

## *Introduction to HTML*

HyperText Markup Language (HTML) is a language for, at its simplest, formatting text. The main reason for the excitement generated in HTML and the World Wide Web is the ability of forming anchors within a document. These anchors are to other documents or graphics images and can be specified with any of the wide range of internet communication protocols such as gopher, ftp or even accessing further HTML documents through http. This leads to a single document becoming interwoven with others and hence the spinning of the World Wide Web.

HTML Notepad can produce HyperText Markup Language which complies with that defined in the HTML Working Group Internet Draft of the HTML 3.0 and 2.0 specification (which their terms and conditions enforce us to say is just a working draft) with a few limited caveats. The definitions for tables and background images as defined in the HTML 3.0 working draft are also produced. Also included are those definitions found in the Netscape V2.0 browser and Microsoft Internet Explorer V3.0.

The HTML specification documents are at least 400 pages long and to design a professional looking page requires many months, maybe even years of practice. In this help file we will instead summarise how to use most of the formatting styles and anchors within HTML 2.0 and 3.0 and the further additional commands available in HTML Notepad.

## *General Structure of HTML Document*

The complete HTML document should be enclosed within <HTML> and </HTML> tags. A document is then made up of the <HEAD> and <BODY> section. General comments and titles should be placed in the head section and the rest of the document should be placed in the body section. The <BODY> tag can also be extended (for compliant browsers) to include information of a background colour and text colours.

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>A nice title for your page</TITLE>
```

```
</HEAD>
```

```
<BODY BACKGROUND="wibble.jpg" BGCOLOR="#000000" TEXT="#FFFFFF">
```

```
<!-- Your html text should go here-->
```

```
</BODY>
```

```
</HTML>
```

### **Title**

This allows the browser to have a title of the page and is also used as the default text for bookmark or hot list definitions in common web browsers.

### **Comments**

Comments are enclosed in the tags <!-- and --> anything even HTML code will not be parsed within a comment field. This type of comment may become superceded by the <COMMENT> style tag.

## *Block Structuring Elements*

### **Paragraph <P> ... </P>**

Determines a paragraph. This is formatted within the browser as having a new line to end it and also spacing from other paragraphs.

### **Headings <H1> ... </H1> through to <H6> ... </H6>**

These headings are all defined in different ways with H1 being of largest size and importance.

### **Alignment of Headings and Paragraphs**

Most browsers now allow the user to specify the alignment of the paragraph and headings. The alignments currently allowed are to the left, center or right of the page.

```
<P ALIGN=CENTER>Wibble fish markov</P>
```

```
<H1 ALIGN=RIGHT>A Heading forsooth</H1>
```

## *Hypertext Elements*

### **Anchor <A> ... </A>**

Anchors are the basis of the hypertext abilities of HTML the writer anchors a particular piece of text or image to another piece of documentation or segment. Below are the different types of anchor.

### **Types Of Anchor**

#### ***HREF***

This allows the definition between the anchor tags to be hypertext. This can be of any URL type such as another World Wide Page such as <http://www.salford.ac.uk/docs/depts/eee/apf.html>.

If the form HREF="#name" is used the link is within the actual document currently being read.

```
<A HREF="http://www.u-net.com/virtua/weeble">Weebles</A>: they wobble but not fall down.
```

#### ***NAME***

This allows the anchor to be referenced in the document. For example, if in glossary.html we have;

```
<A NAME="weeble">Weebles</A>: they wobble but they do not fall down.
```

then the `<A HREF="glossary.html#weeble">` will jump directly to that particular anchor within the document.

## *Data Elements*

All type of data elements are incorporated into HTML Notepad.

### **Line Break    <BR>    Ctrl-Enter**

Shows that a new line must be started at that position regardless of other styling.

### **Horizontal Rule <HR>    Alt-Enter**

Divider of Text. Netscape extensions allow both the SIZE and WIDTH (in pixels or percentages) to be specified. These extensions are available through the Netscape pop-up menu.

```
<hr size=5 width=70 align="center">
```

### **Image   <IMG>   Alt-I**

This will always ask for a source for the image as standard further attributes are;

ALIGN for the alignment of the next section of text, the values are TOP, MIDDLE or BOTTOM.

ALT allows text as an alternative for the graphic (useful for non-graphical browsers such as Lynx).

ISMAP defines the image as a map in which different areas can be clicked though this must be programmed using a number of varieties of programs.

## *Character Format*

All character formats are tags which enclose the particular characters wanting to be styled. The closing tag is of the form `</tagtype>`. HTML Notepad allows the user to select text and then choose a particular tag type and performs the enclosing operator for the user. Some tags described at the bottom of the table are proposed for HTML 3.0 and **may** be implemented in particular in **some** browsers. This is not the place for a description of the battles going on at proposal committees but please use the tags with care.

Format Type	Tag	Appearance	Short Cut Key
Physical Format	B	Bold	Ctrl-B
	I	Italic	Ctrl-I
	TT	Teletype fixed width	
Semantic Format	EM	Italics	
	CITE	Italics	
	CODE	Small font, fixed width	
	KBD	Bold, fixed width	
	LISTING	Denotes listing, Fixed width	
	PLAINTEXT	Fixed width with no processing of tags	
	PRE	Fixed width type	
	S	Strikethrough	
	SAMP	Code sample, small font, fixed width	
	STRIKE	Strikethrough	
	STRONG	Bold	
	U	Underlined	
	VAR	Small Font, fixed width	
Other Format	BLOCKQUOTE	Indents both right and left margins	
	COMMENT	Ignored unless contains HTML	
	BLINK	Blinks text, ugh	
	ADDRESS	Italics	
HTML3 Format	ABBREV	Abbreviations	
	ACRONYM	Acronyms	
	AU	Author format	
	BIG	Bigger font, already implemented in Net 2	
	BQ	Replacement for block quote	
	CREDIT	No typical rendering defined	
	DEL	Show deleted or modified text	
	DFN	not widely implemented but rendered bold	
	FN	Footnote should have id to anchor name	
	INS	Show inserted text	
	NOTE	Indented text	
	PERSON	Mainly useful for automatic indexing	
	Q	Short Quotation	
	SMALL	Small font, implemented in Net2	
	SUB	Subscript, no nesting	
	SUP	Superscript, no nesting	

## *Lists*

To automatically construction a list the writer must have the text broken up by carriage returns (especially the final member which is a common mistake). Select

Lists can be nested but they must be created separately and then copied into correct position with HTML Notepad.

All members of the list must be ended by a carriage return

### **List Item <LI> ... </LI>**

Each item in the list is enclosed with this tag.

### **Ordered List <OL> ... </OL>**

The order is signified by a number. Of a variety of depths which is usually shown through tabs.

### **Unordered List <UL> ... </UL>**

Unnumbered list in every other way similar to ordered List.

### **Directory List <DIR> ... </DIR>**

Used to present a list of items containing up to 20 characters each.

### **Menu List <MENU>...</MENU>**

Usually contains one item per line.

### **Definition List <DL> ... </DL>**

A list of definitions quite simply.

### *Special characters*

HTML defines a large number of special characters, far too numerous to mention here, some common and important ones are;

<	&lt;	&#60;
>	&gt;	&#62;
&	&amp;	&#38;
@		&#64;

This is the generic form for special characters with a potential description as SGML code within the & and ; or as a unique 8 bit number derived from ISO-8859-1 8-bit single byte code graphic character set.

### **Foreign Characters**

Most foreign characters can be created through the use of the make acute, grave, circumflex, tilde and umlaut functions. Select the character to be converted and then use the particular conversion. The extended character set also includes a few relatively common characters.

We also include two character which are specific (currently) to Netscape these are the copyright symbol © and the registered symbol ® as &copy; and &reg; respectively.



## *Tables*

Tables are used to define a series of rows of table cell elements. Each element in the table is enclosed by a table data or table head tag with each row being enclosed by a table row tag. HTML Notepad can create a blank table or load in comma or tab delimited data and form a table around it.

### TABLE

This tag must enclose the table. Common attributes are BORDER which draws a border around your table to a given size. CELSPACING and CELLPADDING allow the user to define space around the text in the table while width defines the whole tables size. Alignment tags control the alignment of the text within the table. HTML 3 extensions include FRAMES and RULES which control the border frames and rules between data element.

### TR, TH, TD

TR must enclose every row in your table. TH and TD enclose each item and are usually drawn as a table head with bold type and table data with a regular type.

### TBODY, THEAD, TFOOT (HTML 3)

The TBODY encloses the main body of the table data and defines the styles for the data. THEAD and TFOOT are used to define the head and foot information of the table which is useful for when the table may be placed over a number of pages. These tags are not as yet widely implemented.

## *Forms Based Inputs*

Forms are the interactive form of HTML they use scripts and executables to perform certain actions on input data. There are two methods of sending information to the executables GET and POST. Each form based script is unique as is the form HTML front end. The text below can also be found in the file example.htm.

The example below will not work as there is no script called wibblefish. We apologise but all our scripts are behind firewalls. This does show most, if not all, of the aspects of forms contained in HTML 2.0.

### **Form Setup**

This forms uses the post method there is also get which defines how information will be sent to the action program.

```
<FORM METHOD="POST" ACTION="http://www.salford.ac.uk/wibblefish">
```

### **Simple input**

```
<P>Your Name: <INPUT NAME="name" TYPE=TEXT size="24"></P>
```

### **Password Input**

```
<P>Your Password: <INPUT NAME="pass" TYPE=PASSWORD size="24"></P>
```

### **Select from a group**

```
<P><SELECT>
<OPTION>wibble
<OPTION>plaice
<OPTION>haddock
<OPTION>dalmation
</SELECT></P>
```

### **Checkbox to choose a number of parameters**

```
<OL>
<LI><INPUT NAME="kent" TYPE=CHECKBOX VALUE="kent"> Kent
<LI><INPUT NAME="canterbury" TYPE=CHECKBOX VALUE="canterbury"> Canterbury
<LI><INPUT NAME="manchester" TYPE=CHECKBOX VALUE="manchester"> Manchester
<LI><INPUT NAME="coventry" TYPE=CHECKBOX VALUE="coventry"> Coventry
</DL>
```

### **Radio buttons to choose only one particular parameter**

They must have the same name and **must** have an explicitly defined value.

```
<INPUT NAME="wib" TYPE=RADIO VALUE="1" >wib1
<INPUT NAME="wib" TYPE=RADIO VALUE="2" >wib2
<INPUT NAME="wib" TYPE=RADIO VALUE="3" >wib3
<INPUT NAME="wib" TYPE=RADIO VALUE="4" >wib4
```

### **Text area**

Multiple Line text inputs are treated as text areas, you can specify the number of rows and columns.

```
<TEXTAREA NAME="comment" ROWS=8 COLS=24 >
Comments
</TEXTAREA>
```

### **Submit or reset the whole form**

```
<P><INPUT TYPE=SUBMIT><INPUT TYPE=RESET></P>
</FORM>
```

## *HTML3 Objects*

This has been implemented in Internet Explorer and is a potential standard way of including all the internet and multimedia applications currently in the market. It is described in a working draft which can found with a number of others at <http://www.w3.org/pub/WWW/TR>

The OBJECT tag subsumes the role of the IMG tag and is backwards compatible.

The object specification is very general to allow any application to be specified to give the designer a better feel of object we run through some [examples](#). We hope to see lots of OBJECT tags used in documents they just feel right and will hopefully supersede all other non-standard application callers.

### **OBJECT**

#### **Attributes**

ID	Defines a document-wide identifier
DECLARE	Used to define an object but not instantiated
CLASSID	URL that identifies an implementation of object
CODEBASE	Specifies URL to find above CLASSID
DATA	URL pointing to objects data
TYPE	Type of object (Internet Media Type)
CODETYPE	Type of object reference by CLASSID
STANDBY	Specify a short text string while loading
ALIGN	texttop, middle, textmiddle, left, center, right
WIDTH	Width of object (can be used to expand image)
HEIGHT	Height of object
BORDER	Border when a hypertext link
HSPACE	Horizontal Space Around Box
VSPACE	Vertical Space Around Box
USEMAP	Client-side area map
SHAPES	Hot-spots to anchors
NAME	Used for information to a FORM

### **PARAM**

Parameters have simple name and value attributes. The PARAM tag is enclosed by the OBJECT tag and if the object is rendered the attributes are sent to the program run.

## *HTML3 Objects Examples*

### **Example**

```
<OBJECT data="TheEarth.avi" type="application/avi">

</OBJECT>
```

The type attribute of the object allows the browser to simply identify whether it can render the data if not the image element is then parsed. If the image cannot be loaded or rendered (as in a browser such as lynx) we then move to the simple alternative text. An elegant method of moving through all the multimedia options.

This gives the designer an innate flexibility to place whatever applications they wish into a document knowing that some information will be received by the reader. Hence for a shockwave presentation.

```
<OBJECT data="shocknew.dcr" type="application/director" width=200 height=200>

</OBJECT>
```

### **Java**

The object tag also allows a common interface to Java (and other potential internet languages).

### **Example**

```
<OBJECT CODETYPE="application/java-vm" CODEBASE="http://www.wibble.com"
      CLASSID="java:CranialLoopItem" HEIGHT=100 WIDTH=100>
<PARAM NAME="img" VALUE="globe">
<PARAM NAME="pause" VALUE="1">
Your browser does not support Java, sorry, and thanks for the bytes.
</OBJECT>
```

The similarities to the Netscape applet tag are obvious. As the designer cannot place the full pathname of the java class (program) on the CLASSID attribute this must be placed in the CODEBASE attribute. The type attribute is then used by the browser to check whether Java can be rendered if not all the enclosed PARAM tags are ignored and the error message is rendered.

The PARAM tags are used to send parameters to the Java or any other application. These are simply name and value attributes.

### *HTML3 Objects and Client-side Area Maps*

The OBJECT tag specification include the USEMAP attribute and can be used in a similar way to Netscape Client-side Area Maps.

There are two methods for producing this effect are given below. One uses the SHAPES attribute to specify that the enclosed text will include hypertext links in the form of shapes while the second uses the USEMAP attribute to point to an area map.

#### **Example1**

```
<object data="wibble.gif" WIDTH=50 HEIGHT=50 SHAPES>
<A HREF="wibble1.html" SHAPE=rect COORDS="0,0,25,25">wibble1</a>
<A HREF="wibble2.html" SHAPE=rect COORDS="25,25,50,50">wibble2</A>
</object>
```

#### **Example2 (The Spyglass Method)**

```
<object data="wibble.gif" WIDTH=50 HEIGHT=50 USEMAP="#wibbly"></object>

<MAP NAME="wibbly">
<AREA SHAPE=rect COORDS="0,0,25,25" HREF="wibble1.html" ALT="wibble1">
<AREA SHAPE=rect COORDS="25,25,50,50" HREF="wibble2.html" ALT="wibble2">
<AREA SHAPE=rect COORDS="0,0,50,50" NOHREF>
</MAP>
```

## *Registering HTML Notepad*

We hope you enjoy using HTML Notepad. If you do find it useful please make the effort to register your copy. By registering you will be added to our mailing list of users who will be told of further releases and you will also be supporting all improvements to this code. Registration covers both Windows 3.1 and 95 versions and you can use both on your personal systems.

	<b>UK &amp; Europe (Pounds Sterling)</b>	<b>International (US Dollars)</b>
<b>HTML Notepad (electronic)</b>	24	45
<b>HTML Notepad</b>	25	48
<b>HTML Notepad (5-10)</b>	22	41
<b>HTML Notepad (10+)</b>	20	37

### **Multiple Registrations**

We already have small discounts upon multiple orders. Site licenses and educational discounts (for multiple orders over 10 only we are afraid) are all negotiable.

### **Credit Card Registration**

At the time of development of HTML Notepad V2.2 and the creation of this help file credit card payment was not available through the Internet from a UK bank (May 1996). If, since then, you have news that we can produce this we would love to hear from you (for UK banks only please). If this has already occurred you may find details at <http://www.cranial.com/software/htmlnote> (which also contains updates, patches etc).

### **How To Register**

HTML Notepad can be registered in two ways.

EITHER: As you may have noticed a register dialog box is included in the Help pop-up. Once your money has been received we will take your registration name and create a 16 digit personal code this can be input and will convert your shareware version into the fully registered version of HTML Notepad.

OR: Send us your snail mail details and we will register your copy here and send it to you on a disk.

Please print out [this form](#) to fill in and register. Please allow 28 days for mail delivery (though we have never taken that long).

If you wished to be invoiced or require a receipt please make this clear on your order.

### **Registration Cost Conversions**

If you wish to pay in any other currency please make the conversion in comparison to US dollars. All cheques, postal orders must be made payable to : ***cranial publishing***. Please also remember to include your registration name which should be greater than 4 and no more than 28 characters long. Once your cheques have cleared (or potentially before we are trusting) we will send via e-mail your registration code.

If you are paying by Eurocheque please make it payable in UK pounds sterling.

### **Upgrading From Previous Versions**

As you can probably tell a lot of new work has gone into HTML Notepad V2.1 we therefore cannot possibly just give away this version to our regular users. To upgrade please pay whatever difference there was between your previous registration cost and the costs shown above.

**Publishing HTML Notepad on a CD-ROM or floppy disk**

We are quite happy for you to publish the shareware version of HTML Notepad on a CD-ROM or a floppy disk. While not a contractual obligation we would be obliged if you sent us a copy of whatever you publish for our own records. We are also happy for HTML Notepad to be used as part of a course in HTML or other lectures.

Snail mail to: Cranial Publishing, 1 Sir Winston Churchill Place, Binley Woods, Coventry, CV3 2BT, United Kingdom.

For more details of HTML Notepad or other pieces of cranial software, e-mail : [info@cranial.demon.co.uk](mailto:info@cranial.demon.co.uk) or those people on-line or waiting in-line then try <http://www.cranial.com/software/htmlnote>

*Cranial Publishing, 20 November 1996, Manchester, UK*

### *Registration Form*

Please print out this form by using File|Print Topic on the menu bar or some other combination in Windows95 (we find a right hand click mouse usually works).

Name :

Registration Name (more than 4 but less than 28 characters, if different to above):

EITHER ELECTRONIC REGISTRATION

E-mail:

OR MAIL REGISTRATION

Company :

Address :

Town / City :

County / Area :

Postcode / Zipcode :

Country :

—

Do you require an invoice or receipt (for mail registrations only) :

Number of copies :

Comments (constructive, anything you would like to see):

Mail with cheque to: Cranial Publishing, 1 Sir Winston Churchill Place, Binley Woods, Coventry, CV3 2BT, United Kingdom.



### *Why HTML Notepad does not look like a browser*

One of the major discussions about HTML Notepad we have had with our testers and those users of previous versions is why there is no method to produce a browser like output from the editor or even edit the document in the browser style.

The simplest argument we can put forward is which browser should we standardise upon, Mosaic which is free and (in its current form) looks quite, Netscape currently the most popular and producing new non-standards every single day or even Arena the HTML 3.0 compliant browser. Instead we allow this crucial decision to be yours not ours.

HTML Notepad assists you in producing the document and also gives the user the ability to get interesting and professional effects in different browsers. With the HTML community in a constant state of flux to tie ourselves to one particular browser would be futile.

We view HTML Notepad as an assistant to the construction of HTML pages rather than the only tool you should use. While we personally find the HTML templates for certain word processors too slow and unwieldy to actually use there is obviously some need for initial construction of the document in a WYSIWYG format (this is especially true of those people beginning the way of the HTML though it is surprising how quickly you wish to access the source as you become more experienced). HTML Notepad was intended for editing the documents after they have been constructed though this has been somewhat superseded by all the tools now available to you. At Cranial Publishing we have now completely given up on using template documents internally and construct everything with HTML Notepad.

The best effects, in our humble opinion, simply cannot be created with a WYSIWYG compliant browser. For instance, the creation of forms really needs to be at the source level rather than some graphical user interface as does most of the work with frames and common icon bars.

## *Cranial's Hints & Tips*

This is intended as a short little text on how to produce some of the effects found in professional web pages. It is relatively personal and does not cover all the possible tricks. We would rather users not find pages we have designed which clearly break the below rules as we know we are hypocrites (and young once) and this is clearly a case of 'do as we say and not as we do'. You may vehemently disagree with what we are saying and if so we would like to hear why but we think these are reasonably fair guidelines.

- Use transparent gif images. These can be created by using a small but useful program called giftrans (do an archie scan on either giftrans.exe (DOS) or giftrans.c (UNIX)). The command line interface is slightly off-putting but the results are well worth it. You can also find this as an option in Paint Shop Pro or many other graphics programs.
- If your images can be small, make them so. If your images must be huge make sure your initial pages do not have the images and/or prepare people for the download time or even better offer alternatives to a more textual pages at the top of the page. Some people pay for the pleasure of downloading your pages.
- As well as images, your pages should be as small as possible while it is much simpler to produce your 200 page document in one HTML page waiting for the 97th page to download so the user can read the first will mean losing readers.
- Background colours and images: Netscape 1.1+ and, hopefully, HTML 3.0 browsers, include the ability to alter the background image and colours of text. HTML Notepad gives a fairly easy interface to this in Netscape|Text Color. Please see the 'example.htm' file for an example. Be careful on its use and do not get too garish. Also remember that you should use the BGCOLOR tag set to the colour of your background image. If your image for some reason does not download your text colours will still look fine.
- On the theme of colouration in your document one HTML trick is to declare multiple <BODY BGCOLOR=??????> with differing colour values this means the browser will have to move through them before rendering the text. It is a relatively cheap way (in network transmission terms) of getting the attention of the user. This is also true of the <BODY BACKGROUND=??> statement but this requires downloading multiple images. This of course is an extremely naughty thing to do as it is no longer true HTML but it does work.
- Use frames or icon bars a lot, because we like them though again it is wonderfully non-standard HTML.
- Although your graphics may be wonderful and the page is well set out in the end the most important thing is the content. The number of pages with bad grammar and abysmal netiquette is ever increasing. Try not to be part of it all.
- Check you pages in more than one browser. Pages that look fine in Netscape look awful in Mosaic. Remember this is your public face. Check and the recheck and then use arena to really mess things up on screen.
- Never have a link to a page which has just under construction on it. Either do not have a link or make sure you have some content for people to read.
- If you have an image map in your document do make sure that underneath there is at least some links to the same points. There is nothing worse than downloading 97k of a 100k image when the packets stop flowing and you can still not go anywhere.
- Never use blink. Simple straight forward to the point. Just don't, *ever*.

## *Code History*

Most of the ideas in the latest iterations of htmlnote are by you the users the latest patches acknowledge those people who find bugs and also some additions.

### **V2.2 26 Nov 96**

New Java menu. Moves Java from Netscape menu to its own. Altered the Applet to automatically include class code.

Whole new javascript section from below.

In Java menu the Goto Line.. function now you can debug JavaScript in HTML Notepad. As both Netscape and Iexplorer give you a line number only to hack through.

JavaScript event handlers (not complete)

New netscape 3.0 SRC for JavaScript.

Param name and value easier dialog.

Hope you all like it.

### **V2.1 (patch3): 20 Nov 96 (never released)**

Added JavaScript output from Save As option. Now you can have your page completely created by JavaScript on Load.

Added some special characters in the conversion to C, Perl and JavaScript “ are now converted to \” and # becomes \# this allows compilation straight from conversion as required.

Small section on how to do JavaScript.

### **V2.1 (patch2)**

Fixed bug in Win95 version which meant you always had toolbar when initialised (Thanks: Edward Castle).

Ctrl+Q for BlockQuote ( for Edward Castle)

TAB now gives a tab. Apologies but we never noticed before (Thanks: Jeffrey L. Fitzgerald et al)

### **V2.1 (patch1)**

When browsing in Windows 3.1 you will always get lowercase. Win95 is still case-sensitive.

New JavaScript template to allow easy document modification time.

Long URLs and images now wrap around in dialog box.

### **V2.1**

New Window option include in File|New to allow you to open multiple HTMLnotes.

To facilitate this we have altered the file name to come first in title bar. Iconise this and the file will come up rather than HTML Notepad all the time.

Fixed AHREF and ANAME, IMG to go lower case when you specify.

Made HR Special Win95 compatible dialog.

Moved Test Page, Browser Setup, Table of Contents, Name Heading etc. into Tools pop-up menu.

Drag & Drop files into HTML Note. The default is just to insert dropped files but if you press CTRL you will start a new file

Evaluation does naught but tells you how long you have been using it on splash screen.

New Yen and Pound sign on character menu.

Insert customised colour menu in Tool pop-up (Why would anyone want this?)

### **V2 (patch1)**

Lists altered to include last item even if not delimited by carriage return.

Evaluation no longer turns off HTMLnote just nags you more.

When you hit 32k now tells you to use splithtm.

Tables now can form headings from first row or column when converting from tab or comma delimited text.

Seems to be some problem with NT and WoW layer which we are having problems identifying. Hopefully one of the by-products of this will be that it is fixed. Windows95 version is definitely completely 32-bit. Cannot see problem...

Added further examples and also a list of references to more material.  
Speed-ups some problems with Perl & C conversion fixed.

## **V 2.0**

HTML 3.0 compliance.  
Help file is exactly how we intended to be from the beginning. Good hope it helps somebody...  
Toolbar designed.  
Windows 95 and Windows 3.1 compliant.  
Included Netscape 2.0 and Microsoft Internet Explorer 2.0 (and 3.0) new tags.  
Imports tables and forms blank table.  
Massive improvements (in memory and speed) in most common functions.  
Log file produced which allows you to log action upon files.  
Convert to C and Perl.  
HTML 3.0 Objects included.

## **v1.19**

Table Of Contents.  
Fully working special characters section.  
Remember chosen font of the user.  
Shareware version can be converted to registered with a registered name and code.  
other enhancements as below

## **v1.18**

Horizontal rule dialog in Netscapism  
Most functions are now Win 95 ready for recompilation.  
Coherent menu system of 1.16 so new version.

## **v1.17**

registered fully operating version of below.

## **v1.16b (too many versions recently we may have to refine this process)**

Added Ctrl+b Ctrl+i for bold and italics.  
Moved menus around so they can all fit into max view of 640x480 section.  
    Test moves into file menu  
    Tables moves from netscapisms to its own section  
    Search, find and replace moves to edit  
    So does options section  
    Reload on Save moves to testing section.

## **v1.16a**

Really fixed UNIX problem this time. Sorry for any inconvenience our source information was incorrect.  
Added initial.htm so when you do File|New you get a newly created document in a particular style.

## **v1.16**

Included automatic conversion from UNIX into DOS format when you load a file...  
Unfortunately on save if you want UNIX format you must save this as a special type given as a file menu option.  
Fixed DEL key.  
Something very odd happened when initialising the background image selector fixed that.  
Added oddles of accelerator buttons for everyone.  
Fixed bug in that if you were minimized and then closed the program and then saved the file it got set to zero. This was bad and no longer occurs, my humblest apologies if this has happened to you.

## **v1.15**

Simply dispensed with the letters.

Fixed slight bug in Name creation which allowed you to create overlong Name strings.  
Potential bug with some combinations of saving setting files to zero. Could not repeat error so instead create a backup every time you save file.  
Slight problem with using edit control is that either minimise or maximise process takes a long (5 seconds!) time. We have moved this to the maximise process and have loaded the wait icon to tell you we are doing something.  
Improved help file to include Fonts curvey and new size.  
Also included warranty information just in case !!

#### **v1.14a**

Added the ability to create forms under new popup Forms|etc...  
Improved HREF, Name and Image dialogs to include previous values.  
Added font curve and font size components (in Netscapisms|Font...)  
Remove tags from text (which can be treated like an undo feature)  
Added Netscape <CENTER> flag, though this appears not to be well liked.  
Added (Netscape?) special character &copy; and &reg;

#### **v1.13**

Added ability to remove HTML tags by saving the document or by selection of area.  
Insertion of text from file.

#### **v1.12**

Added ability to create tables and background images.  
New section to create coloured text for weird, dark backgrounds.

#### **v1.11**

Allow change in Word wrapping (inelegant but it works).

#### **v1.1**

Initialisation file is now stored with the executable rather than in \Windows directory.  
Reload images in Netscape ( code provided by Lee Griffiths ) may be dangerous !  
Fixed all the bugs in v1.0 obviously.

#### **v1.0:**

Added last files.  
Ability to insert owner details into text.  
Insert specific characters and any ASCII code.  
Test by executing Web browser section.  
Added browse button to most dialogs.

#### **v0.1 -> 0.9**

Not released and in chunks, too many changes too mention.

#### **To do (or the dreaming section)**

New syntax coloured editor without 32k limit ( it is going to happen one day we promise... probably).

### *Warranty*

Cranial Publishing reserve the right to make changes to the product described in this help file and any accompanying text at any time and without due notice.

The enclosed software, media and documentation are provided 'as is'. There is no warranty of fitness for a particular use, and no other warranty of any kind either express or implied regarding this software, media and documentation. Accordingly, the entire risk as to the use, results and performance of the software, media and documentation is assumed by you. In no event will Cranial Publishing be liable to you or anyone else for any consequential, special, or indirect damages resulting from use, possession or malfunction of this product, including without limitation, damage to property and, to the extent permitted by law, damages for personal injury.

## *Sounds*

### **BGSOUND**

Specifies a background sound. The sound types allowed are wave, audio and midi files (\*.wav,\*.au,\*.mid). The number of loops can be specified with a finite number or an infinite special parameter.

### **Example**

```
<BGSOUND SRC="wibble.mid" LOOP=infinite>
```

## *Watermarks (Static Background Images)*

### **BGPROPERTIES**

This is another parameter extension to the BODY of the HTML document and influences the rendering of a background image. Currently the only setting is 'fixed' and this forms a watermark.



## *Font Extensions*

### **Font Color**

You can now specify a color for a particular selection of text by enclosing it within the font tag and then using an RGB value for the colour. Within HTML Notepad this is simplified by using the color common dialog.

### **Font Face**

This allows the designer to use different font faces for their text. Each face is specified as the font name used in the MS Windows operating system. This is likely to be superseded by the HTML3 style sheets and is also very MS Windows specific. We would advise only using this option on controlled access areas where only known browsers can be used (such as an Intranet).

### **Example**

```
<FONT COLOR="#FF0000" FACE="Times New Roman">wibble wobble</FONT>
```

## *Video*

### **DynSrc**

An extension to the image tag where a dynamic source (DYN\_SRC) can be defined which points to an animation (\*.avi) file. If applicable the browser will download and render this otherwise it will continue with the simple static image. The designer can also specify CONTROLS which forms video control buttons around the video image.

NOTE: Video images are never small and the bandwidth available is not currently enough for external sites. If you use DYN\_SRC check the pages using a 28.8k modem.

### **Example**

```
<IMG SRC="wibble.jpg" WIDTH=100 HEIGHT=100 DYN_SRC="wibble.avi" CONTROLS>
```

## *Marquees*

The Marquee tag defines an element in which visual scrolling will be used to display the entire content of the element. It can therefore be considered as a subwindow of the document and therefore similar to Netscape frames.

### MARQUEE

This encloses a sub HTML document to be scrolled. There are a number of marquee parameters.

The designer can specify the SCROLLAMOUNT in pixels and the SCROLLDELAY which specifies in milliseconds the delay between each scrolling redraw. The BEHAVIOR of the marquee is either scroll, slide or alternate, the direction of text flow is either left or right. The HEIGHT of the marquee is in pixels with the BGCOLOR of the marquee being a standard RGB value. The LOOPING of text can be either a specific amount or infinite (similar to the video tag)

### *Table Extensions*

Internet Explorer V3.0 renders all the HTML 3.0 tags for tables (including frames, rules, THEAD, TBODY and TFOOT) it also includes three color tags for altering the 3D border of the table.

**BORDERCOLOR, BORDERCOLORDARK, BORDERCOLORLIGHT**

Each of these is an RGB value which specifies a particular border shading for the 3D grid of a table. Experiment to produce some dramatically unattractive effects.

### *Font Size*

The Netscape browser currently allows the designer the ability to increase or decrease the size of their text within a document. This process is made simple in HTML Notepad through the use of the Netscape|Font option. The font size value can be set to any value between -5 and 5.

This font size is the addition or subtraction of the basefont size which can be set to any value (though the defaults for the particular browser as defined by the reader are probably worth relying on) by enclosing any text with the basefont tag.

## *Background Images and Colours*

All color and background image tags are specified within the BODY tag of the document.

### **BACKGROUND**

The background tag sets an image to be the background of the particular web page. This is implemented in most current browsers and is likely to become a standard format (if it has not already). The designer can use gif or jpeg images.

### **BGCOLOR**

This tag in HTML Notepad will call up the standard color dialog box for the designer to choose a color for their background. This is then converted to a RGB value which becomes the tag.

```
BGCOLOR = "#000000"
```

will give a black background color (which we believe causes problems in printing on some browsers by the way). The BGCOLOR tag should be used in all pages where the BACKGROUND image is set to allow the text and link colors defined below to still be shown in the document even if the background image cannot be loaded.

### **TEXT, ALINK, VLINK**

Defines the text, link and used link colors respectively. These can be used in conjunction with the tags above to completely alter the viewing of your document.

### **Example**

```
<BODY BACKGROUND ="wibble1.jpg" BGCOLOR="#000000" TEXT="#ffffff"  
LINK="#ffff00" VLINK="#ff0000">
```

## *Frames*

Frames are a Netscape V2.0+ innovation which allows the designer to produce frames (or sub windows) in a particular document. This can contain a standard table of contents or simply two or more separate areas of discourse.

### **FRAMESET**

The HTML document which defines the layout of further sub-HTML documents uses the FRAMESET tag instead of the BODY tag. The tag can include ROWS and COLS tag which are comma delimited lists of relative frame sizes. These may suffixed with a percentage sign or an asterisk to specify a multiplier of a standard width/height.

### **FRAME**

This defines the actual HTML document which will be drawn within that frame. The frame can have a name defined to allow it to be referenced in the documented. The designed can also define MARGINWIDTH, MARGINHEIGHT and whether the frame will have a scrollbar (SCROLLING).

### **Example**

```
<FRAMESET COLS="25%,75%">
<FRAME SRC="wibble1.html" MARGINWIDTH=3 MARGINHEIGHT=1 SCROLLING=no>
<FRAME SRC="wibble2.html" SCROLLING=auto>
</FRAMESET>
```

### *Client-side Area Maps*

The client side area maps are a possible method to circumvent the CGI script necessary to have an image with hotspots on a world wide page. The new method allows tags to form the hotspots and the client browsers does all the calculations. An image has a new tag called USEMAP which points to a named map in the document

#### **MAP**

The named reference, used by the USEMAP tag within the image, which encloses all the areas which will be hotspots on the image.

#### **AREA**

The hotspot areas on an image. Currently all hotspots are rectangular and must have their SHAPE tag set to rect and the COORDS set to a comma delimited list of four numbers (x1,y1,x2,y2). Potential expansion includes circular hot spots which will require three numbers (x,y and radius) and a polygon (no current browser can produce these effects). Each area should have a link to the next document via a HREF. Each area is checked for a click in the order is specified in the map.

As can be seen in the example below the last area should be the complete image with a link to NOHREF. This will allow any browser to use the map as every click has been specified.

#### **Example**

```
<IMG SRC="wibble.jpg" WIDTH=50 HEIGHT=50 ALT="wibbleimage" USEMAP="#wibbly">

<MAP NAME="wibbly">
<AREA SHAPE=rect COORDS="0,0,25,25" HREF="wibble1.html" ALT="wibble1">
<AREA SHAPE=rect COORDS="25,25,50,50" HREF="wibble2.html" ALT="wibble2">
<AREA SHAPE=rect COORDS="0,0,50,50" NOHREF>
</MAP>
```



## *Java Applet*

Java is a language which runs on the clients machine after being downloaded. This allows it to be relatively fast, non-dependent on the server access time and also a low security risk (as it is kept in virtual quarantine in the browser). This is not an explanation of the Java (please see the HTML Notepad web site for information on Java) but a reference on how to place it into your HTML. The APPLET tag described below may become superseded by the HTML 3 Object tag in the near future as this encompasses a number of other issues.

### **APPLET**

This takes the place of the APP tag which used to define a Java applet. The CODE tag is the name of the file that contains the compiled Java code. This cannot define an absolute URL but must be relative to the base URL of the document. This can be altered by using the CODEBASE tag to define a new absolute base URL for the individual applet.

The WIDTH and HEIGHT tags specify a size for the APPLET to run in while the ALT tag as usual should be used for a browser which cannot perform Java. Alignment tags are used as with an image.

### **PARAM**

The APPLET tag encloses PARAM tags which can be used to send parameters to the Java code. These are defined as simple NAME and VALUE tags. If the browser cannot perform Java applications the APPLET and PARAM tags should be ignored and any other enclosed standard HTML should be browsed.

### **Examples**

```
<APPLET CODE="CranialLoopItem" HEIGHT=20 WIDTH=20 ALT="This was to have been a pretty
animation">
<PARAM NAME="img" VALUE="images/cranial">
<PARAM NAME="pause" VALUE="0">
</APPLET>
```

## *Embedded Applications*

### **EMBED**

This allows the designer to embed documents for specific applications which may or may not be available on the client software. These elements can be utilised by double-clicking on them. They have been used in Netscape examples for VRML, Acrobat and Apple Quicktime media.

The embedded application can have a HEIGHT and WIDTH defined. Each application can have a number of attributes added by using the attribute\_1, attribute\_2 tags which are simple text string parameters

### **NOEMBED**

The noembed tag contains elements which are utilised if the browser cannot produce embedded applications

**NOTE** this functionality is likely to be superseded by HTML 3 Objects

## Testing Pages

### Set Browser

Before testing a page you must point HTML Notepad at your browsers (Tools|Browser Setup) and set at least one as your default. This is done by browsing until an executable is found. Once this is completed the function test page (Tools|Test Page or the glasses icon on the toolbar) will send the particular filename you are working on to the browser hence the file must be saved before testing can occur.

If you wish to use the reload upon save option then you must set a Reload Value in this position. The value of common browsers has already been set. This is the value which is sent to the program to, in effect, select the browser menu option reload. If you have a browser which is unspecified and also have a windows compiler you can load the browser executable into the resource compiler and find the reload value by searching the internally specified menu. Please get in contact if you find new browsers which are unspecified or if you find a reload value we have not included.

This technique is useful for high specification machines where opening and closing your WWW browser is reasonable fast.

As the test page function specified below just sends a command with your filename (in HTML local file format) as the command line option this can be used in conjunction with a HTML checker (such as WebLint) or with multiple browsers.

### Test Page

This runs the browser with the filename (in HTML local file format) as the command line option. If a browser is already running this command may have no effect.

### **Fonts (Large, Small and Curved)**

The Netscape browser currently allows the designer the ability to increase or decrease the size of their text within a document. This process is made simple in HTML Notepad through the use of the Netscape|Font Size and Font Curve options. For font size the size parameter of the dialog box is the only one used and this can be set to any value between -5 and 5.

Font curve takes the selected text and iteratively applies the font size tag from an initial value in increment steps up and down to a maximum (and minimum) value. This can, with certain pieces of text prove, quite appealing (and for others appalling) though it does generate an excessive amount of tags in the code for large amounts of text.

## **Multiple Initialisation Files**

Version 2.0 of HTML Notepad now includes the ability to form multiple initialisation files.. When the user chooses the File|New option the program searches the template directory located underneath the executable file for HTML files (the default initial.htm(l) of previous versions is now located here). The user simply selects the file which is closest for the new task. As these files are HTML files simply located in the template directory the user can add as many files as they wish. They can also change any of the pre-specified templates.

## **File Conversions**

HTML Notepad includes a number of conversion tools to either the UNIX operating system or to produce small sections of code for use in CGI scripts. All conversions can be accessed by using the File|Save As option.

### **UNIX**

As most HTML files are going to be placed upon UNIX servers there is sensibly a working filter for UNIX files. The conversion upon loading a UNIX file is done automatically and is completely invisible. The output of a UNIX file must, on the other hand, be selected. These filters simply convert the carriage return character into the correct format.

### **Text**

The Save without HTML option upon the File pop-up menu automatically creates a DOS text file without any of the standard HTML tags. All special characters will remain.

### **C & Perl CGI Scripts**

These are simple converters which take each line of your HTML and converts it to a printing statement in the language. This is constructed as a subroutine which can be called from the main program. This conversion is not intended to be used with large pieces of HTML text as compiling this into the executable is a waste of resources. In this case it is much simpler to open the HTML file directly in the executable and print this to stdout. Our intention here is to be able to include error or copyright information into the executable which can then not be altered.

### **JavaScript**

With the advent of Netscape 3.0 the HTML web page designer can include JavaScript source in a separate file with the tag

```
<SCRIPT Language="JavaScript" SRC="mycode.js"></SCRIPT>
```

The conversion tool automatically makes this conversion for you to form a JavaScript function which will rewrite your page for you.

## **Building a Table Of Contents**

### **Table Of Contents**

In creating a long WWW page speedy accessibility of information is paramount. HTML allows internal jumps through a document by using a hash referenced anchor to a particular name. Within HTML Notepad these internal links can be produced in the form of a contents page. By positioning your cursor at any point in the document and then using Tools|Table Of Contents option a search will begin through the entire document looking for Name anchors and their respective links. The form of output is a simple text list with each item being separated by a carriage return. The format of the Contents page is left to the discretion of the user.

## **Automatic Name Anchors For Headings**

### **Name Headings**

To make the process of creating internal links easier to the user we have included this function which uses the text of the headings in a document to create anchored names. This function can specifically name headings or do them all. Once created these names can then be used in conjunction with the Table of Contents function to produce a completely automated table of headings in the document.

Please note that this function is not intelligent. It uses the first twelve characters of the headings text to construct the name if these are the same it will form an identical name anchor.



## **Inserting Owner Details**

### **Inserting Owner Details**

At the bottom of every WWW page you will probably see a small amount of text about the authorship and time of creation of the page. HTML Notepad automates this process. Here at cranial we create and maintain hundreds of pages a month all of which need careful correlation with each other the date stamp in the owner details proves invaluable.

All the personal details can be changed through the ownership details dialog box which will include a link to the HTML Notepad home page and the date of page creation.

## **Removing HTML from a selection**

### **Removing HTML**

Made a mistake, created a list when you wanted a table well look no further than the Remove HTML function at the bottom of the HTML menu. This is our form of undo function and will remove all the HTML codes from a selected area. We find this especially useful when redoing other peoples work.

This function is not intelligent and will delete anything between the < > tags even comments. Use with caution.

## **Log Files**

### **Log Files**

This allows the professional user to keep track of all the work being done upon documents (especially useful for billing purposes). It is a relatively simple log of opening and saving of documents with the full path name, opened/saved tag and the date and time. It is intended to be used with a database or filter to calculate the time worked on any one document.

Each time a document is loaded or saved this is appended to the start of the file. If you do a lot of work with HTML Notepad please remember to create a weekly log file.

For an interesting use of this function please see [How To Create a Web Page of Your Work.](#)

## HTML Editor Options

All these options can be found in the Options dialog box, use Edit|Options to access.

### Upper, lower and undefined case tags

This is one of the most requested attributes in the old version of HTML Notepad this allows you to specify whether you wish the tags to be upper or lowercase in your document.

### Toolbar Options

The toolbar can be shown or hidden and in Windows 3.1 can also be positioned at the top or left hand side of the screen.

### Delete Pop-up Menus

The HTML3, Netscape and Internet Explorer menus can all be deleted because you do not wish to design for those systems or you are using a small (640x480) screen and the menu options will not fit.

### Create a backup

Creates a \*.bak file each time you save a document.

### Reload upon save

This reloads your specified default browser every time a file is saved allowing the user to test attributes without having to close and load the browser each time. The function requires that a Reload Value specified, and explained, in the [Set Browser dialog](#) has been set. This function reloads whatever is in the browsers at any one time and is intended to be used in combination with [Test Page](#).

### Warning upon not selecting text

Probably one of the most important options for the professional designer. This switches off the annoying error messages when you do not select text to turn into a tag. This allows you to form tags without any text enclosed.

### Do not use/ Use Log File

See [Log Files](#)

### Multiple Memory Blocks

Each dialog in HTML Notepad has a memory. This, by default, saves 8 of whichever option you are accessing (hence 8 image or 8 HREFs are remembered). This parameter allows you to alter the default. While it may seem to make sense to have as large a value as possible this will slow down dialog box initialisation.

## **Command Line Tools : html2txt and splithtm**

Included in this release are two command line tools which we have been asked for by certain users and we think will be useful to a large number of people

### **HTML2TXT**

A nice simple program to convert any HTML files to simple text files. One can convert a single file or a list of files.

You must use a parameter either

- i for a single file
- l for a list of files seperated by a carriage return.

TIP: Use 'dir /b \*.htm > html.lst' to create a directory listing of all html files then run 'html2txt -l html.lst' to convert them all to text.

### **SPLITHTM**

Splithtm splits up a large HTML or text file into a number of smaller components which can be used in HTML Notepad. The file must have less than six letter as the program appends a number to each new file created.

Usage: splithtm <InputFile> [size of split]

<InputFile> is the single html file

[size of split] defines the Kb to split file 8Kb as default

## **Table Insertion (Creation and Conversion)**

One of the nicest new attributes of HTML Notepad is the ability to insert blank or formatted tables.

### **Inserting a Blank Table**

Select Tools|Insert Blank Table. A dialog box will appear, the number of rows and columns for the table must be specified. Options to form a heading from the first row or column of data are also given. On pressing OK the table will be created all other attributes can be added at this time.

### **Inserting a comma-delimited or tab-delimited table**

This allows HTML Notepad to create tables from data edited within most spreadsheet applications. Comma-delimited text is the simplest format for tabulated text and most applications can save your data in this format. Once saved simply select Tools|Insert Comma Delimited Table... which will allow browsing for the file and then options on how you would like the data presented.

For an interesting use of this function please see [How To Create a Web Page of Your Work.](#)

## **Dragging & Dropping Files**

Included into HTML Notepad 2.1 (it was included in version 2.0 but we forgot to document it) is the ability to drag and drop files directly into the editor.

The default behaviour in the editor is to insert the new file into the text. By pressing the Control key while dropping a new file will be created.

*How to...live*

*Without deviation progress is not possible*

**Frank Zappa (but taken from a Carl Cox single cover)**



## *How to...get started in HTML Notepad*

Well first of all hello and thanks for trying out HTML Notepad. Now to get you started on a HTML Hacking career.

Select File|New (or use the toolbar) you will get a dialog box which shows all the files in the template directory. Initially there should only be two but it is intended that you will edit the default templates and also add your own.

Select initial.htm(1) this will start the file with the default HTML setting of a HTML, HEAD and BODY with some meta tags which can be used by certain search programs to identify the designer or software which created the program. There is no real need to understand these yet.

Move into the area between the <BODY> and </BODY>. These areas identify the main body of the text. Type Hello World. Now select this text and do one of three things to make this a heading (Press Alt+1, Choose HTML|Headings|H1, or choose the H1 button on the toolbar).

You have now created your first HTML page.

Save this with whatever filename you wish and off you go.

The rest of this help file has references for all HTML tags and, we hope, a lot more useful information for you.

### *How to... produce lists*

First of all create your list by typing text and separating each item with a carriage return (this is especially important for the last line as we always miss this one off).

Once completed select all the text including the last members carriage return.

A tip here is to draw the mouse down until you see the cursor at the start of the line below.

Then choose to either create a numbered or bulleted list by selecting List|Ordered <OL> or List|Unordered <OL> respectively.

Once you have created a complete list you may wish to indent sub-headings this can be done by selecting the list items and select List| .... Indent for whichever type of list you have created. You can also include new list items by inputting them directly and then selecting List|List Item.

List types allow you to change the listing format so instead of having 1,2... you can have a,b... or i,ii... etc. These should be placed inside the list tags such as:

`<ol type="I">`

For examples of this please see;

(HTMLNotepadDirectory)/examples/list.htm

### *How to...form an Automatic Table Of Contents*

The Automatic Table of Contents in HTML Notepad forms a collection of names (internal links) within the document. The first step is therefore to name all the headings in the document which would be a slight chore so we have included an automated system for doing this as well.

So in a HTML file with no names only headings used the function Tools|Name Headings to produce internal links for all headings or even a specific group.

Once these internal links are created the cursor can be positioned anywhere in the document and the function Tools|Table Of Contents will build a table of contents. This is constructed as a simple list of items separated by carriage returns which is the format required for producing a list but we have purposely not tied the designer to that format.

For an example of producing a table of contents please see files:

HTMLNotepadDirectory/examples/toc1.htm ... toc4.htm

### *How to...set up a browser*

Previous versions have been tied to the Netscape browser for functions such as reload on save. This version of the software is not tied to a single browser but this added functionality has come at extra complexity for the user.

If all you require is the Test Pages (or Glasses Toolbar Button) function then the setup is simply to select Tools|Browser Setup and then the Add.. button. You must then search for the executable file of your browser. This should be added to the list of setup browsers. You must then select the browser you wish to use as the default. (The design of this system closely mirrors that of a printer which is the closest metaphor we could find). Once selected you can then test a saved page.

HINT: For quick testing of pages we find Internet Explorer to be the simplest and fastest loader. Once you have finished developing your pages other browsers could (and should) be used.

For the reload upon save option to work (File|Options reload on save checkbox) the reload value of the browser must be ascertained. All Windows application have messages being sent to them continually in the form of mouse movements. Each mouse movement or scroll bar change has a unique identifier as does each menu item. The Reload menu item on the browser is what is needed to perform the reload on save option. Within the browser setup dialog box we have included three common browsers reload values (of which the Netscape value is the most stable, Internet Explorer for instance has 27 for Win3.1 and 28 for Win95 both types being Version 2) but we also allow the user to specify the value themselves.

The only way to calculate the reload menu item for browsers is to load their menus into a resource editor (which is part of a compiler system). If you have access to such a system and are using a browser we have not specified with a reload value we appear not to know please get in contact.

### *How to...Create a Web Page of Your Work*

This uses the user defined log file and Insert Comma-Delimited Text to form a HTML document of your work.

Firstly use the log file to remember all the files edited with HTML Notepad over a particular time period. This file can be saved anywhere. Then select Tools|Insert Comma Delimited Text and browse for the log file. Once selected and the options for forming the table set. This creates a table of all work done over the time period.

*Place this on the web and impress customers on how hard you work !*

## *How to...do JavaScript*

JavaScript, a programming language which sits upon your web page and does some cute things.

HTML Notepad does not encompass the vast majority of the attributes of JavaScript as most of the form tags and other sections would require massive alterations which would not benefit the vast majority of users.

```
<script language="JavaScript">
<!--//hide script from old browsers
    document.write( "<br>Last updated "+ document.lastModified );
//end hiding contents --->
</script>
```

The above script has the wonderful ability of adding the last time the document was modified to a page. If like us you are forever forgetting to change this section of the page, it will do it always for you.

The <script> tag encloses the complete script. The language="JavaScript" section specifies the scripting language (the LiveScript name is now out of date). If you are using Netscape 3 and above the script tag can also point to a SRC document (\*.js) which contains the JavaScript. This has one obvious benefit in that users cannot see your code by using document source.

```
<script language="JavaScript" src="docmod.js">
</script>
```

JavaScript is an event driven scripting language. The above examples are simple scripts which reside at specific points within the page. Most JavaScript programs will instead have functions which are called when events are performed within the web page. For instance the code specified below is a script function which actually creates a web page. If you load this particular page into a browser you would actually form the JavaScript pages defined by the doc.write. The function HTMLout (created automatically by HTML Notepad) is called within the body tag when the page has been loaded.

```
<html><head>
<script language="JavaScript">
// JavaScript function for outputing a html page send it a
// doc file and it will be created. See help file for more details.
// Oh nearly forgot, Generated by HTML Notepad V2.1 1996.
function HTMLout( doc ) {
    doc.write( '<html> ');
    doc.write( '<head> ');
    doc.write( '<title>Example</title> ');
    doc.write( '</head> ');
    doc.write( '<body bgcolor="\#ffffff" > ');
    doc.write( '<h1 align="center">Look I am created by JavaScript</h1>' );
    doc.write( '</body> ');
    doc.write( '</html> ');
}
</script></head>
<body onLoad="HTMLout(document)"></body></html>
```

### Event Handler

JavaScript uses event handlers to interface with events (some of which are included in HTML Notepad).

The below code uses the onBlur event upon the text field to place an alert box upon the screen with whatever string

the user has specified. The onBlur event is when the text box loses focus.

```
<html><head>
<title>Second JavaScript with functions</title>
<script language="LiveScript">
<!--//hide script from old browsers
    function getname(str) {
        alert("Hi, "+str+"!");
    }
//end hiding contents --->
</script></head>
<body>Please enter your name:
<form>
<input type="text" name="Button1" value="name" onBlur="getname(this.value)" >
</form></body></html>
```

The JavaScript language has a large number of these events which the JS programmer can use to call her own functions. A useful resource for both JavaScript and Java is <http://www.gamelan.com> where many pieces of further example code can be found.

Tips.

If you use a JavaScript program to draw a new page with a JavaScript function within it you need to break up the </script> closing tag onto two separate lines.

The parameter 'opener' can be used to find the parent window of a child.

On a purely personal level any JavaScript program which require multiple child windows should not be a JavaScript but a Java program. The debugging tools of JavaScript are currently non-existent and the relationship between variables in different windows is ill-defined.

The Jscript (Microsoft dialect of JavaScript) is currently (MSIE 3.0) dissimilar to JavaScript and programs which run within Netscape will not reliably run within Internet Explorer. Testing is paramount.

