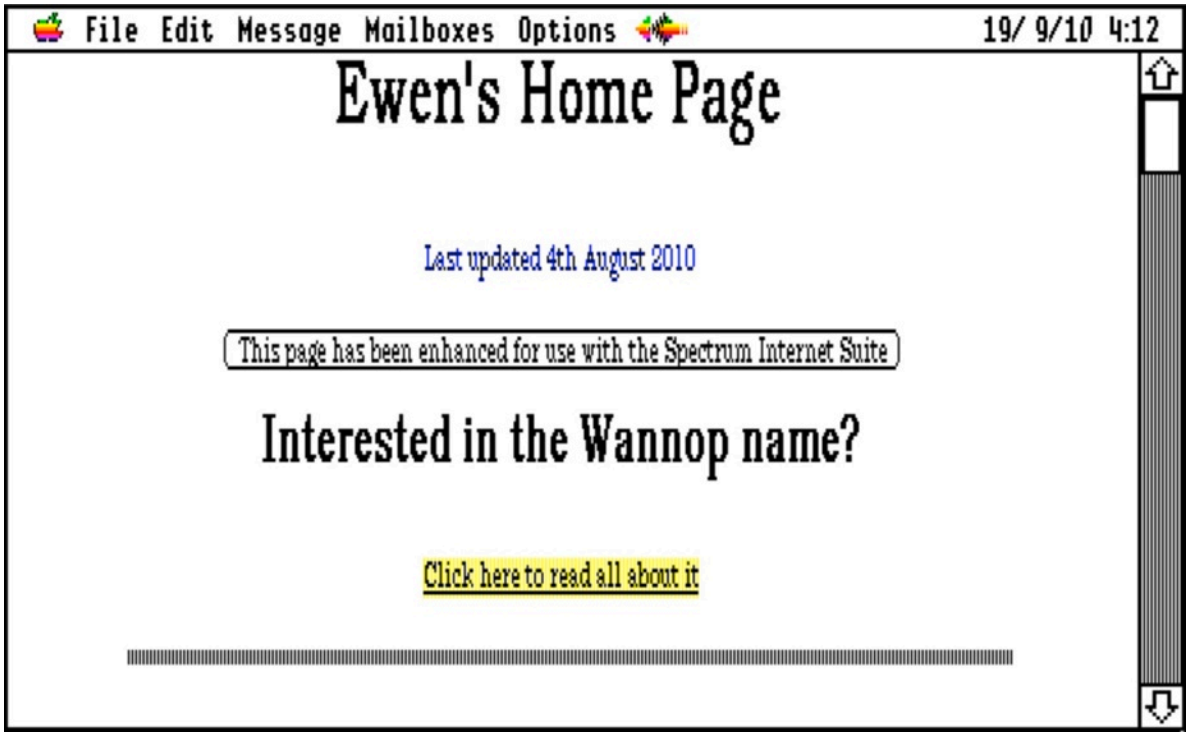
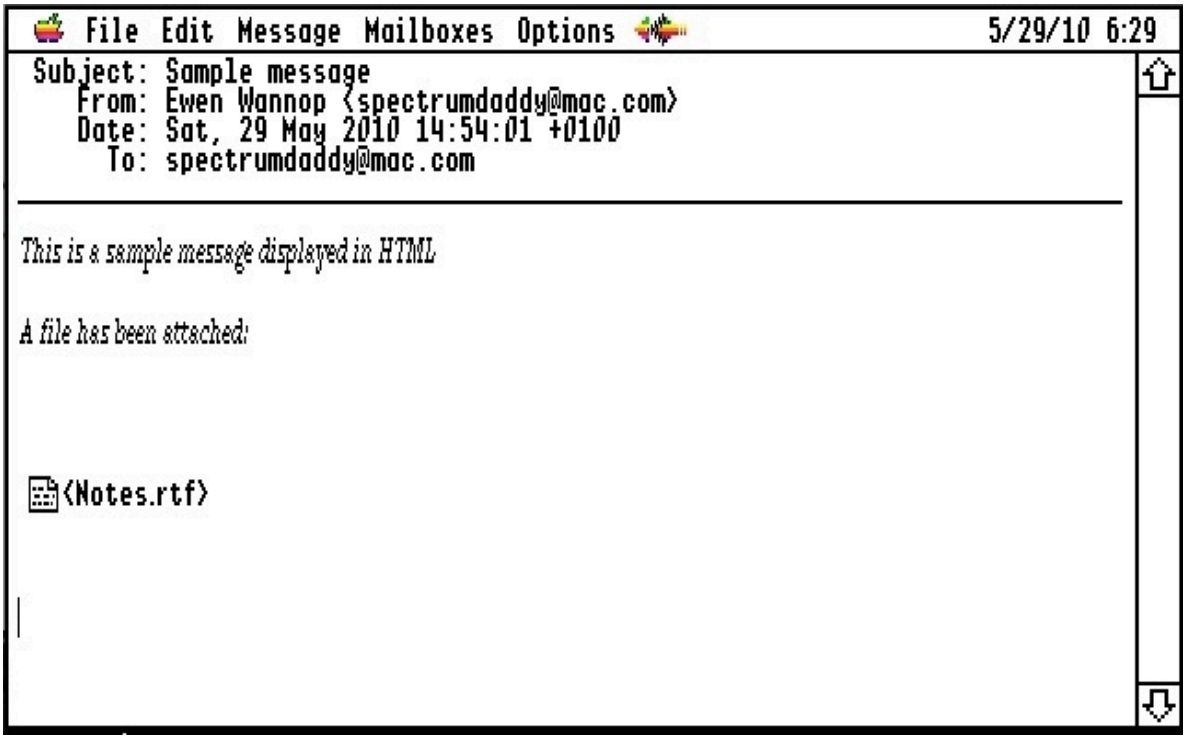
Apple® IIGS

HTML Tool Set Reference Manual





Dedicated to the memory of Joe Kohn 1947-2010

Contents

Introduction to the HTML Tool Set

Back in 1996, after a late night session at KansasFest, Geoff Weiss and myself took it upon ourselves to write an HTML Browser for the IIgs.

Geoff wrote the Spectrum scripts to make it all work, and I wrote an HTML engine that was used to capture and parse the HTML data to build the display.

Spectrum Internet Suite, or SIS as it is better known was born.

Subsequently I used the source code from the engine to add HTML display features to Spectrum, and more recently, to add a custom HTML display to SAM2.

I have now adapted that original code to produce a stand-alone HTML Tool Set.

The HTML Tool Set has been released into the Public Domain, so that anyone who wishes, can build their own web browser, or display HTML within an application.

Full programming details for using the Tool are given within this manual.

It is up to you to do the rest!

If you have any questions about the use of this toolset, please contact:

spectrumdaddy@mac.com

Using the HTML Tool Set

Requirements

The HTML Tool Set should be started after the application has started any other tools that it requires.

The HTML Tool Set requires these tools to be active:

Control Manager, Dialog Manager, Event Manager, Font Manager, Integer Math Tool Set, Memory Manager, Miscellaneous Tool Set, QuickDraw II, QuickDraw II Auxiliary, Scrap Manager, TextEdit Tool Set, Window Manager

The HTML Tool Set requires several custom fonts to be installed. These are supplied with the original archive:

SIS-1, SIS-2, SIS-3, SIS-4, SIS-5

In addition, the HTML Tool Set can use the ByteWorks Talking Tools, which are available on the OPUS][collection from:

http://www.syndicomm.com

To obtain the HTML Tool Set, and any of my other software:

http://homepage.mac.com/speccie

The HTML Tool Set may be used in two ways, either as an engine to build a TERecord from supplied text containing HTML code, and to further display a TERecord in a window where the user can interact with the display.

In all cases where a TERecord is returned, it is the responsibility of the application to kill the TERecord when it has finished with it.

To use the HTML Tool Set, you should first start up any other tools you may require, making sure those required by the HTML Tool Set have also been started. To prime the Tool Set, call **HTMLInit**, checking that the call was successful.

To parse text containing HTML, call **HTMLParse**, and then on return from the call, either display the TERecord within your own window, or within the optional window opened from the **HTMLOpenDisplay** call using the **HTMLInsertTERecord** call. The optional window is opened full screen, drawn just below the menu bar, and contains one TextEdit Control.

The parsing process builds a display based on **HTTP/1.1** specifications. The IIgs display is fairly limited, so you will find that tables may not display correctly, also any embedded scripting will be ignored. The HTML Tool Set builds a page with various links and other data drawn in a zero width font. This allows data to be contained invisibly within the display. This data can then be later retrieved from the **HTMLEventTask** call when the user double-clicks a link.

For simpler applications, such as the HTML display in the SAM2 email client, the TERecord can be displayed within your own window, but if you intend to build a web browser, you will need to use the optional window so the user has full control over the hidden data in the TERecord.

If URL or Link data is returned from the **HTMLEventTask** call, it is up to the application to verify the data is valid, what kind of data it is, and then to take the appropriate action.

If Form data is returned from the **HTMLGetForm** call, it will be up to the application to take the appropriate action in posting the data.

Using the Calls

Simple HTML Parsing

Call sequence:

```
HTMLInit
HTMLParse
HTMLGetError
```

Using the Optional Window

Call Sequence:

```
HTMLOpenDisplay
HTMLInsertTERecord
HTMLEventTask
HTMLCloseDisplay
```

While the window is open, these calls are active:

```
HTMLClearDisplay
HTMLGetTitle
HTMLGotoName
HTMLFindString
HTMLGetForm
HTMLCheckForForm
```

Note:

Loop on the `HTMLEventTask` call till the user indicates by either an `Esc`, `OA-.`, or `OA-W` key press, that the window is to be closed

The Routines

Housekeeping Routines

`HTMLBootInit`	Initialises the HTML Tool Set; called only by the Tool Locator must not be called by an application
`HTMLStartUp`	Starts up the HTML Tool Set for use by an application
`HTMLShutDown`	Shuts down the HTML Tool Set
`HTMLVersion`	Returns the version number of the HTML Tool Set
`HTMLReset`	Resets the HTML Tool Set; called only when the system is reset must not be called by an application
`HTMLStatus`	Indicates if the HTML Tool Set is active

Global Routines

`HTMLInit`	Called by an application to prime the HTML Tool Set for use
`HTMLParse`	Parses raw HTML text into a TERecord
`HTMLGetError`	Returns the last error

Display Routines

`HTMLOpenDisplay`	Opens an optional display window with a single TextEdit Control
`HTMLCloseDisplay`	Closes the optional display window
`HTMLInsertTERecord`	Inserts a TERecord into the open display window
`HTMLClearDisplay`	Clears any text from the open display window
`HTMLEventTask`	Gives user interaction with the open display window
`HTMLGetTitle`	Returns the page Title
`HTMLGotoName`	Jumps to a Name Link within the page in the open display window
`HTMLFindString`	Finds and jumps to a string within the open display window
`HTMLGetForm`	Returns the Form data from an active form
`HTMLCheckForForm`	Returns the number of active Forms in the page of an open display window

$0182 HTMLBootInit

Initialises the HTML Tool; called only by the Tool Locator.

Warning
An Application must never make this call.

Parameters The stack is not affected by this call. There are no input or output parameters.

Errors None

C Call must not be made by an application.

$0282 HTMLStartUp

Starts up the HTML Tool for use by an application.

Important
Your Application must make this call before it makes any other HTML Tool calls.

Parameters The stack is not affected by this call. There are no input or output parameters.

Errors None

C

```
extern pascal void HTMLStartUp ();
```

$0382 HTMLShutDown

Shuts down the HTML Tool.

Important

If your Application has started up the HTML Tool, the application must make this call before it quits.

Parameters The stack is not affected by this call. There are no input or output parameters.

Errors None

C

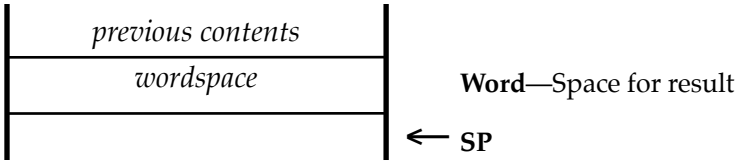
```
extern pascal void HTMLShutDown ();
```

$0482 HTMLVersion

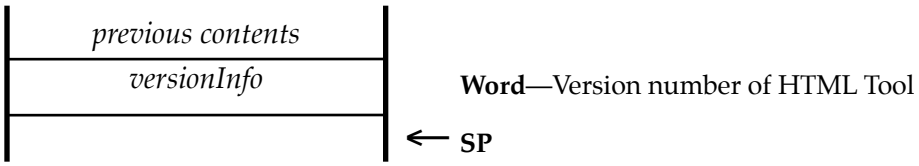
Returns the version number of the HTML Tool.

Parameters

Stack before call



Stack after call



Errors None

C

```
extern pascal Word HTMLVersion ();
Word     versionInfo
```

$0582 HTMLReset

Resets the HTML Tool; called only when the system is reset.

Warning
An Application must never make this call.

Parameters The stack is not affected by this call. There are no input or output parameters.

Errors None

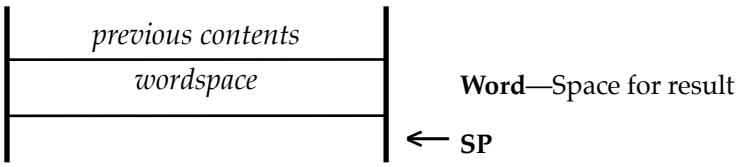
C Call must not be made by an application.

$0682 HTMLStatus

Indicates whether the HTML Tool Set is active.
HTMLStatus returns TRUE if HTMLStartup has been called and HTMLShutDown has not been called. The routine returns FALSE if HTMLStartUp has not been called at all or if HTMLShutDown has been called since the last time HTMLStartUp was called.

Parameters

Stack before call



Stack after call

Errors None

C

```
extern pascal Boolean HTMLStatus ();
Word     activeFlag
```

$0982 HTMLInit

Prepares the HTML Tool for use by an application.

This call must be made by the application before any of the other calls are made.

The call checks that the required fonts are installed, and any required Tools are currently active. It also checks to see if the optional TalkingTools are present.

Parameters

The stack is not affected by this call. There are no input or output parameters.

Errors

$82FE	htNoTools	Required Fonts and Tools not present

C

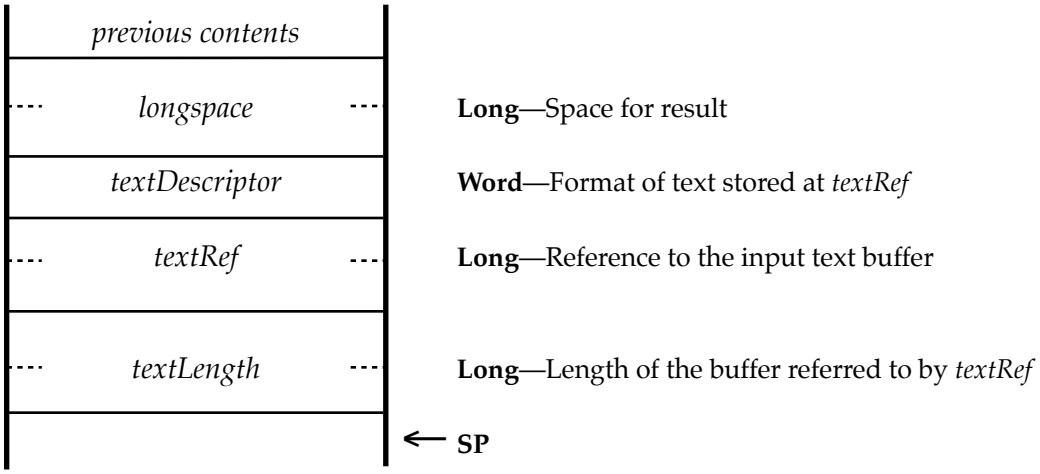
```
extern pascal void HTMLInit ()
```

$0A82 HTMLParse

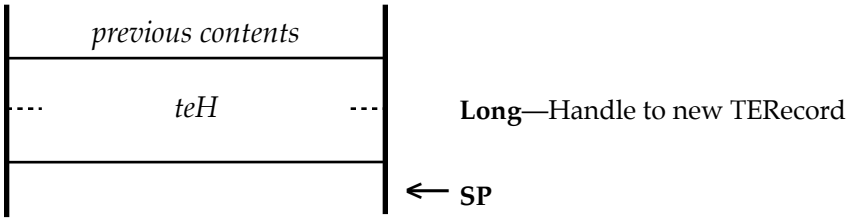
Parses the supplied HTML text into a new TERecord.

Parameters

Stack before call



Stack after call



Errors

$8202	htBadHandle	A bad Handle was supplied
$8206	htNoData	An empty Handle or buffer was supplied
$8207	htFailed	The parsing failed
$8208	htNotInit	HTMLInit has not been called
$82FF	htBadCall	Incorrect parameters supplied

C

```
extern pascal Long HTMLParse (textDescriptor, textRef, textLength);
Long      textRef, textLength, teH
Word      textDescriptor
```

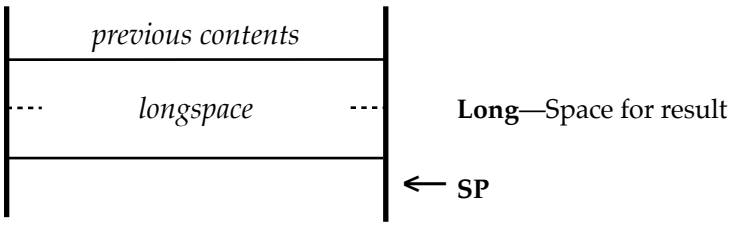
textDescriptor	The format of the text to be parsed, and the type of reference stored in *textRef*.	
Reserved	bits 15-1	Must be set to 0.
dataFormat	bit 0	Defines the format of the text. 0 = *textRef* is a pointer to a text buffer; *textLength* contains the length of the buffer (in bytes) 1 = *textRef* is a handle to the text buffer; *textLength* is ignored
textLength	Length of the buffer referenced in *textRef*. This field is valid only if textRef points to a buffer.	

$0B82 HTMLOpenDisplay

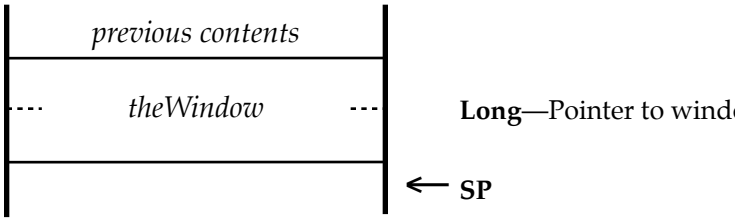
Opens the optional display window. This window is used by many of the calls.

Parameters

Stack before call



Stack after call



Errors

$8202	htWindowOpen	Window open
$8208	htNotInit	HTMLInit has not been called

C

```
extern pascal Long HTMLOpenDisplay ();
Long      theWindow
```

$0C82 HTMLCloseDisplay

Closes the optional display window.

Parameters The stack is not affected by this call. There are no input or output parameters.

Errors

$8204	htWindowClosed	Window closed
$8208	htNotInit	HTMLInit has not been called

C

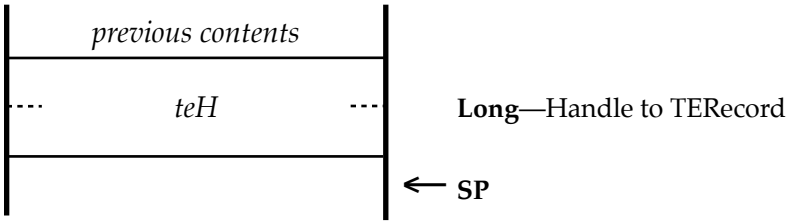
```
extern pascal void HTMLCloseDisplay ();
```

$0D82 HTMLInsertTERecord

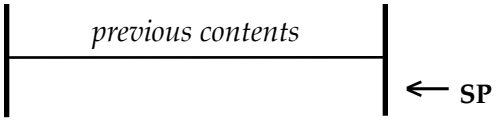
Inserts a TERecord at the end of the text in the optional display window.

Parameters

Stack before call



Stack after call



Errors

$82-2	htBadHandle	Invalid TERecord
$8204	htWindowClosed	Window closed
$8206	htNoData	The TERecord is empty
$8208	htNotInit	HTMLInit has not been called

C

```
extern pascal void HTMLInsertTERecord (teH);
Long     teH
```

$0E82 HTMLClearDisplay

Clears the text from the optional display window.

Parameters The stack is not affected by this call. There are no input or output parameters.

Errors

$8204	htWindowClosed	Window closed
$8208	htNotInit	HTMLInit has not been called

C

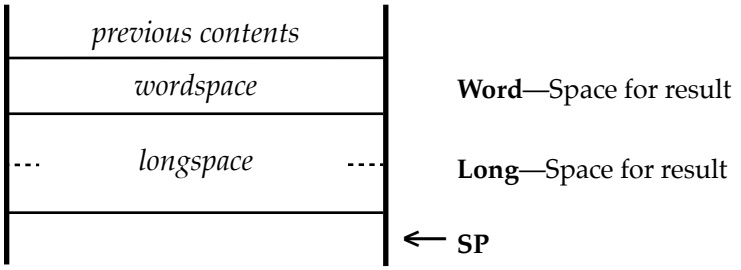
```
extern pascal void HTMLClearDisplay ();
```

$0F82 HTMLEventTask

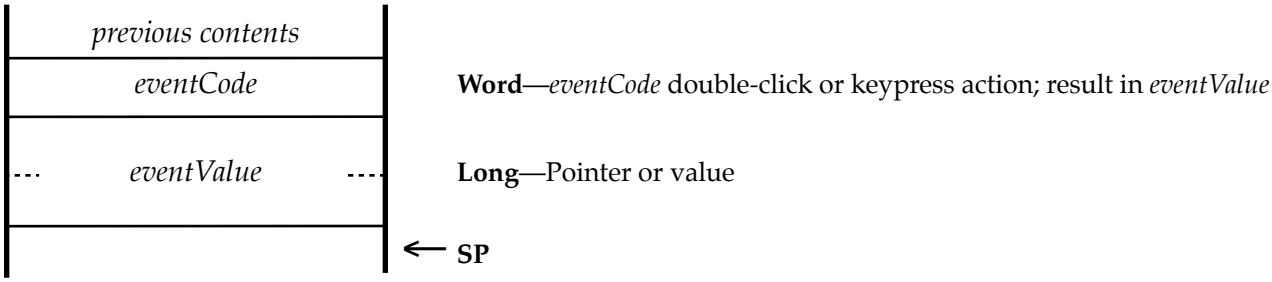
Allows the user to interact with the optional display window.

Parameters

Stack before call



Stack after call



Errors

$8204	htWindowClosed	Window closed
$8208	htNotInit	HTMLInit has not been called

C

```
extern pascal Word Long HTMLEventTask ();
Long      eventValue
Word      eventCode
```

EventRec

eventCode Indicates which item the user double-clicked or which keys they pressed.

$01	URL of a page or email address was double-clicked
$02	URL of an image icon was double-clicked
$05	a GET submit button was double-clicked
$06	a POST submit button was double-clicked
$08	a LINKed icon was double-clicked
$10	Escape key was pressed
$11	OA-. (stop) was pressed
$12	OA-W was pressed

eventValue Returns either a Pointer or a value depending on *eventCode*.

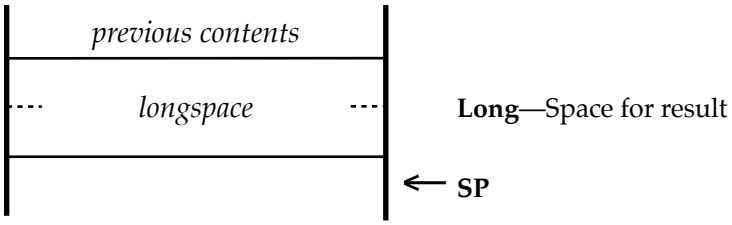
$01	Pointer to String (*wordlength* + *text*)
$02	Pointer to String (*wordlength* + *text*)
$05	LongWord number of a *Form*
$06	LongWord number of a *Form*
$08	Pointer to String (*wordlength* + *text*)
$10	Not applicable
$11	Not applicable
$12	Not applicable

$1082 HTMLGetTitle

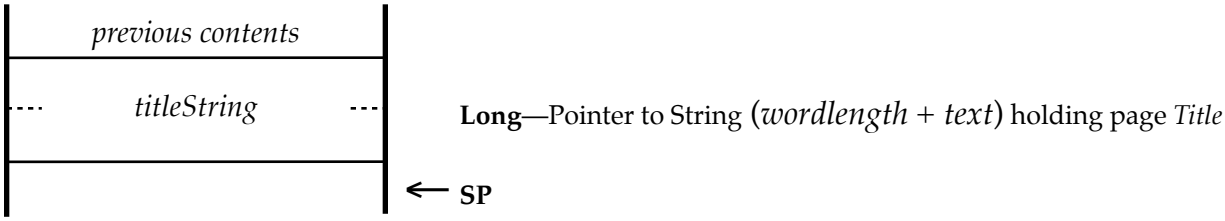
Returns a Pointer to the Title of the page if present. Null if no Title.

Parameters

Stack before call



Stack after call



Errors

$8204	htWindowClosed	Window closed
$8208	htNotInit	HTMLInit has not been called

C

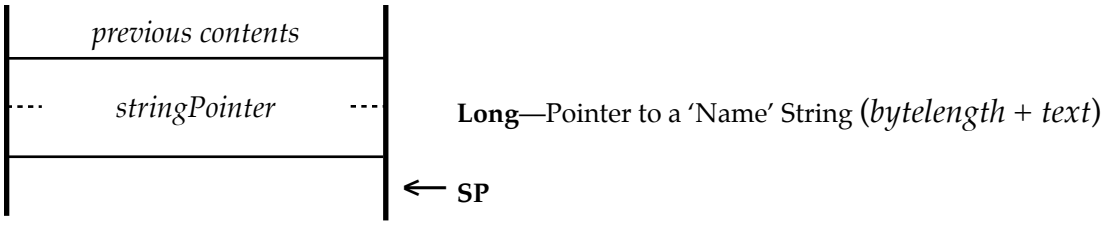
```
extern pascal Long HTMLGetTitle ();
Long     titleString
```

$1182 HTMLGoToName

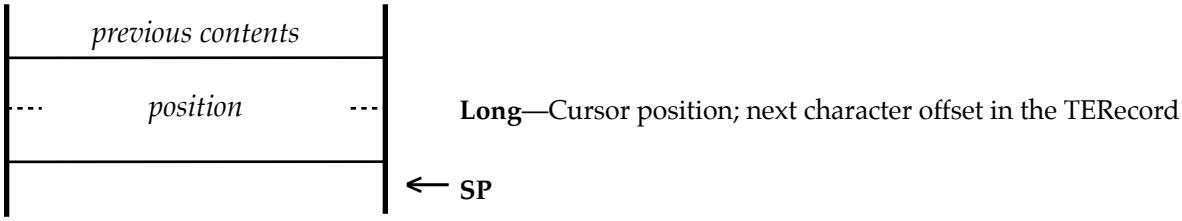
Jumps to a Name Link *<A NAME="Name">*.

Parameters

Stack before call



Stack after call



Errors

$8204	htWindowClosed	Window closed
$8207	htFailed	'Name' String not found
$8208	htNotInit	HTMLInit has not been called
$82FF	htBadCall	Incorrect parameters supplied

C

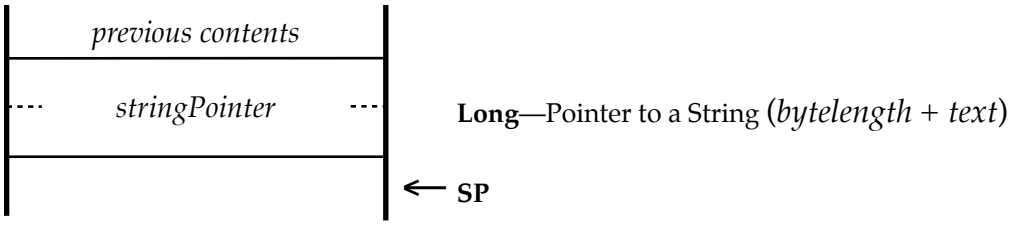
```
extern pascal Long HTMLGoToName (stringPointer);
Long     position, stringPointer
```

$1282 HTMLFindString

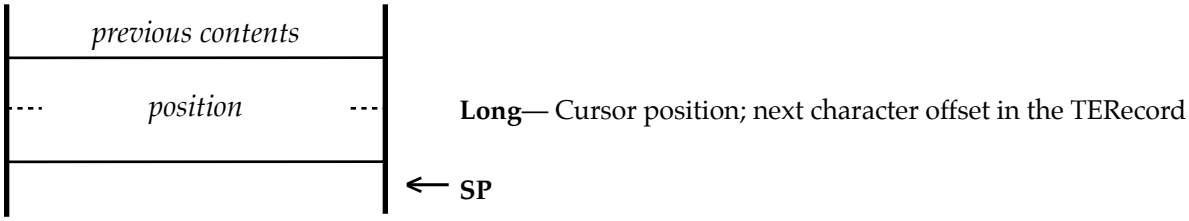
Jumps to the next occurrence of *'String'*, searching from the current cursor position.

Parameters

Stack before call



Stack after call



Errors

$8204	htWindowClosed	Window closed
$8207	htFailed	'Name' String not found
$8208	htNotInit	HTMLInit has not been called
$82FF	htBadCall	Incorrect parameters supplied

C

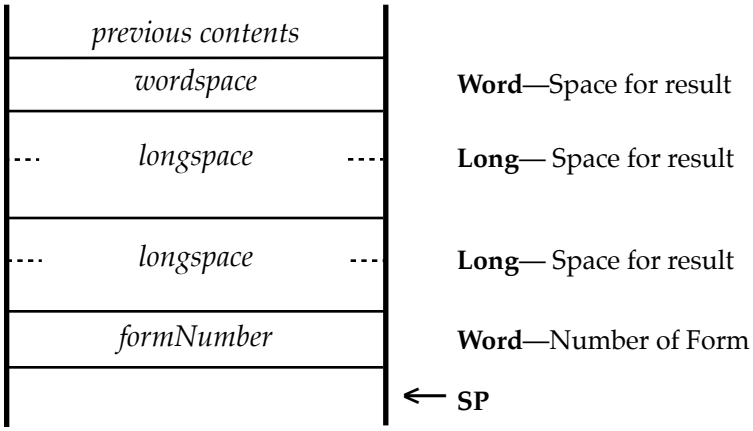
```
extern pascal Long HTMLFindString (position);
Long     position, stringPointer
```

$1382 HTMLGetForm

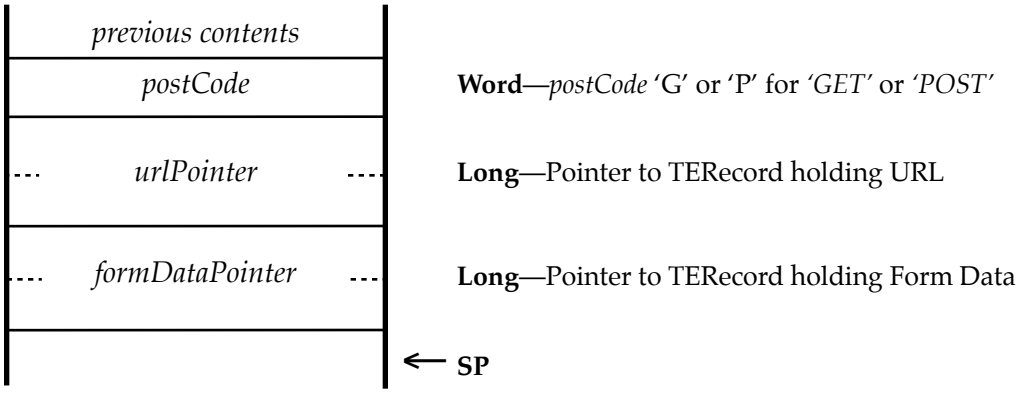
Returns Form data.

Parameters

Stack before call



Stack after call



Errors

$8204	htWindowClosed	Window closed
$8207	htFailed	Form does not exist
$8208	htNotInit	HTMLInit has not been called
$82FF	htBadCall	Incorrect parameters supplied

C

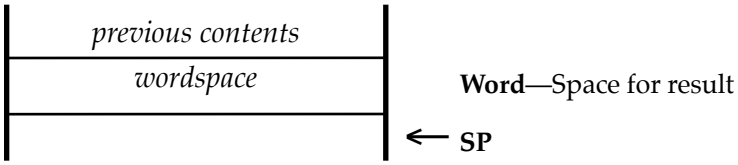
```
extern pascal Word Long Long HTMLGetForm (formNumber);
Long      urlPointer, formDataPointer
Word      formNumber, postCode
```

$1482 HTMLCheckForForm

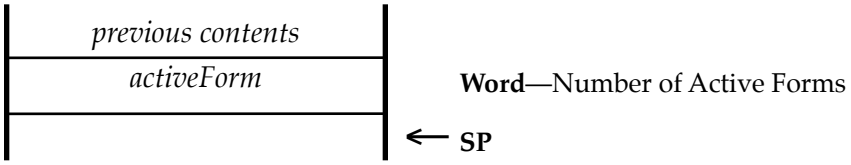
Returns the number of Forms within the page.

Parameters

Stack before call



Stack after call



Errors

None

C

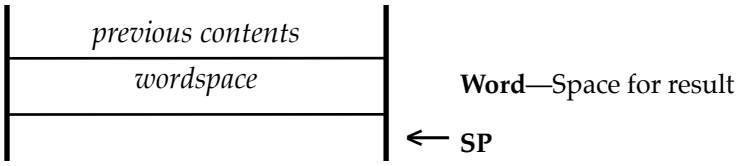
```
extern pascal Word HTMLCheckForForm ();
Word      activeForm
```

$1582 HTMLGetError

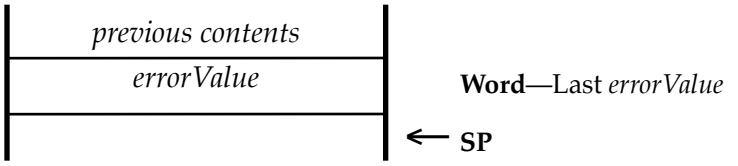
Returns the last *errorValue*.

Parameters

Stack before call



Stack after call



Errors

None

C

```
extern pascal Word HTMLGetError ();
Word      errorValue
```

$8201	htNoFonts	Required fonts not installed
$8202	htBadHandle	Bad Handle value supplied
$8203	htWindowOpen	Display window already open
$8204	htWindowClosed	Display window already closed
$8205	htBadParms	Bad parameters were passed
$8206	htNoData	Data was not present when expected
$8207	htFailed	The call failed to execute
$8208	htNotInit	HTMLInit has not been called
$82FE	htNoTools	Required Tools not installed
$82FF	htBadCall	The call failed through bad Data

Links

To obtain the HTML Tool Set, and any of my other software:

http://homepage.mac.com/speccie

To subscribe to the Juiced.GS magazine:

http://juiced.gs/

To read all about the KFest conference:

http://www.kansasfest.org/

To shop for Apple II software:

http://www.syndicomm.com/