graphic Image" dialog box, directed to the most recently accessed graphic file directory. In general, if you place the cursor on a color or a resource specifier string, Arachnophilia will automatically sort out your intentions.

If Arachnophilia cannot detect any of these things, or if you select some characters before pressing the right mouse button, a shortcut menu will appear which allows you easy access to many of the most-used HTML tags. And this shortcut menu is user-definable – see [Custom Definitions](#).

The File Menu

The File menu has the normal commands for opening, saving, and closing documents, plus some special commands uniquely related to Arachnophilia's purpose.

The File ... New File command opens one of many kinds of document. Some document types are treated in special ways, so be sure that you choose the correct document type.

The File ... Open File command allows you to choose which kind of document you are opening. Arachnophilia remembers which directory contains which file type, so this makes it easy to keep track of file locations.

On the subject of opening files, I have received a number of requests for information about how to open more than one file at a time. Here are some of the ways:

1. Select as many files as you need in the File ... Open dialog.
2. Using File Explorer, select as many files as you need and drag them onto the open, empty Arachnophilia program window.
3. Using File Explorer, select as many files as you need and drag them onto the Arachnophilia program icon.
4. After enabling Arachnophilia's "Open With ..." feature (Tools ... Open With Arachnophilia), in File Explorer select as many files as you wish, then open the File Explorer right-click context menu.
5. Create a shortcut to Arachnophilia with a file list included.

The Close All command closes, and if necessary saves, all your documents at once. This is an easy way to close part of a project and clear your workspace for another task.

The Save All Changed Files command simultaneously saves all open documents that have been changed in any way, with just one mouse press.

The Print and Print Preview commands allow you to print your document, but not like a word processing program would, with page numbers and margin control. This command is primarily used to acquire a scratch-pad style hard copy for programming purposes. For more control of the printed output, I recommend that you transfer your document to a word processing program for printing.

The Send Mail command allows you to e-mail a copy of your work.

The Edit Menu

The edit menu contains two commands to Undo or Repeat an action as well as the usual Cut, Copy, and Paste commands.

Undo works in an obvious way -- it simply copies the entire document into temporary storage at each issued command (not during keyboard text entries), and, if you decide you don't like the outcome of a command, you can simply undo it. If you don't like the undo, you can repeat the action. This method works because most Web pages are relatively short and therefore occupy a small amount of memory.

That's the good news. The bad news is, if you are working on large documents, you will run out of memory by using Undo. So you have the option of turning off this feature (see [Tools Menu](#)). if you find yourself running out of memory in seemingly normal circumstances, try turning off Power Undo.

The Find and Replace commands will search the entire document, or from the cursor to the end of the document, or all open documents, depending on your choices. These dialogs accept two special characters that represent non-printing characters in your document: `\p` represent a line ending, and `\t` represents a tab character. This means if you type `'\p'` (without the ' characters) as part of your entry, Arachnophilia will search for line endings (the character pairs that separate lines in your document). The replace entry also accepts these special characters.

NOTE: To enter a string that contains a literal reverse slash, simple enter the reverse slash twice, like this:

```
\\p
```

This entry searches for the literal string `'\p'`, it does not convert it to a control character or sequence.

The Selection Menu

Each of the functions in this menu operates on a block of text that you have selected with the mouse or keyboard. To operate on the entire document, press Ctrl+A or choose Edit ... Select All. Also remember: in most cases, if you don't like the outcome, you can recover the original selection with Power Undo (use the left-pointing arrow on the toolbar).

NOTE: Each of these commands can be made into a keyboard macro using **Tools ... Toolbars/macros ... Edit Keyboard Macros**. This makes it easier to access your favorite commands.

Convert Chars - Convert Extended to HTML Characters

This command scans the current HTML document for characters that have ASCII codes above 127. These characters may not display correctly in some server environments, so there is a convention in HTML coding that represents them in a form readable by most server systems. Each such character is converted into a "character entity," meaning a special string that can represent that character in many environments that don't support character codes above 127 directly. For example: the character ã is converted to the string "Ã". If there is no character entity for a particular character, this form is used instead: "&#NNN; ", where NNN is the character code expressed as a number. You can type in such codes by hand, but if you use a large number of such characters, you may choose to convert the entire document at once using this tool.

Convert Chars - Convert HTML to Extended Characters

This function performs the opposite task of the above-described function – it converts HTML symbolic characters into their extended equivalents.

Assemble Lines

This function takes the current selection of text, normally individual lines, and assembles them into a paragraph. Use this function to create normal paragraphs out of broken lines, a computing problem for which there is no general solution and which therefore must be done by hand.

There are two options – keep tabs and remove tabs. The remove tabs option removes all tabs from the selected block as it assembles it. This is usually necessary when assembling e-mail messages and indented text.

Remove Tabs Only

This function removes the tabs from a selected block. This can be used to undo part of the effect of Beautify HTML, if you wish to return to an earlier document state.

Text Case -- To Uppercase, To Lowercase

These functions shift the case of the selected block.

Tag Case – To Uppercase, to Lowercase

These functions convert the tags in the selected block to uppercase or lowercase without changing the text case. Example:

Uppercase tags: `Bold Text`

Lowercase tags: `Bold Text`

NOTE: If you want Arachnophilia to consistently use uppercase or lowercase tags, set or reset this option at **Tools ... Options ... Miscellaneous ... Use lowercase system-generated tags**. If this option is selected, all toolbar, system button, HTML file template and other tags will be lowercase. If this option is cleared, all tags will be uppercase.

Tag Delimiters - All "<>" to "<,>"

This interesting function converts the real HTML tag delimiters “<>” into their symbolic equivalents “<,>”. This makes HTML code show up in your page in its original form, instead of being interpreted by the HTML engine. This is used to show actual code in your page, perhaps for a tutorial on the writing of HTML code.

Tag Delimiters - All "<,>" to "<>"

This function reverses the effect of the above function – it converts symbolic HTML delimiters into real ones.

Escape Selection

This programmer’s function converts a selection of text from plain text into a type of text suitable for placing in a C-style program string. Common control characters are “escaped,” that is, they are preceded by “\”. This is also done for any quotes in the selection.

Unescape Selection

This function performs the opposite of the above function – it takes text intended for a program string and converts it into plain text.

Compare Delimiters [({})]

This programmer’s function counts and compares the specified delimiters. If they are not balanced, an error message is printed specifying which delimiters are out of balance and their number. This is a simple version of Beautify Code, intended to assist in debugging short sequences of code and, in particular, solving that perennial programmer’s task of balancing parentheses in complex code expressions.

Strip All HTML Tags

This rather drastic function simply strips out all the HTML tags from your selection. You will want to be careful with this function – apart from Power Undo, there is no way to reverse its effect. This function can be used to extract text from an HTML document for use in another application. This function, along with “remove tabs” and/or “assemble lines,” can be used together to convert an HTML page into a normal, paragraph-oriented text document.

The Tools Menu

The Tools menu contains some explicit commands that are normally carried out automatically, also some program modules for specialized tasks.

Tag Context Coloring

The first three functions in the Tools menu control various aspects of tag context coloring. Tag context coloring is a system that makes reading and editing an HTML document much easier – the tags are colored differently than the text, and the definitions within the tags have yet another color. Also, if the number of tag delimiter characters (“<” and “>”) is unbalanced, special colors appear to alert you to this fact, and to localize the error.

You may choose tag context colors according to your preferences using functions in the Tools ... Options menu.

On some system, the default choice of automatic tag coloring will slow things down too much, so there are some alternatives. Here are the details:

Refresh View Tag Colors

This function (also Ctrl+T) refreshes the tag colors in the viewed window. This is normally done automatically, unless you have chosen to turn off automatic tag coloring (see below). There is also a toolbar button for this purpose.

Refresh Document Tag Colors

This function (also Ctrl+Shift+A) recolors all the tags in your document. This is for cases where the automatic tag coloring feature is turned off and you don't want to press Ctrl+T repeatedly as you move through the document.

Auto Tag Coloring ON/Off

This function (also Ctrl+Shift+T) switches the automatic tag context coloring on and off. Some systems do not perform well when automatic tag coloring is enabled, because of limited speed or resources or some other reason. For speedier performance on such a system, use this option. Then you can use manual tag coloring as required to context-color specific parts of your page.

Table Wizard

This command launches the Arachnophilia Table Wizard, which interactively creates a table for you. There are many choices including number of rows and columns and choice of background color. Table Wizard will even create a table out of block of data you select in you document. Read more about this feature in [Table Wizard](#).

NOTE: If you select some data in your document, launch Table Wizard and choose a table size

of one row, one column, your document selection will be placed in the data cell. This is a very easy way to enclose a text block or graphic.

List Wizard

This command launches the List Wizard, the tool for automating list creation. Read more about this tool at [List Wizard](#).

Convert RTF to HTML

This command, enabled only for RTF documents, converts them into HTML pages using the Arachnophilia RTF Translator Module. Options for this translation are located in the Tools ... Options menu (see below).

Analyze Site

This command will analyze your site for internal consistency. See [Arachnophilia Site Analyzer](#) for more information.

Update Web Site

This command uses the Internet FTP protocol to upload new or changed pages to your Web site. See [The Site Update Tool](#) for more information.

... In the Beautify submenu

Beautify HTML / Analyze HTML

This command will reformat your HTML document, indenting it as though HTML is an actual computer language (it is not), and will then list any syntax errors it finds. This function is a very valuable way to discover and correct errors. You may use the Beautify HTML / Analyze HTML dialog to interactively locate errors – just use the dialog to select those tags you are having problems with and run Analyze again. In most cases, the missing tag's location will show up clearly.

For more on the feature, see [Beautify HTML / Analyze HTML](#).

Beautify Code

This command will find and correctly indent program code listings, such as Perl/CGI scripts and C/C++ programs, and code segments within you HTML pages. While indenting the code listing, it finds any unmatched pairs of the structural characters “{,},(,)”, thus performing a simple syntax check on your program. If it finds any unmatched structural characters, it will alert you to this fact with an error message. For more on this function, see [Beautify Code](#).

...In the Toolbars/Macros submenu

Edit Toolbars

This command launches the Toolbar Editor, which allows you to create new toolbars and edit existing toolbars, creating any number of custom commands. Read about this subject at [The Toolbar Editor](#).

Edit Keyboard Macros

This function allows you to create keyboard macros that carry out either system commands, or commands that you write. Read more about this in [Keyboard Macros](#).

...In the Lists submenu

List System Commands

This list shows system commands to assist you in creating keyboard macros. The system commands are also available as dropdown lists in both the keyboard macro and toolbar editors.

List User Macro Definitions

This display offers an easy way to see what commands you have created.

List Keyboard Shortcuts

This is a list of the keyboard shortcuts that are available in Arachnophilia. Some of them are standard Windows shortcuts, others are unique to Arachnophilia.

List ASCII Character Codes

A comprehensive list of ASCII characters, their numerical equivalents in Hex, Decimal and Octal bases, and their HTML entities if defined.

...In the Set Tab Size submenu

Set Tab Size

This function sets the default tab size for your document display. This function is mostly used to make program listings easier to edit, and is used in conjunction with the Beautify Code and Beautify HTML functions described below.

You may also use `Ctrl+Q` and `Ctrl+W` to set the tabs size conveniently.

...In the Insert HTML Tag submenu

This submenu is a backup for other ways to insert tags that rely on the launching of a file dialog

box or a color dialog box.

... Back on the main Tools menu

Calculate Graphic Size

This function creates WIDTH and HEIGHT values for an IMG tag. This function is automatic if you select a graphic in the course of creating an IMG tag. This menu option is provided to update old tags that may have been created in an older environment. Simply place the editing cursor anywhere in the name of the graphic, like this:

```
<IMG SRC="graphic|name.jpg">
```

and select this function from the menu (the '|' character indicates the location of the editing cursor).

You may also use Ctrl+N to access this function.

Fix line endings

This function converts the line endings in your file from the UNIX style (a single line feed) to the Windows style (a carriage return plus a line feed). Use this function if you see unexplainable errors while using the Beautifier, or if you have imported the file from a UNIX server directly. There are any number of strange consequences related to using a UNIX formatted file in Windows.

Your FTP client program can solve this problem for you, if you allow it to, and Arachnophilia's FTP client intelligently converts on upload.

Toggle Word Wrap

This feature conveniently keeps all the text in view, but sometimes you want to allow the lines to extend off the screen to the right. For example, when you analyze your site using the [Arachnophilia Site Analyzer](#), you will want to use a fixed pitch font and turn off word wrap for best results. Also, when you use Analyze HTML, you will need to turn off Word Wrap in order to locate errors.

Base Calculator

This cute little calculator is used to convert number bases. If you type a number in any of the supported bases, the calculator will perform its conversion on each entered digit. It will tell you if you have exceeded its 32-bit number-size limit, and it will tell you if you use a digit not appropriate to the current base.

Open with Arachnophilia

This command launches a dialog that allows you to associate Arachnophilia with many Windows file types for the purposes of editing. It won't prevent these files from being opened and run by the default applications, it just places an additional option on the File Explorer right-click "context" menu with the label "Open with Arachnophilia."

The **Tools ... Options** menu takes care of several Arachnophilia setup functions. This menu has sections for Conversions, Display, Tag Coloring and Miscellaneous.

The Conversion Options

These options control how Arachnophilia converts documents into HTML pages.

Automatically convert/Ask to Convert/Do not convert

This option chooses a level of automation for the process. If “Automatically convert” is chosen, RTF documents are converted as they are loaded. See [Arachnophilia File Import methods](#) for more on this subject.

Convert tabbed lines into Tables/Tabs into Spaces

This function controls how Arachnophilia interprets tabs while importing Rich Text Format (RTF) documents and converting them into HTML. If your document contains tabs that are not part of tables, Arachnophilia may get confused and turn them into tables you didn’t intend. To control this behavior, simply select “Convert tabs into spaces,” but remember that you did this, because later you may want to create tables. In order to do this, you must re-enable the default option “Convert tabbed lines into Tables.”

Convert Extended Chars into HTML

This function converts extended characters (characters with numerical codes greater than 127) while importing an RTF document. These characters may not display correctly in some server environments, so there is a convention in HTML coding that represents them in a form readable by most server systems. Each such character is converted into a “character entity,” meaning a special string that can represent that character in many environments that don’t support character codes above 127 directly. For example: the character Ñ is converted to the string “Ã”. If there is no character entity for a particular character, this form is used instead: “&#NNN; “, where NNN is the character code expressed as a number. If your server accepts extended characters and you want to reduce the size of your page, or if you want to be able to see and edit the original characters in the Arachnophilia HTML file, disable this function.

Convert “<” and “>” into Visible Chars

This function controls how Arachnophilia interprets these two special characters. Normally these characters enclose HTML tags, so if they appear in normal text, they must be treated in a special way. If this function is enabled (the default), these characters will be made visible in the HTML page, so a line reading “5 is > 3” will be displayed correctly. Conversely, if you have an import document that contains HTML tags, and you want to preserve these tags in their literal form, disable this function.

Choose output Background Color

This option chooses a default background color for the converted HTML documents.

The Display Options

Choose Default Font

The default font option allows you to choose a different font and size for the editing display. I recommend a fixed-pitch font such as Courier New, because HTML code looks much better when such a font is used.

Choose Background Color

This option allows you to select a background color that is different than the Windows default color. This function permits you to see (to some extent) what your HTML page will look like when viewed on a browser. Just choose the same color for your HTML editing screen that you have chosen for your browser display.

Use Default Windows Background Color

This option makes the program use whatever background color your system uses by default. It overrides any background color choice you may have made.

Use Word Wrap

This feature conveniently keeps all the text in view, but sometimes you want to allow the lines to extend off the screen to the right. For example, when you analyze your site using the [Arachnophilia Site Analyzer](#), you will want to use a fixed pitch font and turn off word wrap for best results. Also, when you use Analyze HTML, you will need to turn off Word Wrap in order to locate errors. This function is duplicated at Tools ... Word Wrap.

The Tag Coloring Options

Arachnophilia will automatically color the tags in an HTML document, making it easy to distinguish among normal text, HTML tags and the definitions within tags. These buttons allow you to customize the color choices. Your choices are preserved between Arachnophilia work sessions.

There are some default colors that are not on this list. These colors only appear when the number of “<” and “>” symbols is unbalanced. This is an efficient way to detect a common HTML coding error – an orphan tag delimiter (either “<” or “>” alone).

If you have an extra “<” symbol, a purple color appears below it on the page. Conversely, if you have an extra “>” symbol, the error color is red. You will quickly learn how to read the colors and locate coding errors that are very hard to spot on an ordinary HTML editor.

On slower computers, you may notice that automatic tag coloring slows the display. In this case, try disabling the feature and using the manual options instead.

Miscellaneous Options

This grab bag of options covers a lot of ground, but don't fit very well into a particular category.

Small toolbar icons – if you have a small display size, you may wish to use the small icon toolbar.

Full pathnames – this option displays the full path for a file on the application's title bar. This is useful while working on a large project with many files with the same name, such as “index.html.”

Maximize Documents – This option defaults to maximized for newly opened documents.

Enable Power Undo – This option allows up to 32 “undos” and “redos” of actions taken in the editor, and the 32 recoveries are unique to each document. This allows you to recover completely from mistakes but, in exchange, this option requires a lot of memory. If you receive memory error messages or the program runs too slowly when editing large files, turn this option off.

Use Character Entity Strings

This option chooses whether to use “character entity” strings to represent extended characters, or to use the default notation that simply represents each extended character as a number. If this option is selected, the character “Ã” will be represented as “Ã”, if this option is not selected, this character will be represented as “Ã”. This option affects the RTF import feature and the Tools ... Convert Extended Characters function.

Create Windows Line Endings on Save

This option deals with one of the more vexing problems with DOS/Windows based systems – each line ends with two characters (a carriage return and a line feed), unlike civilized, rational operating systems like UNIX. There is an FTP client program (WS_FTP) that, if it detects even one line feed without an accompanying carriage return, will strangle and die. So, even though this is not an Arachnophilia problem, I have decided to respond to the many complaints I receive about this problem. Select “Filter Line Ending on Save” to automatically prevent a failure of WS_FTP. This option takes some time while saving large files, so you may want to turn it off under those circumstances.

Create Unix Line Endings on Save

While saving, this option creates Unix line endings in your file, the opposite of the above option. Only use this feature if you are experienced and know the consequences of having mixed-format files in a Windows file system.

NOTE: Only one of the two preceding options can meaningfully be chosen at a time, even though both can be enabled at once. And, to prevent either option from being exercised, disable both of them.

Allow HTML Features in non-HTML Files

This option solves the problem that no finite number of file suffixes will ever be enough – someone always has a new file suffix that is supposed to represent an HTML file. This option simply enables the HTML features of Arachnophilia, no matter what the file type. This option should be used with care.

Calculate and Update Graphic Size Tags

This feature automatically creates “WIDTH” and “HEIGHT” tags when you insert or change a graphic file tag. In some cases this will not be desirable – for example, you may wish to stretch a graphic to use it as a line. In this case, you should disable this feature.

This feature automatically calculates the size of graphic when the tag is created, or you may simply place the editing cursor in the tag and select Tools ... Calculate Graphic Size.

Use Lowercase System-Generated Tags

Many people prefer the appearance of lower-case tags. An equal number of people prefer uppercase tags. If selected, this option makes all system-generated tags lowercase, including newly opened HTML documents, toolbars, and converted RTF documents. If not selected, all the described tags are made uppercase.

NOTE: If you have already-created documents whose tag case you want to shift, you may do this with the functions on the Selection Menu.

Keyboard Macros

This powerful feature allows you to define 26 keyboard macros (expansions of a keystroke), one for each letter of the alphabet. While using the editor, just press Alt-(letter) to activate your macro. If you want, you can embed Arachnophilia menu commands in your macros. You can also make one macro call another, and you may specify a number of repetitions.

IMPORTANT: If you are accustomed to pressing Alt+F to access the File menu, Alt+E to access the edit menu and so forth, don't use these particular macro letters. Any Alt+Key that is not defined simply reverts to its ordinary Windows role – accessing menu items.

Where are my Macros?

The actual macro table is saved in the Arachnophilia program directory at `toolbars\keymacros.txt`. You may edit this file directly if you must, but the macro editor is easier to use. To enable the macro editor, choose Tools ... Edit Keyboard Macros or press the macro toolbar button. To list the current definitions of your macros, select Tools ... List User macro Definitions.

System Commands

You can embed Arachnophilia commands into your macro -- this allows a high level of automation and customization. You may list all the permitted commands by choosing Tools ... List System Commands. Here are some examples of their use:

The macro “<CENTER>\p[tablewizard]\p</CENTER>” will center a place for a table and launch the table wizard to allow you to customize the table. The special string “\p” is converted into a line ending by the macro generator – this is a convenient way to add a line ending without necessarily typing it (although you can type it if you want – the macro editor will accept either form). Also see “Special Character Sequences” below.

You can even execute one macro from another. The special System command “[macro:?” causes the macro with the letter “?” to be executed.

What value does this have? Well, think about it. You can create one macro with a style for a table cell:

```
Macro A:<TD align=center>my data</TD>
```

(remember: you don't type “Macro A:”, just the characters that follow).

Then you can specify a row of such cells with another macro:

```
Macro B:<TR>[macro:a] [macro:a] [macro:a] [macro:a]</TR>\p
```

Then you can assemble the entire sequence into a table:

```
Macro C:<TABLE>\p[macro:b] [macro:b] [macro:b] [macro:b]</TABLE>\p
```

Then, having written your individual macros, you can go back to the top macro and change the cell formatting for the entire table by typing just one entry.

For example, if you change Macro A to this --

```
Macro A:<TD align=center> [fulltaggraphic]</TD>
```

-- and run Macro C, the program will ask you for 16 graphic file names (one at a time) and create a 4 x 4 table of the results.

What happens is macro C sets up the overall structure for the table, then calls Macro B four times to create the rows. Macro B creates the rows and calls Macro A four times per row to create the table data cells.

Repetition

Instead of typing “[macro:b] [macro:b] [macro:b] [macro:b]” as in the example above, you may simply type “[macro:b4]” This means “execute the B macro 4 times.” This works if you don’t need to insert any characters between successive repetitions of the macro. And remember: if you do want special characters to surround the macro result, you can always put them in the called macro, as in the examples above.

You can even call a macro from within a macro (this is an option for malcontents and deeply disturbed persons). What happens if you enter this:

```
Macro X:{[macro:x]}
```

The answer: the macro calls itself recursively 16 times, then a counter detects this condition and prints an advisory (you also see a big collection of curly braces). This is mostly to catch an otherwise fatal condition, usually caused by a typographical error.

Special Character Sequences

You can make multi-line entries for each macro, and you can add special characters to accomplish formatting not ordinarily possible. Here is a list of special “escape” sequences to permit special characters to be entered:

String Meaning

\a	Bell
\b	Backspace
\f	Formfeed
\n	Linefeed
\r	Carriage Return

`\p` Line Ending (`\r\n`)
`\t` Horizontal Tab
`\v` Vertical Tab
`\nnn` Character number in octal notation
`\xnn` Character number in Hexadecimal notation

The "nnn" in the list above are to be replaced with the appropriate numerical digits.

Because the reverse slash is used to identify these special entries, it is not available with a single keystroke, but if you want the reverse-slash character to appear in your macro's output, simply type it twice - `"\"`.

Finally, you may want to position the cursor in a particular way after the macro is finished.

Here's an example:

```
Macro A:<IMG SRC="[inserttaggraphic]" ALT="">|</A>
```

In this tag, the character `"|"` indicates the position of the cursor, so when the tag has been created, the program places the cursor where you will be typing the user-visible part of the tag.

The Toolbar Editor

You may edit the contents of the custom toolbars from within Arachnophilia. Simply select Tools ... Edit Toolbars. When you do this, Arachnophilia will temporarily hide the existing toolbars (in preparation for changes) and launch the toolbar editor.

All the capabilities that apply to [Keyboard Macros](#) and [Custom Definitions](#) also apply to the toolbar editor – you may include system commands, references to keyboard macros, and special characters.

The toolbar editor is broken into two sections – “Select/Edit Toolbar” and “Edit Toolbar Commands.” You use “Select/Edit Toolbar” to choose which toolbar you want to edit, then you use “Edit Toolbar Commands” to make any changes you want.

You can add, copy and delete toolbar commands, and you can copy commands between toolbars. You can even create new toolbars using the “Select/Edit Toolbars” section.

Please remember: The toolbar editor gives you a lot of power. If you delete a command from a toolbar, it is gone and the disk file containing the toolbar will reflect this change. If you delete a toolbar, the disk file is also deleted. So use these features with caution, **and you may also want to back up the default toolbar files** to a safe location. The toolbar files are located in the Arachnophilia program directory, in a subdirectory named “toolbars.”

Also remember: If you use the Excel Spreadsheet that was used to edit commands in earlier versions of Arachnophilia, the spreadsheet will overwrite any changes you have made from within the program.

Once you have made your changes, press “Close,” Arachnophilia will rebuild the toolbars and command menus, and your new toolbars and commands will be in effect. And, when you exit Arachnophilia, your changes will be saved.

The Arachnophilia Site Analyzer

This function, available at Tools .. Analyze Site, create a cross-reference list of your site's files, organized as a tree structure.

The basic structure of a Web site is similar to a tree, with a file (usually named "index.html") as its root. A visitor to your site first sees index.html, then branches out from there through other pages you offer and to other places on the Web by way of hypertext links, which form the branches of the tree.

As development progresses, some things may happen that you might prefer did not. Here is a short list:

- You may decide against using a particular resource in your pages, and you may then change the identifying tag to point at another resource instead. But, being human, you forget that the resource is still stored in your site directory.
- You may make a typographical error in identifying a resource, and then not test that tag to verify that it is working.
- Someone on the Internet may change their identifying Uniform Resource Locator address without visiting your house, telling you personally, and delivering a box of candy to take you over the disappointment. Thus, that tag in your pages will no longer point to anything desirable.

The Arachnophilia Site Analyzer will list your site's resources in three sublists, organized as tree structures:

1. Referenced on-site resources, meaning resources that both exist and are connected to the tree.
2. Unreferenced on-site resources, meaning resources that exist but *do not appear to be* connected to any part of the tree.
3. Calls to unresolved or off-site resources, which are references to Internet Uniform Resource Location specifications, and also any calls that cannot be connected to a known resource.

List (1) is a reassurance that there are resources that both exist on your site and that are referenced in your pages.

List (2) show any resources that do not appear to have references. The Site Analyzer cannot detect all uses of a resource, so don't just delete all the items on this list without first trying to determine if they are in use.

Site Analyzer won't detect references to resources inside CGI scripts.

Site Analyzer also cannot detect the content of any subdirectories that follow the Web naming

convention that only requires the name of the directory. In this case, the browser detects a file named "index.html" on its own and launches it. To solve this problem, just add the page name to the end of your calling tag -- instead of calling for "mysubprogram," call instead for "mysubprogram/index.html." The browser will treat these two calls the same way, but Site Analyzer is then able to add that "branch" to its tree.

List (3) generated by Site Analyzer contains all off-site references in your pages (mostly Internet sites), sorted alphabetically, so you can easily test their validity using your browser.

When the analysis is complete, a resizable dialog box opens. You may click on one of the categories in the same way you open a directory in Explorer. The first tier (or level) consists of the categories mentioned above. The second tier consists of actual resources or resource names. The third tier, by far the most interesting, lists the HTML pages that called the resource.

The third tier is by far the most useful aspect of the Site Analyzer. If you want to see the original resource reference, just click on the page's name in the third tier, and that page will be opened and the cursor will be placed next to the original call. This arrangement is particularly useful in list (3) (calls to unresolved or off-site resources), because this is where typographical errors tend to wind up. This feature makes it easy to correct such errors interactively, by opening each page and correcting links that have no destination.

In the "Unreferenced Resource" list there obviously is no third tier, so if you click on a resource in this list, you are offered the option of deleting the resource. Be careful, for the reasons outlined above -- there are many reasons why Site Analyzer might not find a reference call.

After you have made a number of editorial corrections, you will want to regenerate the list -- just close the Site Analyzer dialog and re-select Tools ... Analyze Site.

Beautify HTML / Analyze HTML

The Beautify HTML / Analyze HTML system consists of a group of functions and dialogs that can:

- Clean up the appearance of your HTML pages, making them easier to maintain and extend, and
- Locate and help you correct HTML syntax errors.

The second function may turn out to be very important on complex pages with syntax errors – such pages may never display the way you expect, and (if Netscape is the target browser) may not display at all until the errors are corrected.

You may launch Beautify HTML / Analyze HTML by choosing Tools ... Beautify ... Analyze HTML or by pressing Ctrl+L.

If you simply want to beautify your page and do not expect errors (who does?), simply press Ctrl+E (beautify HTML). If it turns out there are errors in your page, the Analyze HTML dialog will launch automatically.

If any errors are found, a second dialog with an error list will be launched to assist you in locating errors.

*NOTE: If you see unexplainable errors, be sure to use the **Fix line endings** function on the Tools menu, then try again. Improperly terminated lines will create apparent syntax errors.*

Analyze HTML detects three classes of errors:

Error Class 1 -- Unpaired Tags

Unpaired tags are HTML tags that should have a “partner” but do not. In this example:

```
<HEAD>
  <TITLE>
    My Page!
</HEAD>
```

there is a <TITLE> tag, but no “partner” </TITLE> tag.

Unfortunately, because of the nature of this error, and because of the potential complexity of your page, it is not possible to determine which line contains the error. You must establish the location of the missing tag using other methods.

One way is to use the Class 2 error list (below) to locate a line that should have the missing tag but does not.

NOTE: Be sure to correct all Class 1 errors before trying to locate and correct Class 2 and 3 errors – these other error lists are not entirely valid until Class 1 errors have been corrected.

Error Class 2 – Mismatched Tags

In this error class, a tag is not matched to its partner at the same indentation level. Here is an example:

```
I want both
<B>
  <I>
    bold and italic
  </B>
</I>
for my phrase.
```

This HTML code is generally accepted by browsers, but it is not syntactically correct, because there are “crossed tags,” tags that are out of sequence. Here is the correct version:

```
I want both
<B>
  <I>
    bold and italic
  </I>
</B>
for my phrase.
```

Note the order of the opening and closing `` and `<I>` tags.

This may seem like a trivial error, but Class 2 error detection has a much more important purpose – it can locate tags that must not be placed as they are, that will result in incorrect display or no display. Here is an example:

```
<TABLE>
  <TR>
    <TD>
      This is my table data.
    </TR>
  </TD>
</TABLE>
```

In this seemingly acceptable HTML table, the `</TR>` and `</TD>` tags are reversed. This will result in unpredictable display behavior and should be corrected. This tag error will appear on both the Class 2 and Class 3 lists.

NOTE: When using Arachnophilia’s RTF File Import feature, there will be crossed-tag errors that cannot be avoided, because of the internal formatting methods used in word processing

documents. These errors should be disregarded.

Error Class 3 – Misused Tags

This error class displays tags that cannot be used as they are, as in the table example above. Some tags must be “nested” within other tags to function properly, and will result in display errors or a nonfunctioning page if they are not placed correctly.

Locating Errors

The easy way to locate Class 2 and Class 3 errors is simply to click the error messages in the error list dialog. This will take you to the line containing the error in your document.

Another useful technique is to use the editing cursor to identify a particular indentation level, and then scroll up or down through your document to locate its “partner,” if there is one.

Use Selected Tags

This powerful feature allows you to select a “problem” tag or tags from a dropdown list and then press “Analyze” to indent the document based only on the tags you have chosen, instead of all tags. This helps identify lines that lack a particular tag by indenting only those lines having the tag or its partner.

To assist in this process, you may change the amount of indentation for each level by pressing “Bigger Tab Stops” and “Smaller Tab Stops.”

Beautify Display Problems

Sometimes, after using Beautify HTML, your page will not display as it did. This is because there are some page formatting choices that can result in small display differences. To solve this problem, simply retain the un-beautified version of the page and discard the beautified version after you have located and corrected any syntax errors.

Always test the beautified version of your page on a browser before overwriting the original version.

Beautify Code

Nothing looks worse than a computer program with ragged indentation. Have you ever wondered how programmers get their program listings to look so orderly? Most use a program available on UNIX called “cb,” meaning “C Beautifier.”

But this program isn’t generally available outside the UNIX world, so I decided to add one to Arachnophilia. If you have one of several file types on display, you can clean up the code with a click. Here are the valid file types:

- HTML and ASP pages with JavaScript and JScript in them.
- Perl programs.
- CGI programs.
- C and C++ programs.

The beautifier behaves differently for each of these, but it will produce a nice-looking result for all. Just put your program on display and select Tools ... Beautify ... Code, or press Ctrl+H. And, if you don’t like the result, just Undo it.

There is one error message that can be generated by the Beautifier -- “Indentation Error.” This error arises when there are unmatched braces or parentheses. There must be an equal number of “{ and “}” symbols, and “(“ and “)” symbols in the script to avoid this error. Therefore this consistency check is a simple check of the script’s overall syntactical correctness. Any errors detected by the Beautifier would also be caught by the language compiler or interpreter later, so this saves time and effort.

Locating the syntax error is quite simple -- just follow the script until the text seems to be too far from, or too close to, the right margin. Directly above will be the site of the error.

The Code Beautifier works with Active Server Pages than contain code blocks marked off with <% and %>. To assure correct formatting, please place these special tokens on separate lines.

If you enter this:

```
<% if(x) { %>
```

the ASP interpreter will be quite happy, but Arachnophilia will become confused and report an error, because it will not see the code. Instead, enter this example as:

```
<%  
if(x) {  
%>
```

This will allow Arachnophilia to format the page correctly and keep an accurate count of the delimiters.

This rule does not apply to another common ASP notation:

You are visitor number <%=VisitorCount%>

This example is OK because it does not contain any delimiters that need to be counted.

The Commands Menu

Most of this menu is a duplicate of the functions offered by the user-defined and default toolbars. It is mostly included for compatibility with keyboard-only operations (which I confess I haven't tried), and for those who prefer to browse a menu tree.

In each submenu you have the option of launching the associated toolbar, so it is an easy way to review the contents of those toolbars, as well as gaining access to commands you don't use very often.

The Preview Menu

The preview menu allows you to activate the selected browser, or quickly switch browsers. The Launch Selected Browser function is duplicated on the toolbar.

You may identify any browsers to the program, or you may use the internal browser. The internal browser shows up in the program window just like a document, and allows you to see changes you have made very quickly, even on each keystroke if you wish (see Instant View Mode below).

If you don't have Microsoft Internet Explorer 4.0 installed on your system, you may not be able to use the internal browser, depending on which other programs and accessories you have installed on your system.

At the bottom of this menu is an option to identify browsers. This is how you add browsers to your Arachnophilia installation. You may identify up to six browsers. Using multiple browsers is good practice, because no two browsers act exactly the same. You may want to change the coding of your pages to accommodate these differences.

Arachnophilia uses a feature of Windows called DDE (Dynamic Data Exchange) to switch from the main Arachnophilia display to the browser display without requiring the user to relaunch the browser each time. Basically, after launching your browser the first time, you may simply press the main toolbar preview button and the browser's display is updated with the new information. This system works on nearly all browsers -- on one older version of Microsoft Internet Explorer (2.0), because of a coding error, you have to touch the browser's title bar to refresh the display. For owners of this browser, I recommend an upgrade, free at the time of writing.

***NOTE:** If your browser fails to launch during preview, Arachnophilia will ask you to disable automatic launch. If you choose this option, you may later undo this choice by returning to the Identify Browser menu and re-enabling DDE, which is the communication mechanism that allows automatic launching.*

Instant View Mode

The internal browser permits very fast preview updates on the internal browser, in fact if your document is not too large, on each keystroke if you choose. Choose "Instant View Mode" to enable updating on each keystroke (this feature is only available on the internal browser). Set up your display so your active document and the internal browser are displayed at once (use Windows ... Tile Horizontally, as just one approach), then type in the document and watch the changes in the preview window. If you are editing a large document, Instant View may be too slow, so remember to disable this feature for large files.

This is a very powerful way to change your page's contents dynamically. If you change a color or a font size, you see the result instantly. If you change the formatting of a table, you see how things turned out right away. And you can easily find errors, because the effect of every typed character is previewed instantly.

This feature will not work on large, complex files, because on such files the browser's update time is slower than an average person can type. Be sure to disable Instant View to work on large files.

Internal Browser Settings

This submenu allows you to control the behavior of the internal browser. It includes font size and window size options. The second of these options is a very useful way to preview your page as an average visitor might see it, perhaps with a window size smaller than that used in development.

The Window Menu

The Windows menu allows you to arrange opened documents in several ways. It also lists open documents and allows you to switch between them.

Creating Frames

A complete explanation of frames is far beyond the scope of these help pages, but you can get started with a simple example, and then you may complete your education using the resources available in bookstores and the Web itself.

The use of frames makes it possible to break the browser window into several smaller windows, thus allowing you to present complex information in a flexible, accessible way. You may wish to display an index of resources while showing each selected resource in a separate window -- frames let you do this.

The Frames toolbar (of the default toolbars) has the most important tags for frame creation. At the time of writing, there are still many browsers that do not support frames, so it is a good idea to include a no-frames page along with your frames-capable page (see example below).

The first tag to understand is called `<Frameset>`. This tag defines either rows or columns, depending on which option is chosen. If you write `<Frameset rows="20%,20%,*">`, you will define three rows, the first two each occupying 20% of the browser's height, and the third row (marked with `*`) occupying the remainder of the space.

A command of `<Frameset cols="20%,20%,*">` does exactly the same thing, except in this example, columns are defined instead of rows.

You may create multiple columns and rows by nesting the `<Frameset>` tags. Here is an example:

```
<HTML>

<Head>
<Title>Frames Example</Title>
</Head>

<Frameset rows="20%,*">
<Frame src="toprow.htm" name="title_area">
<Frameset cols="20%,*">
<Frame src="leftcol.htm" name="index_area">
<Frame src="display.htm" name="display_area">
</Frameset>
</Frameset>

<Noframes>

(Put a no-frames version of your page here)

</Noframes>

</HTML>
```

On a frames-capable browser, this example would produce a display with a title row at the top, an index column at the left, and a large display area at the lower right. By the way, most frames-capable browsers will refuse to display anything until the subsidiary pages have been written.

A typical use for this frame setup is to change the contents of the *display area* by clicking links in the *index area*, while keeping both in view. Each link would look more or less like this:

```
<A href="choice.htm" Target="display_area">Make this choice!</A>
```

Remember when we set up the frames in the above example? We remembered to use the "name" option to give each window a name. Now we can refer to the windows by name from any other pages we create. This greatly increases the control you have over the frame system and also reminds you, with a well-chosen name, which window is the target for each page.

Obviously, using frames means that you have many more pages to write, but each will probably be smaller. The first page sets up the frame system and usually provides for a non-frame version of its contents. All subsidiary pages use the frame system that is set up in the first page.

This has been only a brief introduction to the subject of frames -- there are more frame tutorials and resources at www.netscape.com and www.microsoft.com as well as elsewhere on the Web.

An Introduction to JavaScript

Java is a programming language that can be used to deliver platform-independent applications across the Web. At present, the language has two forms: Java, which is a compiled, highly structured language for large projects, and JavaScript, which is a scripting language that requires much less effort than full Java. In exchange for this convenience, JavaScript has some limitations that exist mostly for security.

With JavaScript, you can create interactive programs that provide immediate results without requiring separate CGI scripts or multiple pages -- the results appear right on the page that carries the program! Or, if you prefer, you can launch separate pages and applications for more involved tasks.

The JavaScript program source is delivered along with the Web page that carries it, so JavaScript is presently regarded as public domain -- you cannot protect the source code for your JavaScript program. JavaScript also cannot open, read or write files or make system calls -- this is why it is relatively secure.

Once again, there is much more to the JavaScript language than can be covered in these help pages, but I will provide a few examples to get you started, then you can complete your training using textbooks and web resources.

The Frames toolbar (of the default toolbars) has several shortcuts to aid JavaScript program development. The first is "script," which, when activated, looks like a combination of HTML tags:

```
<Script LANGUAGE="JavaScript">
<!-- Hide this from older browsers

(the JavaScript program goes here)

// end hide -->
</Script>
```

The `<Script></Script>` tags set the JavaScript program apart from the rest of the HTML code. But, since there are many browsers that do not support JavaScript, you may want to hide the JavaScript code from them. The comment tags accomplish this, so the JavaScript program listing won't spoil the appearance of your page.

Now let's just do something silly to get started. Position the cursor in the middle of the Script block and press the "Alert" button on the Frames/Java toolbar. This will be the result:

```
<Script LANGUAGE="JavaScript">
<!-- Hide this from older browsers

alert("|");

// end hide -->
</Script>
```

As in previous examples, the vertical bar "|" shows the position of the cursor. Now Type "Hello World!" and press the Arachnophilia preview button. You will see a dialog box with the message "Hello World!"

Now let's write a more complex program. Move the cursor above the "alert" line and press the "prompt" button on the Frames/Java toolbar. You will see this:

```
<Script LANGUAGE="JavaScript">
<!-- Hide this from older browsers

var response = prompt("|", "y/n");

alert("Hello World!");

// end hide -->
</Script>
```

Now type "Enter a phrase:" where the cursor is located and then change the "alert" line to read:

```
alert("You Typed: " + response);
```

Then press the Arachnophilia preview button. You will see a prompt box that asks you to type something, then, after you have, you will see an alert dialog that will show what you typed.

There is nothing sacred about what the "prompt" button places in your program -- it is just a typical entry. For example, you could use any variable name instead of "response," and you might want to say "Type your response here" instead of "y/n" in the default area. But as you learn more about JavaScript, you will understand this -- and you will find many more things that JavaScript can do!

JavaScript has been designed to resemble the C++ programming language as much as possible, to accommodate the many people who have learned that language. Thus, if you are familiar with C++ you should feel right at home with JavaScript.

JavaScript can do many things not covered in this brief tutorial, so I recommend that you browse your local bookstore and the web for references to JavaScript. Also, because the source code of JavaScript-equipped Web pages is located right in the page, you can search for well-designed JavaScript applications and simply download them to see how they were written.

A sample of JavaScript applications can be found at <http://www.arachnoid.com> (the author's site) and tutorials and resources are available (or should become available shortly) at www.netscape.com and www.microsoft.com. Or you can simply use your favorite web search tool with the search string "JavaScript."

Working with Front Page

Microsoft Front Page® does some things very nicely. There are also some things it won't do at all, and finally (in version 1.1) there are some syntactically correct HTML tags that Front Page will simply destroy.

When one imports existing HTML code into Front Page, one must change many parts of otherwise acceptable HTML code to prevent Front Page from arbitrarily changing or losing parts of the original code.

Arachnophilia can work with Front Page and minimize some of the latter's undesirable behaviors. The first step is to make Arachnophilia one of the editors that Front Page uses for special tasks. To do this, run Front Page Explorer, select Tools ... Configure Editors, click Add, enter a file type of "*.htm", Type "Arachnophilia" for "Editor Name," then locate Arachnophilia using the provided browser function.

The second part of the process is to import your existing HTML pages into the Front Page environment, a complex process beyond the scope of this document. Refer to the Front Page help screens for more on this process.

Once having imported your pages, you will need to find some types of HTML code that Front Page will object to and then change in ways you probably won't like. Here is an example:

I use an image tag to produce a horizontal line. It looks like this:

```
<IMG src="redchip.gif" alt="" height=2 width=95%>
```

This tag uses a small image (that can be as small as 1 by 1 pixel) to make a color horizontal line. My horizontal line automatically adjusts to the width of the screen, just like the default HTML horizontal line tag, but it can be any color you desire. This tag is completely acceptable HTML code, it works in all recent browsers, but Front Page will find and eat this tag, forcing the image to have the width and height of the original graphic file, thus destroying its usefulness as a line.

To solve this problem, I have included a special tag called a "bot" (Microsoft's term) that essentially hides selected HTML code from Front Page. The Bot tag is located on Arachnophilia's "Struct" default toolbar. When the Bot tag is used, the result looks like this:

```
<!--VERMEER BOT=HTMLMarkup StartSpan -->  
<IMG src="redchip.gif" alt="" height=2 width=95%>  
<!--VERMEER BOT=HTMLMarkup EndSpan -->
```

This tag can be applied to any HTML code that Front Page objects to, but that you want to keep. And believe me when I tell you, there are plenty of tags that Front Page will have for lunch. At the time of writing I have used Front Page for only a few hours, but most of that time has been spent trying to keep Front Page from disabling some of the best parts of my Web pages.

Editorial comment: Front Page is a very ambitious project that will make HTML page creation

available to many people otherwise daunted by the complexity of HTML page development. But, like so many applications, it overemphasizes the designer's wishes and almost completely ignores the user's wishes. In many cases, Front Page will take syntactically correct HTML code and render it nonfunctional in attempting to fit the code into Front Page's narrow conception of what is acceptable HTML code.

Arachnophilia emphasizes the opposite -- it takes into account the needs of the user by providing field-customizable toolbars, as just one example.

Basically, with Front Page, Microsoft's software designers are in charge. With Arachnophilia, *you are in charge* (if you want to be). But Arachnophilia can't do some of the things Front Page can do.

The methods on this page provide a way for Arachnophilia to work with Front Page and (to some extent) keep Front Page under control.

The terms "Microsoft" and "Microsoft Front Page" are registered trademarks of Microsoft Corporation.

Autocopy Mode

Arachnophilia takes care of a difficult issue in Web page development automatically. This feature, called Autocopy, allows Arachnophilia to automatically copy your resource selections to your working directory. This has two effects:

- All the selected resources for your page end up in one directory. This is a reliable way to maintain control over those resources.
- Because all the resources are in the same directory as your Web pages, you may use relative links instead of absolute links. This is a way to make your pages more reliable and more portable, because you can move the entire package to any site and it will work as intended.

Instead of a link that, because it is located in another directory, looks like this:

```
<IMG SRC="c:\data\network\mywebpage\mypicture.gif" >
```

, using Autocopy, yours will look like this

```
<IMG SRC="mypicture.gif" >
```

because the resource is located in the same directory as the page that refers to it.

The Autocopy mode requires that two conditions be met:

- You must choose a working directory for your HTML pages, and
- You must enable Autocopy when Arachnophilia asks you to.

The first time you try to attach a resource file to your page, if you haven't yet saved your Web page, Arachnophilia will ask you if you want to save it. If you respond with Yes, (highly recommended) then Arachnophilia will ask whether it can enable Autocopy.

After Autocopy has been enabled, no matter where you find your resources, they are copied into your working directory and their names are converted to the relative form.

The best part of Autocopy is that, when you have created and tested your Web page(s) at your personal computer, you can upload the entire package to your Internet Service Provider and it may work perfectly the first time. This is not an everyday occurrence.

The single exception to the Autocopy method is a link to a local HTML page. This type of resource will not be copied, because the page you have linked to may have its own links, and if the page is moved its links will stop working. For this reason, you should create all your HTML pages within the directory structure that you intend to maintain when you have uploaded your site onto the Internet.

HTML Page Setup

This dialog box appears when you create a new HTML page. You can use it to choose the color of text and all the HTML tags. You can choose a title for your page, the title that appears on the title bar of the browser that visits your site.

Your choices are preserved between uses of Arachnophilia, so you only really have to state your preferences once.

All the choices in this dialog can be edited later, so it won't hurt to pass this up by pressing "OK" if you aren't interested in sorting out the meanings of these options.

File Types and Suffixes

Web server programs use the suffix of a file to determine what kind of file it is. For example, an HTML page might have the suffix `.html`. An example of a file using this suffix would be “`mypage.html`”. A graphic file might have the suffix `.jpeg` or `.gif`. Arachnophilia uses these standard suffixes also.

Some environments use non-standard file suffixes, therefore Arachnophilia may recognize more than one file suffix for the same file type. Here is a chart of the file suffixes recognized by Arachnophilia:

File Type	Recognized Suffixes
HTML	<code>.html, .htm, .shtml, .asp, .stm, .idc, .htx, .ssi, .cfm</code>
Text	<code>.txt, .text, .doc, .bat, .log, .tbd</code>
CGI Script	<code>.cgi</code>
Perl Script	<code>.pl, .perl, .pm</code>
Sound File	<code>.mid, .rmi, .wav, .ra, .ram</code>
Graphic File	<code>.gif, .jpg, .jpeg, .png</code>
URL File	<code>.url</code>
RTF Text File	<code>.rtf</code>
C, C++ Source File	<code>.cpp, .c, .cc, .h</code>
Java Source File	<code>.java</code>

In Arachnophilia 3.0+, you may also disable HTML detection, so that any file is assumed to be a valid HTML source file. See Tools ... Options ... Miscellaneous.

The Arachnophilia Template System

When you open a new file in Arachnophilia, the program uses a file template appropriate for that file type. The templates are located in the “templates” subdirectory of the Arachnophilia program directory. You can customize the templates to meet your personal requirements.

Some of the templates contain special labels that allow Arachnophilia to customize the file’s content when it is opened. For example, here is the default HTML file template (before opening):

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">

<HTML>

<HEAD>
<TITLE>[arach_title]</TITLE>
<META NAME="GENERATOR" CONTENT="[arach_version]">
<META NAME="FORMATTER" CONTENT="[arach_version]">
</HEAD>

<BODY BACKGROUND="" BGCOLOR="[arach_bg]" TEXT="[arach_text]"
LINK="[arach_link]" VLINK="[arach_vlink]" ALINK="[arach_alink]">

|

</BODY>

</HTML>
```

In this template, the vertical bar ‘|’ indicates the position that will be given to the editing cursor when the file is first opened. The labels enclosed in [brackets] are special, reserved names that Arachnophilia uses to customize the file when it is opened. These labels are unique to the HTML file type. All of the file types that use templates also accept the label [arach_filename], so that the default filename provided on opening can appear in the file. For example, here is the default C++ file template:

```
// [arach_filename]

void main(void)
{
    |
}
```

If you don’t want the default filename to appear within the file, simply remove the special label. This applies to all file types that use templates, and also to the special labels used in the HTML file template – you can remove any that you don’t want. If you DO want a default filename to appear within an HTML file, use this format:

```
<!-- [arach_filename] -->
```

This allows you to add the default filename while conforming to the requirements of HTML syntax.

NOTE: If you create your own personal templates, be sure to back them up to a separate directory. This will prevent their being erased if you update to a new version of Arachnophilia.

Here is a list of all the special labels and their meanings:

All Files

<i>Label</i>	<i>Meaning</i>
[arach_version]	“Arachnophilia Version (number)”
[arach_filename]	The temporary file name given to this file, “untitled(number).suffix”

HTML Files only

<i>Label</i>	<i>Meaning</i>
[arach_bg]	Background color provided by the HTML setup dialog – see HTML Page Setup
[arach_text]	Text color as above
[arach_link]	Link color as above
[arach_vlink]	Visited link color as above
[arach_alink]	Active link color as above

The Table Wizard

The Table Wizard has two modes: (1) it will create an empty table with a specified number of rows and columns, or (2) it will automatically create a table from a selection of data in your document. Here are the details:

Manual Table Creation

Place the editing cursor in your document where you want the table to go (without selecting any text), and choose Tools ... Make Table. Select the number of rows and columns you want in your table, choose a background and border color if you wish, and set the other options. Press "Create." The table will be created at the location of the cursor in your document.

If you decide to change the table, simply press the left arrow on the toolbar (Power Undo) to recover the original document, make the changes you want, and press "Create" again.

Automatic Table Creation

The Table Wizard has a rather remarkable feature that will automatically turn a block of data into a table. You might have a block of data that you have created, or you may wish to paste a section of a spreadsheet directly into your Arachnophilia document. There are any number of Windows applications that will place their table-oriented data on the clipboard, separated by commas, tabs, or some other delimiter character. You can make tables out of these with a few keystrokes.

Let's start with a simple example. Open an HTML document and type in the following data (or simply copy it from this page):

```
State, Capitol  
California, Sacramento  
Oregon, Salem  
Washington, Olympia  
Idaho, Boise  
Montana, Helena
```

Now select this block of text by dragging the mouse across it and launch Table Wizard. Table Wizard will collect the information from your block of text and prepare to make the table. If you see that it intends to make a 1-column table, this means the wrong "delimiter" has been chosen. The "delimiter" is the character that separates individual items of information in the rows in our example table. In this example, the delimiter is a comma, so use Table Wizard's Data Delimiter dropdown list to choose this character, then press "Refresh" to allow Table Wizard to recalculate the table values. Now you will see that Table Wizard is ready to create a 6 x 2 table.

In our example table, the top row contains names for the data, so choose "Bold First Row" to emphasize this line. You may also want to choose "Titlecolor First Row" as well, to color the background of the title row, and choose a color with the "Title row/col" button. Now press "Create." Table Wizard will automatically create the table you have specified. If you don't like

the outcome, simply press the toolbar “Undo” button (left arrow), change your settings, and try again.

Table Wizard will create tables with any number of rows and columns, and it will warn you if some of the rows have fewer data cells than others. In this case, Table Wizard will “pad” blank data into those rows that are short, in order to make the table consistent in size from row to row.

Here is another application -- you may wish to place a simple box around some text, using a table with a single data cell. For this purpose, simply select the text and launch Table Wizard. If there are no line breaks in the block, and if you choose the special delimiter “(none),” Table Wizard will create a simple box around your text. If there are line breaks in the block, each will be placed in its own row in the table. As you experiment with Table Wizard, you will realize it can do many such tricks and will save you a lot of time in page design.

Remember: if you change the data in your table, or if you change your choice of delimiter, you will need to press “Refresh” to allow Table Wizard to resample the document before creating the table.

Also: To prevent the generation of a “width” definition in your table tag, simply set the width value to zero.

The List Wizard

The List Wizard help you make lists. In HTML, a list looks like this:

1. Animal
2. Vegetable
3. Mineral

This is called an “Ordered List” – it has a *sequence* of identifying marks. The other kind of list is called “Unordered.” Here is an example:

- Animal
- Vegetable
- Mineral

Notice how this list has marks that are not in a sequence.

List Wizard lets you create both kinds of lists. You can create a bare list with a specified number of items using the **Manual Mode**, or you can select some lines of data in your document and let List Wizard create a list for you using the **Automatic mode**.

Manual Mode Example:

Position the editing cursor in your document where you want the table to go. Launch List Wizard (Tools ... List Wizard). Choose the style you want and the number of items and press “Create.” Here is a typical result:

```
<OL type="1">  
  <LI> Item 1  
  <LI> Item 2  
  <LI> Item 3  
  <LI> Item 4  
</OL>
```

On a browser, this list would look like this:

1. Item 1
2. Item 2
3. Item 3
4. Item 4

Automatic Mode Example

The real value of List Wizard is in creating lists from existing data. Let’s say you have this list:

Animal

Dog
Horse
Bear
Vegetable
Carrot
Leek
Radish
Mineral
Quartz
Feldspar
Granite

Reading this list, It becomes clear that we need a main list with some sublists for the examples of each main category – “Dog, Horse, Bear” should be a sublist under “Animal.”

A digression: The best way to select text for List Wizard is to place the mouse cursor against the left margin (so the cursor points to the right) next to the top item and drag downward, thus selecting the entire line. Another way, using the keyboard, is to position the editing cursor on the left margin at the beginning of the first desired line, hold down the shift key, and use the down-arrow key to select the lines.

In both these methods, the point is to select entire lines of text, not parts of lines. Arachnophilia shows you what text has been selected by making its background dark.

In this example, I will show selected text in **red**. First, select the “Dog, Horse, Bear” sublist, like this:

Animal
Dog
Horse
Bear
Vegetable
Carrot
...

Now launch List Wizard. This time, List Wizard will detect the fact that you have selected some text and go into **Automatic Mode**. In Automatic Mode, List Wizard figures out how many items there are in the list, you don’t have to tell it. Choose “Ordered List” and select the “A, B, C” style from the dropdown list, and press “Create.” Here is the result:

```
Animal
<OL type="A">
  <LI> Dog
  <LI> Horse
  <LI> Bear
</OL>
Vegetable
```


Carrot

...

Perform the same steps for the two remaining sublists – the Vegetable and Mineral sublists. Now we have this:

Animal

```
<OL type="A">
  <LI> Dog
  <LI> Horse
  <LI> Bear
</OL>
```

Vegetable

```
<OL type="A">
  <LI> Carrot
  <LI> Leek
  <LI> Radish
</OL>
```

Mineral

```
<OL type="A">
  <LI> Quartz
  <LI> Feldspar
  <LI> Granite
</OL>
```

Now for the interesting part – we can create a master list that ties the three sublists together. Select the entire list including the “Animal, Vegetable, Mineral” category headings as well as the sublists. Select a style of “1, 2, 3” and press “Create.” Here is the result:

```
<OL type="1">
  <LI> Animal
  <OL type="A">
    <LI> Dog
    <LI> Horse
    <LI> Bear
  </OL>
  <LI> Vegetable
  <OL type="A">
    <LI> Carrot
    <LI> Leek
    <LI> Radish
  </OL>
  <LI> Mineral
  <OL type="A">
    <LI> Quartz
    <LI> Feldspar
    <LI> Granite
```

```
</OL>
</OL>
```

Now this list will display correctly on a browser, but it is difficult to interpret in the editor display, So let's beautify it – Press Ctrl+E to invoke the HTML Beautifier. Here is the result:

```
<OL type="1">
  <LI>
    Animal
    <OL type="A">
      <LI>
        Dog
      <LI>
        Horse
      <LI>
        Bear
    </OL>
  <LI>
    Vegetable
    <OL type="A">
      <LI>
        Carrot
      <LI>
        Leek
      <LI>
        Radish
    </OL>
  <LI>
    Mineral
    <OL type="A">
      <LI>
        Quartz
      <LI>
        Feldspar
      <LI>
        Granite
    </OL>
</OL>
```

This way of listing the result is obviously easier to interpret – and List Wizard can still read and change it if you desire. Let's say we want to change the style of the main list from "1, 2, 3" to "I, II, III". Select the entire list, press "refresh" if the List Wizard is still in view, or launch it if it is not visible. Choose a style of "I, II, III" and press Create. List Wizard will re-combine the list to its state before the HTML Beautifier was invoked, and make the requested change.

To select data items from a formatted list, for each desired item always select the " tag plus any additional lines that belong to it. If you simply want to change the style of an existing list,

select the list-definition tags “,” or “,” as the first and last lines of your selection.

The Site Update Tool

Arachnophilia contains a tool that allows you to move your pages and resources onto the Internet. This tool uses a method known as FTP (File Transfer Protocol).

The Site Update Tool assumes you have an internet account with an Internet Service Provider (ISP). This means you have been given some instructions about your account, such as a user name, a password, and other information.

You may keep up to eight Internet account profiles in the Site Update Tool. Arachnophilia remembers this information for you.

To use the Site Update Tool, simply load a page that is located in your project directory, then select **Tools ... Update Web Site**. If you want, you may arrange your desktop so you can see this help screen and the Site Update Tool at once.

To use this tool successfully, you must provide some information.

- The “Local Directory,” the directory on your computer where the files are located. To save time, you can specify this directory by simply loading one of the pages from the desired directory into Arachnophilia before activating the Site Update Tool. Then, when you use the Site Update Tool, you simply press “Use Current Directory.”
- The “Remote Directory,” the directory where you want your pages to be located on your ISP’s Web server. This directory might have the name “public_html,” or another similar name. In the event of difficulty, your ISP will be able to tell you the correct name.
- The “Server Name.” This is the name that your ISP uses to identify the server computer’s FTP service. For example, if your ISP is called “myserver.com,” chances are you will use “ftp.myserver.com” as an FTP server name. If this name doesn’t work, you may have to ask your ISP how to access their FTP uploading service.
- Your “User Name,” also known as your logon name.
- Your password. You have the option of saving your password or entering it each time you use the Site Update Tool.

If you get into difficulty using the Site Update Tool, you should ask your ISP for help. Some old-fashioned ISP services require you to give them the pages you want uploaded. In any case, file transfers using FTP are more complex than simply writing a home page – you have to know more technical details.

Some Details

The detailed procedure for using the Site Update Tool is as follows:

- Create the pages and resources you want to upload onto the Internet. Save them in a special, separate directory that has been set aside for your Web site. In this example, let’s say you have created a directory called `c:\web\project_dir`.
- When you create directories and pages, be sure that there are no spaces in the names. Internet

servers won't tolerate spaces, and the Site Update Tool therefore cannot either. For example, instead of calling a file "my favorite sites.html," call it "my_favorite_sites.html" instead.

- Pay attention to the case of the file names of your pages, graphics, and other resources. This is one of the most common problems in Web page development. The Windows operating system does not care what case you use for a file name – the names "MYPIC.GIF" and "mypic.gif" are treated the same. But on a Web server, case matters. You may find that your page will work perfectly on your home machine but not at all when you upload it – *file name case is often the reason*. You should make a policy to be consistent about case, and lowercase is generally preferred.
- Load a page from your site directory into Arachnophilia. This is just an easy way to tell Arachnophilia where your files are.
- Select Tools ... Update Web Site. A dialog will appear (see above) awaiting your information entries. Be sure to select your project directory by pressing "Use Current Directory."
- The "Remote Directory" item should be filled in with the name of your HTML resource directory with your ISP. A typical name is "public_html."
- Now enter your user name, otherwise known as your logon name.
- Enter your password.
- If your site has subdirectories, you may want to include those subdirectories in your upload. To do this, choose the option "Include Subdirectories."
- At this point, you may want to *activate your modem* and log onto your ISP's server before pressing the "Update" button.

Upload Strategies

The Site Update Tool uses one of several strategies to save time in performing file uploads. The point of these strategies is to only upload files that are changed. Another point is to access the local machine for file information, not the Web server -- this saves a lot of time.

- The Archive Method. The most efficient way to detect changed files is to sample their "archive bit." The archive bit is part of the directory entry for a changed file. If you choose the Archive method for updating, Arachnophilia will upload all changed files and then reset the archive bit on each uploaded file, thus preventing a repetition of the same upload later. But if you make further changes to that file, the archive bit will alert Arachnophilia that it needs to upload the file again.
- 0 It is important to realize that you can manually set or reset archive bits using File Explorer, either one file at a time or for an entire directory. See your Windows documentation for more on this method.
- The Age Method. Some users will have other purposes for the archive bits and will not want to reset them. Another strategy is to select an age for new files. In this method, if you select an age of 24 hours, all files changed within the most recent 24 hours will be uploaded onto your site. You can choose any age for this method, from one minute to several weeks worth of hours.
 - Upload All Files. This final method is useful when you are just setting up a site. Upload All

Files takes longest, but all files are certain to be uploaded.

The Site Update Tool creates any directories it needs to as it updates the site. If you create new directories at home, Arachnophilia will create and fill identical directories on your Web site.

The only thing you can't do with The Site Update Tool is delete files from your Web site. For this you need a program called an "FTP client." Many are available, and some are even free.

More Details

Here are some general notes about FTP and The Site Update Tool. At risk of repeating myself, when you are designing your site, always be consistent about the case of file names, that is, the HTML tags that call for files should be in the same case as the file names themselves. This is a common problem when uploading for the first time – if you pay no attention to case on your local machine, your page will still work, because Windows doesn't care about case. But UNIX servers do care.

Another issue is translation. Windows uses a rather peculiar internal format for text files. At the end of each line, there are two control characters – a carriage return and a line feed. On other types of operating systems like UNIX, there is just one – a line feed. This means that a program that uploads files to the Internet has to pay attention to the file type being uploaded. If it is a text file, it must be translated. If it is a graphic or binary file, it must be copied exactly.

Arachnophilia automatically determines the file type before uploading, and so long as you use one of the file types recognized by Arachnophilia, this method will successfully cooperate with your Web server computer.

Troubleshooting & Common Problems

Speed Problems

If your system responds slowly to keyboard input, if the display falls behind your typing or doesn't seem to be in touch with the present time, here are some remedies.

1. Turn off Instant View (**Preview ... Instant View Mode**). This feature requires a great deal of processing power.
2. Turn off Automatic Tag Coloring (**Tools ... Auto Tag Coloring on/off** or **Ctrl+Shift+T**). This feature also requires substantial processing time, although not as much as Instant View.
3. Turn off Power Undo (**Tools ... Options ... Miscellaneous ... (unselect) Enable Power Undo**) – on some systems with slow hard drives, this feature requires a lot of time. Disabling this feature also saves memory.

Memory Problems

If you receive memory error messages from Arachnophilia or Windows, here are some remedies.

1. Turn off Power Undo (**Tools ... Options ... Miscellaneous ... (unselect) Enable Power Undo**) – on some systems with little spare hard drive space, you will run out of virtual memory and memory error messages will begin to appear.
2. Install a larger hard drive and/or more RAM. Both these steps will increase program speed and capacity.
3. Build smaller Web pages. This remedy is also a public service – large Web pages can almost always be advantageously broken into smaller sections.
4. Work on fewer pages simultaneously.

I can't use the internal browser

Use of the internal browser requires that Microsoft Internet Explorer 4.0 or newer be installed on your system. I would have provided a way for all browsers to be so treated, but only one allows this method.

External browsers now keep launching new copies of themselves

This happens if you respond to a particular dialog asking if you want to disable auto-launch. To restore the original settings, use menu option **Preview ... Identify Browser**, choose the problem browser and go through the identification steps again. Be sure to check "This browser is DDE capable."

I can't use some automatic editing features on some pages, or Beautify HTML reports nonexistent errors in some of my pages.

This happens because some pages, in particular those that were downloaded from an Internet server, have incorrect line endings. This is a technical matter that will probably bore most people,

so simply load the page in question and choose **Tools ... Fix Line Endings**. This function will report how many incorrect line endings it has fixed, if any. Be sure to save the page after this repair has been made.

If I select a group of documents using the right-click “open with Arachnophilia” feature, many copies of Arachnophilia are launched.

Be sure that Arachnophilia is already running before selecting more than one document with the right-click feature.

I selected some document types with the “Open with Arachnophilia” feature (Tools ... Open with Arachnophilia), but I don’t see Arachnophilia on Explorer’s right-click menu.

In the **Tools ... Open with Arachnophilia** dialog, you must press “Apply” after selecting the document types.

I want to choose particular files to upload with the Site Update Tool.

No problem – here’s how:

1. Launch Windows Explorer, the file browsing tool in Windows 95/NT.
2. Select the files you want to upload by pointing with the mouse. You may press the Shift or Ctrl key to select multiple files.
3. Press the right mouse button to launch Explorer’s "context menu."
4. In this menu, choose "Properties."
5. Enable the "archive" attribute at the bottom of the dialog.

Now, if you select the Archive update method, the Site Update Tool will upload the files you have selected.

How do I switch between maximized screens in Arachnophilia?

Use the **File Bar** to choose document windows. Enable it using **Commands ... File Bar**.

Or, if you prefer using the keyboard, use **Ctrl+Tab**.

Using these methods, you can use the Internal Browser in full screen mode, yet easily switch back to your document.

Other interesting keyboard shortcuts are listed in **Tools ... Lists ... Keyboard Shortcuts**.

I want to convert a plain-text file to HTML.

Since a plain-text file has no special formatting information, all you need to do is provide line breaks that a browser will recognize. Here’s how:

1. Paste the plain text into a new, empty HTML document between the `<BODY>` and `</BODY>` tags.
2. Select the text block.
3. Using the **Selection ... Find/Replace** dialog, replace all “\p” with “
\p” (without the quotes).

In some cases you can be lazy/clever and simply enclose the text block with `<PRE> ... </PRE>` tags. This method will not work for long lines, such as paragraph-formatted text, because the lines will not automatically wrap around within the browser’s window. The first described method is the best general solution.

How do I transfer a complete tag-colored version of my page to a general-purpose word processor such as Word?

1. Turn off Automatic Tag Coloring (**Tools ... Auto Tag Coloring on/off**).
2. Tag-color the entire document (**Tools ... Refresh Document Tag Colors** or Ctrl+Shift+A).
3. Press Ctrl+A to select the entire document.
4. Press Ctrl+C to copy the entire document onto the clipboard.
5. Run the word processing program, open a blank document, and press Ctrl+V (Paste).

Word processing programs that accept RTF (Rich Text Format) documents and clipboard contents will show the formatting (such as Beautify HTML indentations) and the tag colors.

How do I open multiple files?

Here are some of the ways:

1. Select as many files as you need in the File ... Open dialog.
2. Using File Explorer, select as many files as you need and drag them onto the open, empty Arachnophilia program window.
3. Using File Explorer, select as many files as you need and drag them onto the Arachnophilia program icon.
4. After enabling Arachnophilia’s “Open With ...” feature (Tools ... Open With Arachnophilia), in File Explorer select as many files as you wish, then open the File Explorer right-click context menu.
5. Create a shortcut to Arachnophilia with a file list included.

The CareWare Idea

*I never saw a wild thing
Sorry for itself.
A small bird will drop frozen dead
From a bough
Without ever having felt sorry for itself.*

-- D. H. Lawrence

Even though this page has a big, impressive title, it is about a simple idea which I hope I can convey in a few words.

Here are the main points:

- Economic principles lie behind many more human activities than most of us realize. We are almost constantly exchanging something for something else.
- Many economic transactions don't involve money. In traditional societies, and sometimes even this one, people trade using favors, influence, even pure ideas, instead of money.
- Sometimes money is not the best way to convey value. *And sometimes money is so completely inappropriate that it destroys the transaction.* CareWare is one of those transactions.
- CareWare doesn't involve money, but it is a transaction nevertheless. Something is delivered, something is received. Adam Smith's invisible economic hand moves through the CareWare economy just like everywhere else. I can't ask for something more than I am giving, but I can ask for an appropriate exchange.

In CareWare, the "buyer" gets something of value in exchange for something the "seller" wants. And what does the seller want? The general answer is "Anything except money," but I prefer the really remarkable transactions, which you recognize instinctively when you see them.

Here is an example -- here is a hypothetical transaction between myself and you. I have a program called "Arachnophilia" which is a rather nice Web page editor and workshop, but I don't want your money in exchange, I want something else. So I say "This is what I am offering, and here is what I want in return." Simple, right? I have been doing this for a little while now, and the responses have been very interesting. Many people think I have religious beliefs -- I don't, at least not religious beliefs recognizable by a person of normal intellectual gifts. Others try to push the transaction in a more traditional direction for one reason or another -- I recently received this message:

Careware, eh? A small price to pay for such a great software package. My pen quivers over my checkbook in case you change your mind.

This response could arise from any number of causes, and bartering about the *medium* as well as the *size* of the payment is a normal part of economic transactions, so maybe I am picking on this particular correspondent a little. But the funny part is *CareWare is by no means a small price to pay*.

For example, here is a payment I will accept for a copy of Arachnophilia --

To own Arachnophilia, I ask that you stop whining about how hard your life is, at least for a while. When Americans whine, nearly everybody else in the world laughs. We have so much, and yet we manage to:

- **Overlook great examples of beauty around us,**
- **Miss our most important opportunities,**
- **Manage to make ourselves miserable by expecting something even better to come along.**

Every time we whine about how tough we have it, apart from the fact that we look ridiculous, we make it harder for people around us to appreciate how much we have. We encourage people to overlook the things we do have, the gifts of man and nature. We provide a context to dismiss everything as not good enough, to be miserable in the midst of plenty.

Don't get the wrong impression -- many things are unjust, things that should be struggled against until they are made right. My complaint is with people who can't find even one thing to take joy in, to appreciate. These people not only make themselves miserable, but they infect others with the attitude that the world should right itself, by itself, before they will take simple pleasure in anything.

So here is my deal: stop whining for an hour, a day, a week, your choice, and you will have earned your copy of Arachnophilia. Say encouraging words to young people, make them feel welcome on the planet Earth (many do not). Show by example that we don't need all we have in order to be happy and productive.

Paul Lutus, Port Hadlock, WA

Arachnophilia Distribution Policy

Arachnophilia may be freely distributed and copied, so long as no fee is charged. Arachnophilia may not be made part of a product for which a fee is charged.

Notwithstanding the foregoing, Arachnophilia is not in the public domain and is © Copyright 1996-1998, Paul Lutus, who retains all present and future rights in and to Arachnophilia.

The determination of fitness of Arachnophilia for any particular purpose shall be solely with the recipient, and Paul Lutus shall not be responsible for any damages, direct, indirect, or consequential, that may arise from use of Arachnophilia.

HTML (Hypertext Markup Language) is the language with which Web pages are written. Originally invented by Tim Berners-Lee while at CERN, HTML is undergoing changes and improvements as time passes.

Browser: a generic term used to describe a program that can read and display Web Pages.
Examples: [Microsoft Internet Explorer](#), [Netscape](#) (click these links to acquire these browsers).

Web: a slang term for the World Wide Web, the interacting entity composed of the Internet, HTML pages posted there, and browsers.

Web Page: A document written in HTML, that allows one to interactively gain access to other resources by way of active links.

Internet: A worldwide network of computers and people.

Unix: a very popular computer operating system. Much of the present ability of Windows and other similar operating systems was derived from Unix. Unix allows multiple users to run multiple processes on a single computer, and facilitates interconnection between computers.

Object: In computer parlance, an object is a package consisting of (1) data and (2) computer program elements designed to act on the data.

Hypertext: a scheme by which one can navigate through an information resource by activating links.

No, no. Don't confuse this with REALITY. This is WINDOWS. Reality requires a Browser.

Telnet: an older Internet communication protocol that resembles a text editor screen, except that you can issue commands to a remote system in what is called a "Unix shell session."

Internet Service Provider: A company that provides Internet access to individuals and businesses. Most good providers allow several megabytes of storage on their server per subscriber, thus allowing one to post a Web page with plenty of graphic images and other resources.

Relative Link: a Web page link that presumes that its target is a local resource, rather than spelling out exactly where it is located. Also see [Absolute Link](#).

Relative Link:

Absolute Link:

Absolute Link: a Web page link that presumes that its target is not a local resource, therefore spelling out exactly where it is located. Also see [Relative Link](#).

Absolute Link:

Relative Link:

Tag: a term used in HTML programming to describe the fundamental unit of communication between the HTML script and the browser that tries to read it. Example: in an HTML script, "This is <I>very</I> important" will be shown by a browser as "This is *very* important" because the browser understands "<I>phrase</I>" to mean "Italicize all between <I> and </I>."

Rich text Format: a file format supported by many Microsoft applications that preserves the appearance of the document, but can be easily transported to another application. Arachnophilia supports Rich Text Format.

Hyperlink or Link: a reference in a Web Page that refers to another Web page or some other kind of resource. When you click on the link, you jump to what the link describes. Also see [Absolute Link](#), [Relative Link](#).

Select, Selection: A method to choose a part of a document. You most often select text in one of these ways: (1) hold down the shift key while moving the cursor across the desired text with the arrow keys, or (2) press the left mouse button and drag the mouse cursor across the desired text.

Pixel: Short for Picture Element. One of the matrix of graphic elements that make up a graphic image. A 640 by 480 VGA graphic image has 640 horizontal pixels per row and 480 vertical pixels per column, for a total of 30,720 pixels. Pixels are the basic units out of which graphic images are created -- each displays a color dot.

Arachnophilia: A fondness for spiders and Webs of all kinds. One having this condition is called an "Arachnophilic."

<<YourApp>> Help Index

How To ...

<<add your application-specific "how to" topics here>>

Commands

[File menu](#)

[Edit menu](#)

[Record menu](#)

[View menu](#)

[Window menu](#)

[Help menu](#)

File menu commands

The File menu offers the following commands:

<u>New</u>	Creates a new document.
<u>Open</u>	Opens an existing document.
<u>Close</u>	Closes an opened document.
<u>Save</u>	Saves an opened document using the same file name.
<u>Save As</u>	Saves an opened document to a specified file name.
<u>Print</u>	Prints a document.
<u>Print</u>	Displays the document on the screen as it would appear printed.
<u>Preview</u>	
<u>Print Setup</u>	Selects a printer and printer connection.
<u>Send...</u>	Sends the active document through electronic mail.
<u>Exit</u>	Exits <<YourApp>>.

Edit menu commands

The Edit menu offers the following commands:

<u>Undo</u>	Reverse previous editing operation.
<u>Cut</u>	Deletes data from the document and moves it to the clipboard.
<u>Copy</u>	Copies data from the document to the clipboard.
<u>Paste</u>	Pastes data from the clipboard into the document.
<u>Paste Link</u>	Pastes from the clipboard a link to data in another application.
<u>Insert New</u>	Inserts and embeds an object, such as a chart or an equation in a document.
<u>Object</u>	
<u>Links</u>	List and edit links to embedded documents.

View menu commands

The View menu offers the following commands:

<u>Toolbar</u>	Shows or hides the toolbar.
<u>Status Bar</u>	Shows or hides the status bar.

Window menu commands

The Window menu offers the following commands, which enable you to arrange multiple views of multiple documents in the application window:

<u>New Window</u>	Creates a new window that views the same document.
<u>Cascade</u>	Arranges windows in an overlapped fashion.
<u>Tile</u>	Arranges windows in non-overlapped tiles.
<u>Arrange Icons</u>	Arranges icons of closed windows.
<u>Split</u>	Split the active window into panes.
<u>Window 1,</u> <u>2, ...</u>	Goes to specified window.

Help menu commands

The Help menu offers the following commands, which provide you assistance with this application:


- Help Offers you an index to topics on which you can get help.
- Topics
- About Displays the version number of this application.

New command (File menu)

Use this command to create a new document in <<YourApp>>. Select the type of new file you want to create in the File New dialog box. << Remove previous sentence if your application supports only one document type. >>

You can open an existing document with the Open command.

Shortcuts

Toolbar: 
Keys: CTRL+N

File New dialog box

<< Delete this help topic if your application supports only one document type. >>

Specify the type of document you wish to create:


<< List your application's document types here >>

Open command (File menu)

Use this command to open an existing document in a new window. You can open multiple documents at once. Use the Window menu to switch among the multiple open documents. See Window 1, 2, ... command.

You can create new documents with the New command.

Shortcuts

Toolbar: 
Keys: CTRL+O

File Open dialog box

The following options allow you to specify which file to open:

File Name

Type or select the filename you want to open. This box lists files with the extension you select in the List Files of Type box.

List Files of Type

Select the type of file you want to open:

<< List your application's file types here. >>

Drives

Select the drive in which <<YourApp>> stores the file that you want to open.

Directories

Select the directory in which <<YourApp>> stores the file that you want to open.

Network...

Choose this button to connect to a network location, assigning it a new drive letter.

Close command (File menu)

Use this command to close all windows containing the active document. <<YourApp>> suggests that you save changes to your document before you close it. If you close a document without saving, you lose all changes made since the last time you saved it. Before closing an untitled document, <<YourApp>> displays the Save As dialog box and suggests that you name and save the document.


You can also close a document by using the Close icon on the document's window, as shown below:



Save command (File menu)

Use this command to save the active document to its current name and directory. When you save a document for the first time, <<YourApp>> displays the Save As dialog box so you can name your document. If you want to change the name and directory of an existing document before you save it, choose the Save As command.

Shortcuts

Toolbar: 
Keys: CTRL+S

Save As command (File menu)

Use this command to save and name the active document. <<YourApp>> displays the Save As dialog box so you can name your document.

To save a document with its existing name and directory, use the Save command.

Send command (File menu)

Use this command to send the active document through electronic mail. This command presents a mail window with the active document attached to it. You may then fill out the To: field, Subject: field, etc., and add text to the body of the message if you wish. When you are finished you may click the "Send" button to send the message.

File Save As dialog box

The following options allow you to specify the name and location of the file you're about to save:

File Name

Type a new filename to save a document with a different name. A filename can contain up to eight characters and an extension of up to three characters. <<YourApp>> adds the extension you specify in the Save File As Type box.

Drives

Select the drive in which you want to store the document.

Directories

Select the directory in which you want to store the document.

Network...

Choose this button to connect to a network location, assigning it a new drive letter.

<< Add other File Save As dialog box options depending on which ones your application chooses via the OFN_ flags of the OPENFILENAME structure used by the CFileDialog. >>

1, 2, 3, 4 command (File menu)

Use the numbers and filenames listed at the bottom of the File menu to open the last four documents you closed. Choose the number that corresponds with the document you want to open.

Exit command (File menu)

Use this command to end your <<YourApp>> session. You can also use the Close command on the application Control menu. <<YourApp>> prompts you to save documents with unsaved changes.

Shortcuts

Mouse: Double-click the application's Control menu button.



Keys: ALT+F4

Undo/Can't Undo command (Edit menu)

<< Your application's user interface for Undo may differ from the one described below. Modify this help text accordingly. >>

Use this command to reverse the last editing action, if possible. The name of the command changes, depending on what the last action was. The Undo command changes to Can't Undo on the menu if you cannot reverse your last action.

Shortcuts

Toolbar: 
Keys: CTRL+Z or
ALT-BACKSPACE

Redo command (Edit menu)


<< Write application-specific help here. >>

Cut command (Edit menu)

Use this command to remove the currently selected data from the document and put it on the clipboard. This command is unavailable if there is no data currently selected.

Cutting data to the clipboard replaces the contents previously stored there.

Shortcuts


Toolbar: 
Keys: CTRL+X

Copy command (Edit menu)

Use this command to copy selected data onto the clipboard. This command is unavailable if there is no data currently selected.

Copying data to the clipboard replaces the contents previously stored there.


Shortcuts

Toolbar: 
Keys: CTRL+C

Paste command (Edit menu)

Use this command to insert a copy of the clipboard contents at the insertion point. This command is unavailable if the clipboard is empty.

Shortcuts

Toolbar: 
Keys: CTRL+V

Toolbar command (View menu)

Use this command to display and hide the Toolbar, which includes buttons for some of the most common commands in <<YourApp>>, such as File Open. A check mark appears next to the menu item when the Toolbar is displayed.

See [Toolbar](#) for help on using the toolbar.

Toolbar



The toolbar is displayed across the top of the application window, below the menu bar. The toolbar provides quick mouse access to many tools used in <<YourApp>>.

To hide or display the Toolbar, choose Toolbar from the View menu (ALT, V, T).

<< Add or remove toolbar buttons from the list below according to which ones your application offers. >>

Click To



Open a new document.



Open an existing document. <<YourApp>> displays the Open dialog box, in which you can locate and open the desired file.



Save the active document or template with its current name. If you have not named the document, <<YourApp>> displays the Save As dialog box.



Print the active document.



Remove selected data from the document and stores it on the clipboard.



Copy the selection to the clipboard.



Insert the contents of the clipboard at the insertion point.



Reverse the last editing. Note: You cannot undo some actions.



Go to the first record in the current selection.



Go to the previous record in the current selection.



Go to the next record in the current selection.



Go to the last record in the current selection.

Status Bar command (View menu)

Use this command to display and hide the Status Bar, which describes the action to be executed by the selected menu item or depressed toolbar button, and keyboard latch state. A check mark appears next to the menu item when the Status Bar is displayed.

See [Status Bar](#) for help on using the status bar.

Status Bar



The status bar is displayed at the bottom of the <<YourApp>> window. To display or hide the status bar, use the Status Bar command in the View menu.

The left area of the status bar describes actions of menu items as you use the arrow keys to navigate through menus. This area similarly shows messages that describe the actions of toolbar buttons as you depress them, before releasing them. If after viewing the description of the toolbar button command you wish not to execute the command, then release the mouse button while the pointer is off the toolbar button.

The right areas of the status bar indicate which of the following keys are latched down:

Indicator	Description
CAP	The Caps Lock key is latched down.
NUM	The Num Lock key is latched down.
SCRL	The Scroll Lock key is latched down.

New command (Window menu)

Use this command to open a new window with the same contents as the active window. You can open multiple document windows to display different parts or views of a document at the same time. If you change the contents in one window, all other windows containing the same document reflect those changes. When you open a new window, it becomes the active window and is displayed on top of all other open windows.

Cascade command (Window menu)

Use this command to arrange multiple opened windows in an overlapped fashion.

Tile command (Window menu)

Use this command to arrange multiple opened windows in a non-overlapped fashion.

Tile Horizontal command (Window menu)

Use this command to vertically arrange multiple opened windows in a non-overlapped fashion.

Tile Vertical command (Window menu)

Use this command to arrange multiple opened windows side by side.

Window Arrange Icons Command

Use this command to arrange the icons for minimized windows at the bottom of the main window. If there is an open document window at the bottom of the main window, then some or all of the icons may not be visible because they will be underneath this document window.

Split Command (Window menu)

Use this command to split the active window into panes. You may then use the mouse or the keyboard arrows to move the splitter bars. When you are finished, press the mouse button or enter to leave the splitter bars in their new location. Pressing escape keeps the splitter bars in their original location. << In a single document interface application, this command will appear on the View menu. >>

1, 2, ... command (Window menu)

<<YourApp>> displays a list of currently open document windows at the bottom of the Window menu. A check mark appears in front of the document name of the active window. Choose a document from this list to make its window active.

Index command (Help menu)

Use this command to display the opening screen of Help. From the opening screen, you can jump to step-by-step instructions for using <<YourApp>> and various types of reference information.

Once you open Help, you can click the Contents button whenever you want to return to the opening screen.

Using Help command (Help menu)

Use this command for instructions about using Help.

About command (Help menu)

Use this command to display the copyright notice and version number of your copy of <<YourApp>>.

Context Help command



Use the Context Help command to obtain help on some portion of <<YourApp>>. When you choose the Toolbar's Context Help button, the mouse pointer will change to an arrow and question mark. Then click somewhere in the <<YourApp>> window, such as another Toolbar button. The Help topic will be shown for the item you clicked.

Shortcut

Keys: SHIFT+F1

Title Bar

<< Show your application's title bar here. >>

The title bar is located along the top of a window. It contains the name of the application and document.

To move the window, drag the title bar. Note: You can also move dialog boxes by dragging their title bars.

A title bar may contain the following elements:

- Application Control-menu button
- Document Control-menu button
- Maximize button
- Minimize button
- Name of the application
- Name of the document
- Restore button

Scroll bars

Displayed at the right and bottom edges of the document window. The scroll boxes inside the scroll bars indicate your vertical and horizontal location in the document. You can use the mouse to scroll to other parts of the document.

<< Describe the actions of the various parts of the scrollbar, according to how they behave in your application. >>

Size command (System menu)

Use this command to display a four-headed arrow so you can size the active window with the arrow keys.



After the pointer changes to the four-headed arrow:

1. Press one of the DIRECTION keys (left, right, up, or down arrow key) to move the pointer to the border you want to move.
2. Press a DIRECTION key to move the border.
3. Press ENTER when the window is the size you want.

Note: This command is unavailable if you maximize the window.

Shortcut

Mouse: Drag the size bars at the corners or edges of the window.

Move command (Control menu)

Use this command to display a four-headed arrow so you can move the active window or dialog box with the arrow keys.



Note: This command is unavailable if you maximize the window.


Shortcut

Keys: CTRL+F7

Minimize command (application Control menu)

Use this command to reduce the <<YourApp>> window to an icon.


Shortcut

Mouse: Click the minimize icon  on the title bar.
Keys: ALT+F9

Maximize command (System menu)

Use this command to enlarge the active window to fill the available space.

Shortcut

Mouse: Click the maximize icon  on the title bar; or double-click the title bar.
Keys: CTRL+F10 enlarges a document window.

Next Window command (document Control menu)

Use this command to switch to the next open document window. <<YourApp>> determines which window is next according to the order in which you opened the windows.

Shortcut

Keys: CTRL+F6

Previous Window command (document Control menu)

Use this command to switch to the previous open document window. <<YourApp>> determines which window is previous according to the order in which you opened the windows.

Shortcut

Keys: SHIFT+CTRL+F6

Close command (Control menus)

Use this command to close the active window or dialog box.

Double-clicking a Control-menu box is the same as choosing the Close command.



Note: If you have multiple windows open for a single document, the Close command on the document Control menu closes only one window at a time. You can close all windows at once with the Close command on the File menu.

Shortcuts

Keys: CTRL+F4 closes a document window
 ALT+F4 closes the <<YourType>> window or dialog box

Restore command (Control menu)

Use this command to return the active window to its size and position before you chose the Maximize or Minimize command.

Switch to command (application Control menu)

Use this command to display a list of all open applications. Use this "Task List" to switch to or close an application on the list.

Shortcut

Keys: CTRL+ESC

Dialog Box Options

When you choose the Switch To command, you will be presented with a dialog box with the following options:

Task List

Select the application you want to switch to or close.

Switch To

Makes the selected application active.

End Task

Closes the selected application.

Cancel

Closes the Task List box.

Cascade

Arranges open applications so they overlap and you can see each title bar. This option does not affect applications reduced to icons.

Tile

Arranges open applications into windows that do not overlap. This option does not affect applications reduced to icons.

Arrange Icons

Arranges the icons of all minimized applications across the bottom of the screen.

Ruler command (View menu)

<< Write application-specific help here. >>

Choose Font dialog box

<< Write application-specific help here. >>

Choose Color dialog box

<< Write application-specific help here. >>

Find command (Edit menu)

<< Write application-specific help here. >>

Find dialog box

<< Write application-specific help here. >>

Replace command (Edit menu)

<< Write application-specific help here. >>

Replace dialog box

<< Write application-specific help here. >>

Repeat command (Edit menu)

Use this command to repeat the last editing command carried out. The Repeat menu item changes to Can't Repeat if you cannot repeat your last action.

Shortcut

Key: F4

Clear command (Edit menu)

<< Write application-specific help here. >>

Clear All command (Edit menu)

<< Write application-specific help here. >>

Next Pane

<< Write application-specific help here. >>

Prev Pane

<< Write application-specific help here. >>

Modifying the Document

<< Write application-specific help here that provides an overview of how the user should modify a document using your application.

If your application supports multiple document types and you want to have a distinct help topic for each, then use the help context i.d. generated by running the MAKEHELP.BAT file produced by AppWizard. Alternatively, run MAKEHM as follows:

```
makehm IDR_HIDR_,0x2000 resource.h
```

If the IDR_ symbol for one of your document types is, for example, IDR_CHARTTYPE, then the help context i.d. generated by MAKEHM will be HIDR_CHARTTYPE.

Note, AppWizard defines the HIDR_DOC1TYPE help context i.d. used by this help topic for the first document type supported by your application. AppWizard produces an alias in the .HPJ file for your application, mapping HIDR_DOC1TYPE to the HIDR_ produced by MAKEHM for that document type. >>

No Help Available

No help is available for this area of the window.

No Help Available


No help is available for this message box.

<< If you wish to author help specific to each message box prompt, then remove the AFX_HIDP_xxx values from the [ALIAS] section of your .HPJ file, and author a topic for each AFX_HIDP_xxx value. For example, AFX_HIDP_INVALID_FILENAME is the help topic for the Invalid Filename message box. >>

Print command (File menu)

Use this command to print a document. This command presents a Print dialog box, where you may specify the range of pages to be printed, the number of copies, the destination printer, and other printer setup options.

Shortcuts

Toolbar: 
Keys: CTRL+P

Print dialog box

The following options allow you to specify how the document should be printed:

Printer

This is the active printer and printer connection. Choose the Setup option to change the printer and printer connection.

Setup

Displays a Print Setup dialog box, so you can select a printer and printer connection.

Print Range

Specify the pages you want to print:

All Prints the entire document.

Selectio Prints the currently selected text.

n

Pages Prints the range of pages you specify in the From and To boxes.

Copies

Specify the number of copies you want to print for the above page range.

Collate Copies

Prints copies in page number order, instead of separated multiple copies of each page.

Print Quality

Select the quality of the printing. Generally, lower quality printing takes less time to produce.

Print Progress Dialog

The Printing dialog box is shown during the time that <<YourApp>> is sending output to the printer. The page number indicates the progress of the printing.

To abort printing, choose Cancel.

Print Preview command (File menu)

Use this command to display the active document as it would appear when printed. When you choose this command, the main window will be replaced with a print preview window in which one or two pages will be displayed in their printed format. The print preview toolbar offers you options to view either one or two pages at a time; move back and forth through the document; zoom in and out of pages; and initiate a print job.

Print Preview toolbar

The print preview toolbar offers you the following options:

Print

Bring up the print dialog box, to start a print job.

Next Page

Preview the next printed page.

Prev Page

Preview the previous printed page.

One Page / Two Page

Preview one or two printed pages at a time.

Zoom In

Take a closer look at the printed page.

Zoom Out

Take a larger look at the printed page.

Close

Return from print preview to the editing window.

Print Setup command (File menu)

Use this command to select a printer and a printer connection. This command presents a Print Setup dialog box, where you specify the printer and its connection.

Print Setup dialog box

The following options allow you to select the destination printer and its connection.

Printer

Select the printer you want to use. Choose the Default Printer; or choose the Specific Printer option and select one of the current installed printers shown in the box. You install printers and configure ports using the Windows Control Panel.

Orientation

Choose Portrait or Landscape.

Paper Size

Select the size of paper that the document is to be printed on.

Paper Source

Some printers offer multiple trays for different paper sources. Specify the tray here.

Options

Displays a dialog box where you can make additional choices about printing, specific to the type of printer you have selected.

Network...

Choose this button to connect to a network location, assigning it a new drive letter.

Page Setup command (File menu)

<< Write application-specific help here. >>

