

Chapter 11 Using the Source Editor

This chapter describes the basic functionality of the GoLive CyberStudio *Source Editor*, explains its interaction with the *Palette*, provides information on selected menu commands, and gives instructions on how to edit Web pages in source mode. You should be familiar with the contents of this chapter before starting to edit HTML code in source mode.

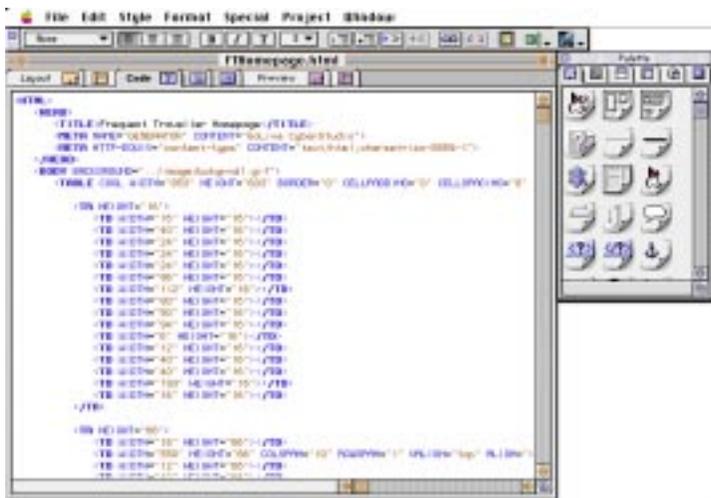
General

The GoLive CyberStudio *Source Editor* is a convenient text-oriented Web page editor that makes your GoLive CyberStudio Web publishing package complete.

The *Source Editor* adds more than basic HTML editing functionality to GoLive CyberStudio, allowing you to view your document's HTML source code, check the HTML syntax, fine-tune your code, or even build Web pages from scratch.

A typical *Source Editor* screen appears below.

The Source Editor Screen



The *Source Editor* comes with a set of powerful features that provides valuable assistance to the HTML-savvy user. Syntax highlighting, drag & drop, and other tools ensure that HTML source code editing is not the tedious process it used to be.

User Profiles

The GoLive CyberStudio *Source Editor* has been designed with the seasoned HTML programmer in mind.

If you are a *seasoned HTML programmer* with a good working knowledge of Hypertext Markup Language and have been using other text-oriented HTML editors, you will find the *Source Editor* very useful for tweaking and fine-tuning the underlying HTML code.

After completing the layout of your Web page in *Layout* mode, you may want to use the *Source Editor* to check the structure of your HTML code and make minor corrections, if necessary. You can use the *Source Editor* to your advantage to add, for example, custom attributes or edit string values.

Basic Features

The editing functionality of the *Source Editor* is similar to that of any other text editor for the Macintosh:

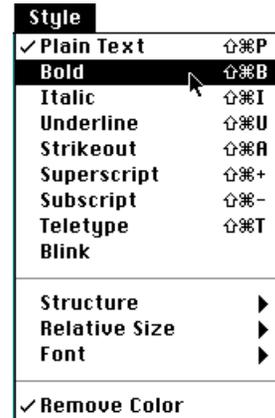
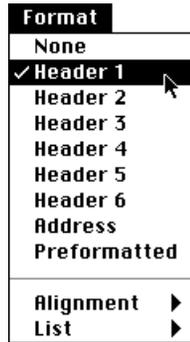
- You can create a new document within the *Source Editor* by choosing the New command from the *File* menu. A blank window will be opened on the screen, allowing you to type or paste text from another document, application, or text clippings from the Finder.
- After you have done some basic editing, you can save your new document using the *Save* command from the *File* menu.
- You can also open an existing GoLive CyberStudio, HTML, or text document using the *Open* command from the *File* menu.
- Typing inserts characters at the insertion point, which is displayed as a blinking vertical bar. Selected text is replaced by the first typed character, and the insertion point is placed after the first typed character.
- Selected text in the frontmost (active) window is highlighted using the default highlight color selected in the Color control panel.
- As you type, the *Source Editor* scrolls the window's contents horizontally or vertically so that the insertion point is visible.

Text Formatting

The *Source Editor* allows you to apply the basic text formatting commands from the *Format* and *Style* menus (see *The Style and Format Menus* on page 80 for reference) to selected text in the *Body* section of your page.

The following screenshots show the text formatting commands available in the *Format* and *Style* menus.

*Source Editor Commands
in the Edit Menu*



To format a paragraph in *Source* mode, proceed as follows:

- 1 Triple-click the paragraph to highlight it.
- 2 Go to the *Format* menu and choose the desired paragraph style.



To format selected text in *Source* mode, proceed as follows:

- 1 Double-click the text to highlight a single word, or click and drag to highlight more text.
- 2 Go to the *Style* menu and choose the desired character style.

Syntax Highlighting

Syntax highlighting works in the background, continuously monitoring user input and comparing it with the content of the tag database.

When the syntax highlighting feature recognizes a tag, it highlights the start and end tags and tag attributes to indicate that the syntax is complete.

Syntax Highlighting in the Source View

The default color for HTML tags is blue.

The default color for tag attributes is brown.

The default color for visible content is black.

```

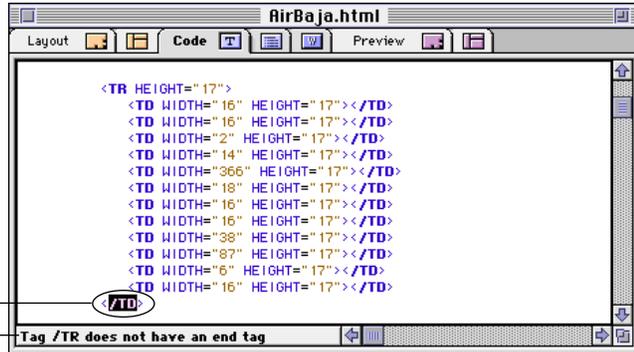
<HTML>
<HEAD>
  <TITLE>Frequent Travelers Department</TITLE>
  <META NAME="GENERATOR" CONTENT="GoLive CyberStudio">
  <META HTTP-EQUIV="Content-type" CONTENT="text/html; charset=iso-8859-1">
</HEAD>
<BODY>
  A niche for business and pleasure travelers who spend much of
  their time on the move, the Frequent Travelers department offers
  the expertise of frequent flyer guru Peter Smith's Frequent Flyer
  services and inside tips from investigative reporter Jody Rothchild.
  <P>
  <IMG SRC="../../../image/greenberg.gif" ALIGN="middle"><P>
  Get ready to laugh as columnist Tony Johnson announces his annual
  Dubious Achievement Awards. It's time for the annual celebration
  of the f... stup... most b... thingr... have har...
  heer
  
```

Syntax highlighting uses eight different colors to visually separate tags, attributes and document content, thus making it easy for authors to keep track of their HTML code. An example of highlighted HTML code is shown above.

Syntax Checking

Syntax checking ensures that your Web pages contain only valid and error-free code. The built-in syntax checker parses your HTML code against the coding rules stored in the tag database and highlights code elements presumed faulty.

Syntax Checking in the Source View



The syntax checker has found a faulty end tag.
The status line identifies the fault.



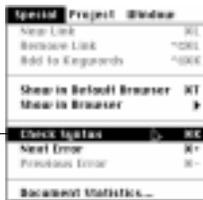
To launch the syntax checker, proceed as follows:

- 1 Choose the *Check Syntax* command from the *Special* menu.
- 2 The syntax checker should highlight the first code element presumed faulty now.
- 3 Check the status line at the lower left corner of the window to learn more about the error.
- 4 Correct the error.
- 5 Choose the *Next Error* command from the *Special* menu or press Command-Option-“+” (the plus sign on the numeric keypad) to continue to the next error.

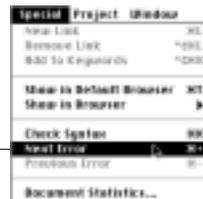
To return to the last error, choose the *Previous Error* command from the *Special* menu or press Command-Option-“-” (the minus sign on the numeric keypad).

Syntax Checking Commands in the Special Menu

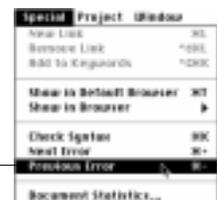
Choose Check Syntax to launch the syntax checker.



Choose Next Error to continue.



Choose Previous Error to go backward.



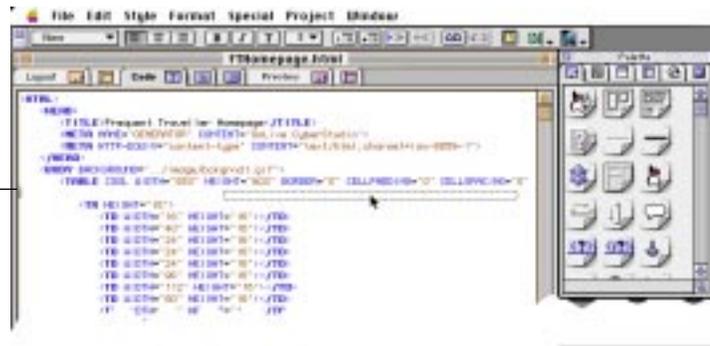
Drag & Drop Support

Drag & drop insertion is available for *Body* and *Forms* tags from the *Palette* as well as text clippings from the *Finder* or any other software that supports Macintosh drag & drop.

The following screenshot illustrates the drag & drop insertion of a line tag from the *Palette*.

Drag & Drop Insertion of a Line Tag

While you are dragging, the cursor moves along the margin to indicate the current insertion point.



The *Source Editor* lets you drop *Palette* items in the flow of your HTML code and use them as editable tag templates.

After inserting a tag, you can edit its attributes and attribute values while the *Source Editor* monitors your input and highlights typing errors or incomplete syntax.

Drag & drop coloring is also supported. You can drag a color from the preview pane of the *Color Picker* (see *Coloring Text and Objects* on page 301) and drop it on your HTML code.

Further drag & drop options include pages and URLs from the *Project* window. Dropping a page or URL icon creates a link instantly, referencing the page or URL you have dropped.

Automating Operations Using AppleScript

*GoLive CyberStudio's
AppleScript Dictionary*

Click here to view an AppleScript command.



The following sections outline how GoLive CyberStudio's *Source Editor* works with AppleScript to automate tasks.



To use AppleScript, you need the AppleScript extension installed in your Extensions folder as well as the Script Editor.

You also need a working knowledge of AppleScript to use the following script command reference to your advantage. For instructions on how to use AppleScript, please refer to the appropriate documentation available from Apple Computer.

Containers

If the user doesn't specify any container, the script addresses the document with the active window. If this document is not in HTML mode, the script automatically switches to HTML mode. Scripts can also address a particular window or document directly, using either its position on screen or its name.

Lines, Characters, and Selections

The *Source Editor* window contains Objects of the Line and Character types. These objects represent physical lines or characters, which can be addressed as single entities or in groups. You can use the usual AppleScript syntax to address lines and characters:

```
Line 1  
Character 1234  
the last Line  
Characters 10 thru 20 of Line 11.
```

The *Character* object exists either at the document or line level. At the document level, it refers to a single character within the document, starting at position 1. At the line level, it refers to a single character on the specified line, starting at position 1. Both the line and column or character specifications start at position 1.

The *Selection* object reflects the current selection. The selection is invalid if nothing is selected. This sets the *Is Visible* property of the selection to *false*.

A selection can be created in many ways—for example, by manipulating the *Starting Point* or *Ending Point* properties of the *Selection* object or by using the *Select* command. The *Select* command requires a line or character object or a from/to specification such as “from {line,column} to {line, column}”. All Paste operations select the pasted text by default.

The current selection can be deleted using the *Delete Selection* or *Delete* commands, while the *Deselect* command cancels the current selection.

All objects have *Property Contents* to allow direct access to their content. The *Length* property indicates the number of characters contained in the object.

Line and *Character* objects have additional common properties. The *Index* property indicates the number of an object, the number of a line, or the position of a character within a document.

Cursor Positioning

The *Position* property is also common to both objects. This property contains a list made up of two elements and uses a {line, column} syntax. There are several ways to specify the cursor position within the editor window. You can use the *Go to* command to place the cursor in a line or on a character. *Go to* expects a *Line* or *Character* object as its function argument:

Go to Character 1 **of** Line 15

Go to Line 10

Go to **the last** Character

A more direct approach to positioning uses the *Line Index* and *Column Index* properties. These properties contain the current line and column, and they can be read out and set:

set This Line to 123

Core Elements

Objects

Window

The *Window* object supplies information on a window. It is comprised of the pertinent document objects.

Bounds	<i>Rectangle</i>	<i>R/W</i>	<i>Window size and position.</i>
Has Closebox	<i>Bool</i>	<i>R/O</i>	<i>True if the window has a close box.</i>
Has Titlebar	<i>Bool</i>	<i>R/O</i>	<i>True if the window has a title bar.</i>
Hide on Deactivate	<i>Bool</i>	<i>R/O</i>	<i>True if the window disappears upon deactivation.</i>
Is Floating	<i>Bool</i>	<i>R/O</i>	<i>True if the window is a floating window.</i>
Is Resizable	<i>Bool</i>	<i>R/O</i>	<i>True if the window has a growbox.</i>
Is Zoomable	<i>Bool</i>	<i>R/O</i>	<i>True if the window has a zoombox.</i>
Index	<i>Long</i>	<i>R/O</i>	<i>Index of windows in the window list (starts at 0).</i>
Title	<i>Text</i>	<i>R/O</i>	<i>The window name.</i>
Visible	<i>Bool</i>	<i>R/O</i>	<i>True if the window is visible.</i>

Methods

Close	<i>Object</i>	<i>Closes the specified object (window or document).</i>
Open	<i>Object</i>	<i>Opens the specified object.</i>
Save	<i>document</i>	<i>Saves a document.</i>
	<i>[File filename]</i>	<i>The name of the file to be saved.</i>
New Document		<i>Opens a new document.</i>
Activate Window	<i>window</i>	<i>Activates a window.</i>

General Text Handling Elements

Document

The document object is comprised of the entire document. It contains general properties, such as the current selection.

Clipboard	<i>Text</i>	<i>R/W</i>	<i>This property allows for accessing the clipboard. It can be used as a function argument for the Insert Text method.</i>
Contents	<i>Text</i>	<i>R/W</i>	<i>The content of the document.</i>
Length	<i>Long</i>	<i>R/O</i>	<i>The length of the text in the document.</i>
Font Name	<i>Text</i>	<i>R/W</i>	<i>The name of the font used.</i>
Font Size	<i>Short</i>	<i>R/W</i>	<i>The size of the font used.</i>
Font Face	<i>Enums</i>	<i>R/W</i>	<i>The attributes of the current font. The list includes the Bold, Italic, Underline, Outline, Shadow, Condense and Extend elements. You can omit the braces if you wish to specify a single attribute only.</i>
Tab Stops	<i>Short</i>	<i>R/W</i>	<i>The tab stop position.</i>
MIME Type	<i>Text</i>	<i>R/O</i>	<i>The MIME type of the document. This function returns "text/html" by default.</i>
Syntax Highlighting	<i>Bool</i>	<i>R/W</i>	<i>Toggles Syntax Highlighting on and off.</i>
Auto Indenting	<i>Bool</i>	<i>R/W</i>	<i>Toggles Auto Indenting on and off.</i>
Line Index	<i>Long</i>	<i>R/W</i>	<i>The number of the current line.</i>
Column Index	<i>Long</i>	<i>R/W</i>	<i>The current column.</i>
Current Line	<i>Object</i>	<i>R/O</i>	<i>The current line as a line object.</i>
Selection	<i>Object</i>	<i>R/O</i>	<i>The current selection.</i>
Status Line	<i>Text</i>	<i>R/W</i>	<i>The content of the status line.</i>

Line

The *line* object refers to a specific line.

<i>Index</i>	<i>Long</i>	<i>R/O</i>	<i>The line number, starting at 1.</i>
<i>Position</i>	<i>list</i>	<i>R/O</i>	<i>The Position as a list with the {line, 1} syntax.</i>
<i>Contents</i>	<i>Text</i>	<i>R/W</i>	<i>The content of the line.</i>
<i>Length</i>	<i>Long</i>	<i>R/O</i>	<i>The length of the text on the line.</i>

Character

The *Character* object refers to a specific character.

<i>Index</i>	<i>Long</i>	<i>R/O</i>	<i>The position of the character, relative to the beginning of the document at position 1.</i>
<i>Position</i>	<i>list</i>	<i>R/O</i>	<i>The position as a list with the {line, column} syntax.</i>
<i>Contents</i>	<i>Text</i>	<i>R/W</i>	<i>The content of the character.</i>
<i>Length</i>	<i>Long</i>	<i>R/O</i>	<i>The length of the character (constant 1).</i>

Selection Object

The *Selection* object exists all the time, even if there is no current selection. The *Is Visible* property indicates whether the selection exists or not. The selection can be modified by changing the *Starting Position* and *Ending Position* properties. Changing the *Contents* property replaces the current selection with the specified text.

The *Selection* object can be addressed through the *Selection* property, not by its name.

<i>Starting Position</i>	<i>list</i>	<i>R/W</i>	<i>The beginning of the selection.</i>
<i>Ending Position</i>	<i>Long</i>	<i>R/W</i>	<i>The end of the selection.</i>
<i>Is Visible</i>	<i>Bool</i>	<i>R/O</i>	<i>True if a selection exists.</i>
<i>Contents</i>	<i>Text</i>	<i>R/W</i>	<i>The content of the selection.</i>
<i>Length</i>	<i>Long</i>	<i>R/O</i>	<i>The length of the text in the selection.</i>

Methods

Insert Text	"Text"	Inserts the function parameter as text at the current position.
Insert File	filename	Inserts the specified file at the current cursor position.
Find	"String"	Searches for the specified string. The result is true if the string has been found. The found string will be selected.
	[Applying] Option(s)	Applies the specified options. One or more options may be specified. Multiple options must be specified as a list in braces.
	Available Options:	
	Ignore case	Distinguishes between uppercase and lowercase.
	Wraparound	Wrap around the beginning or the end of the document.
	From Top	Start the search at the beginning or the end of the text or the current selection.
	Backwards	Search direction is backwards.
	Word	Search word by word.
Replace	"String"	Searches for the specified string and replaces it with another string. The result is true if the string has been found and replaced. The found and replaced string will be selected. If the direct parameter is omitted, the current selection is replaced with the specified string.
	Using "String"	The string used for replacing.
	[Applying] Option(s)	Applies the specified options. One or more options may be specified. Multiple options must be specified as a list in braces.
	Available Options:	
	Ignore case	Distinguishes between uppercase and lowercase.
	Wraparound	Wrap around the beginning or the end of the document.
	From Top	Start the search at the beginning or the end of the text or the current selection.
	Backwards	Search direction is backwards.
	Word	Search word by word.

Undo			<i>Undoes the last changes.</i>
Redo			<i>Restores data to the state before the last Undo action.</i>
Delete		<i>Object</i>	<i>Deletes the specified object, which may be the selection object, a line, or other. If the object is not specified, it deletes the current selection.</i>
Select		<i>Object</i>	<i>Makes a selection. The specified object may be a suite of lines or characters—for example, “Lines 3 thru 7”.</i>
	<i>From</i>	<i>List</i>	<i>Alternatively, the selection may be specified using two lists with a {line, column} syntax for the beginning and end of the selection.</i>
	<i>To</i>	<i>List</i>	
Select paragraph			<i>This command searches a blank line before and after the current line and selects the text between the two blank lines and the subsequent blank line, positioning the cursor on the blank line. In the example given below, the cursor is placed on the line “1st Line”. The asterisks mark the lines that will be selected after executing this command.</i> <i>1st line</i> <i>* 2nd line</i> <i>* 3rd line</i> <i>* 4th line</i> <i>etc.</i>
	<i>[Between</i>	<i>Object]</i>	<i>Optionally, the selection can be confined to particular lines by specifying a range explicitly. Example: “Select Paragraph Between Lines 3 thru 5”.</i>
Deselect			<i> Cancels the current selection.</i>
Go to		<i>Object</i>	<i>Sets the current position to the specified line or character object.</i>
Delete spaces			<i>Deletes all spaces and tabs from the current position to the end of the line or to the next character that is neither a space nor a tab.</i>

HTML Extensions

Methods

The following methods wrap the current selection in a start tag and end tag of a specified type. Multiple tag types can be specified using a list. The following types are supported:

Style	Tag Type
<i>Address</i>	<ADDRESS>selection </ADDRESS>
<i>Blink</i>	<BLINK>selection </BLINK>
<i>Blockquote</i>	<BLOCKQUOTE>selection </BLOCKQUOTE>
<i>Bold</i>	selection
<i>Citation</i>	<CITE>selection </CITE>
<i>Code</i>	<CODE>selection </CODE>
<i>Definition</i>	<DFN>selection </DFN>
<i>Emphasis</i>	<EMP>selection </EMP>
<i>Italic</i>	<I>selection </I>
<i>Keyboard</i>	<KBD>selection </KBD>
<i>Listing</i>	<LISTING>selection </LISTING>
<i>Preformatted</i>	<PRE>selection </PRE>
<i>Sample</i>	<SAMP>selection </SAMP>
<i>Strikethrough</i>	<S>selection </S>
<i>Strong</i>	selection
<i>Subscript</i>	_{selection}
<i>Superscript</i>	^{selection}
<i>Teletype</i>	<TT>selection </TT>
<i>Underline</i>	<U>selection </U>
<i>Variable</i>	<VAR>selection </VAR>

Style Tag	“String”	The general form of the Style method uses the specified text as a tag. You can also specify attributes: Style Tag “A HREF=#Marker” selection
Heading	Size	Formats the current selection as a header. Size may range between 1 and 6.
	[with Alignment] Left Right Center	Alignment options: Left Right Center
Paragraph	[Align]	Inserts a P tag at the current cursor position.
	Left Right Center	Alignment options: Left Right Center
Base Font	Size	Inserts a BASEFONT tag at the current cursor position. Size may range between 1 and 7.
Font	[“Name”]	Wraps the current selection in a FONT tag. The name attribute is the font name.
	[Size Word]	Font size. This size ranges between 1 and 7.
	[with Smaller]	Reduces visible font size by using a negative relative size. Assumes -1 if not specified.
	[with Larger]	Reduces visible font size by using a positive relative size. Assumes +1 if not specified.
List	Art	Formats the current selection as a list. The following list styles are available
		Ordered numbered list (OL) Unordered simple list (UL) Definition definition list (DL/DD) Term definition list (DL/DT) Directory Menu
	[Type type]	The type of bullet can be specified for simple and numbered lists:
		Simple lists: Disc solid bullet Circle hollow bullet Square solid square

			<i>Numbered lists:</i> <i>Arabic</i> <i>Upper roman</i> <i>Lower roman</i> <i>Upper alpha</i> <i>Lower alpha</i>
Table	["Title"]		Formats the current selection as a table, converting an optional title into a CAPTION tag. Each selected row is converted to one or more table cells.
	[Separator	"characters"]	Each row consists of one or more cells separated by the specified character. Common separators include the comma or tab character. If the number of columns has not been specified explicitly, the script determines the maximum number of entries in any one line of text within the highlighted section. Example: The following lines create a table with three cells per row when specifying "," as a separator: One Two,three Four,five,six Seven,eight
	[Rows	Word]	Row count. If the specified row count exceeds the maximum number of cells that can be generated out of the text, the script generates blank cells.
	[Columns	Word]	Column count. If the specified column count exceeds the maximum number of cells that can be generated out of the text, the script generates blank cells. Specifying the number of columns prohibits the script from determining table width, provided a separator character has been defined.
	[Border	Word]	Specifies the width of the border of the table.
	[Cell spacing	Word]	Specifies inter-cell spacing.
	[Cell padding	Word]	Specifies the cell padding.
	[Width	Text]	Specifies table width. Table width must be entered as text in order to allow for values such as "50%".
	[with Heading]		Specifying this parameter formats the first row of the table as a table header.
Link	"URL"		Defines the current selection as a hyperlink, using the parameter as the destination URL.
Document layout	["Title"]		Converts the current document to HTML, inserting HTML, HEAD, and BODY tags and entering an optional title in a TITLE tag within the HEAD section. The BODY tag can be modified using a variety of options:
	[Background	"URL"]	Specifies the background image.

[Background color	“color”	Specifies the background color of the page.
[Text color	“Color”	Specifies text color.
[Link color	“Color”	Specifies link color.
[Visited link color	“Color”	Specifies the color of links already viewed.
[Active link color	“Color”	Specifies the color of the active link.
[Heading	“Text”	Inserts the specified text as a header into the Body section using a H1 tag.
[Load script	“Text”	Uses the text as a script for the ONLOAD attribute. After the script has been run, the previous first line of the document is the current line.

Check Syntax

Verifies the syntax of the HTML document, returning the number of any error found.

Get Syntax Error

Index

Get syntax error uses an index between 1 and the number of the error found. The faulty code element is highlighted and the error description is displayed in the status line. The following error codes may be returned:

- 0 - Invalid error index
- 1 - Tag expected but not found
- 2 - Unknown Tag
- 3 - Found end tag without start tag
- 4 - No end tag found for specified tag
- 5 - Tag end character “>” expected but not found
- 6 - Found attributes in an end tag
- 7 - Specified tag has no end tag
- 8 - Unknown attribute
- 9 - Required attribute not found
- 10 - Attribute value not found
- 11 - Attribute does not support values
- 12 - Value specified without attribute
- 13 - Illegal value
- 14 - Illegal color code
- 15 - Value is not a number
- 16 - Unknown HTML special character
- 17 - Use “&,” instead of “&”
- 18 - End of file

Source Editor Preferences



The *General*, *Colors*, and *Font* tabs in the *Source* group of the *Preferences* dialog box let you customize various basic settings that influence the behavior and appearance of the *Source Editor*.

To open the *Source* group of the *Preferences* dialog box, proceed as follows:

- 1 Go to the *Edit* menu and choose *Preferences*.
- 2 Locate and click the *Source* icon on the scrolling sidebar.
- 3 Click the tab with the desired options.

General Settings

The *General* tab contains the following options:

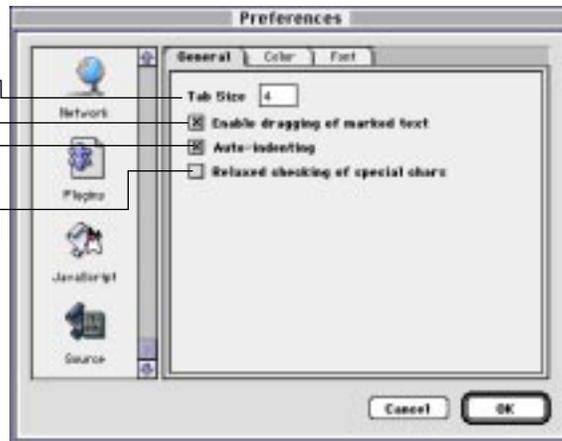
Selecting Code Preferences

Enter the indentation (in character widths) for lower-level tags.

Use this checkbox to toggle drag & drop on and off.

Use this checkbox to toggle the boldface screen font for HTML tags on and off.

Use this checkbox to toggle the relaxed checking of special characters on and off.



The *Tab Size* text box sets the width of the indentation (in characters) added to lower-level tags.

The *Enable dragging of marked text* checkbox toggles the internal drag & drop support on and off.

The *Auto-indenting* checkbox toggles the automatic indenting of lower level tags on and off.

The *Relaxed checking of special chars* relaxes the rules applied to special characters when checking the syntax. Enabling this option allows you to use uppercase-only characters as found in older HTML files and instructs the syntax checker to stop looking for a trailing semicolon. Example: The `<tag>` string will be interpreted as `<tag>`.

Syntax Highlighting Preferences

Customizing Syntax Highlighting

Use this checkbox to turn syntax highlighting on and off.

Use this checkbox to toggle the boldface screen font for HTML tags on and off.

The *Colors* tab contains the following options:



The color section allows for viewing the default colors used for syntax highlighting.

The *Syntax Highlighting* checkbox toggles syntax highlighting on and off.

The *Bold Typeface for Tags* checkbox selects a bold typeface for tags to make them stand out more clearly from the rest of the code.

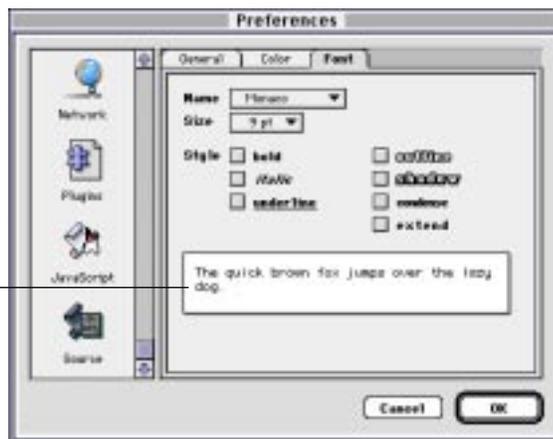
Font Preferences

Selecting a Custom Font

Use these options to select a custom font and style for plain text displayed in the editor window.

Preview your font preferences here.

The *Fonts* tab contains the following options:



You can select a custom font, font size, and font style for your HTML code to override the default *Monaco 9pt plain* setting.

