247
HTP

The HTP (hypertext procedures) and HTF (hypertext functions) packages generate HTML tags. For example, the HTP.ANCHOR procedure generates the HTML anchor tag, <A>.

This chapter contains the following topics:

> **See Also:**
>
> For more information about implementation of this package:
>
> - *Oracle Fusion Middleware Administrator's Guide for Oracle HTTP Server*
> - *Oracle Fusion Middleware User's Guide for mod_plsql*

HTP Operational Notes

Certain operational notes apply to the HTP package.

For every HTP procedure that generates one or more HTML tags, there is a corresponding HTF function with identical parameters with the following exception:

- The PRINTS Procedure and the PS Procedure do not have HTF function equivalents. Use the ESCAPE_SC Function or the ESCAPE_URL Function if you need a string conversion function. Note that while there is a ESCAPE_SC Procedure that performs the same operation as the PRINTS Procedure and the PS Procedure, there is no procedural equivalent for the ESCAPE_URL Function.
- The FORMAT_CELL Function does not have an HTP equivalent. The function formats column values inside an HTML table using TABLEDATA Function which does have an HTP equivalent in the TABLEDATA Procedure. The advantage of this using the FORMAT_CELL Function is that it allows for better control over the HTML tables.

The function versions do not directly generate output in your Web page. Instead, they pass their output as return values to the statements that invoked them. Use these functions when you need to nest calls. To print the output of HTF functions, call the functions from within the HTP.PRINT procedure. It then prints its parameters to the generated Web page.

HTP Rules and Limits

If you use values of the LONG datatype in procedures such as HTP.PRINT, HTP.PRN, HTP.PRINTS, HTP.PA or OWA_UTIL.CELLSPRINT, only the first 32 K of the LONG data is used. The LONG data is bound to a VARCHAR2 datatype in the procedure.

HTP Examples

These commands generate a simple HTML document.

```
CREATE OR REPLACE PROCEDURE hello AS
BEGIN
    HTP.HTMLOPEN;            -- generates <HTML>
    HTP.HEADOPEN;            -- generates <HEAD>
    HTP.TITLE('Hello');      -- generates <TITLE>Hello</TITLE>
    HTP.HEADCLOSE;           -- generates </HEAD>
    HTP.BODYOPEN;            -- generates <BODY>
    HTP.HEADER(1, 'Hello'); -- generates <H1>Hello</H1>
    HTP.BODYCLOSE;           -- generates </BODY>
    HTP.HTMLCLOSE;           -- generates </HTML>
END;
```

HTP Summary of Tags

HTP tags can be grouped by type, to make understanding them easier.

HTML, HEAD, and BODY Tags

HTMLOPEN Procedure, HTMLCLOSE Procedure - generate <HTML> and </HTML>

HEADOPEN Procedure, HEADCLOSE Procedure - generate <HEAD> and </HEAD>

BODYOPEN Procedure, BODYCLOSE Procedure - generate <BODY> and </BODY>

Comment Tag

COMMENT Procedure - generates <!-- and -->

Tags in the <HEAD> Area

BASE Procedure - generates <BASE>

LINKREL Procedure - generates <LINK> with the REL attribute

LINKREV Procedure - generates <LINK> with the REV attribute

TITLE Procedure - generates <TITLE>

META Procedure - generates <META>

SCRIPT Procedure - generates <SCRIPT>

STYLE Procedure - generates <STYLE>

ISINDEX Procedure - generates <ISINDEX>

Applet Tags

APPLETOPEN Procedure, APPLETCLOSE Procedure - generate `<APPLET>` and `</APPLET>`

PARAM Procedure - generates `<PARAM>`

List Tags

OLISTOPEN Procedure, OLISTCLOSE Procedure - generate `<OL>` and `</OL>`

ULISTOPEN Procedure, ULISTCLOSE Procedure - generate `<UL>` and `</UL>`

DLISTOPEN Procedure, DLISTCLOSE Procedure- generate `<DL>` and `</DL>`

DLISTTERM Procedure - generates `<DT>`

DLISTDEF Procedure - generates `<DD>`

DIRLISTOPEN Procedure, DIRLISTCLOSE Procedure - generate `<DIR>` and `</DIR>`

LISTHEADER Procedure - generates `<LH>`

LISTINGOPEN Procedure, LISTINGCLOSE Procedure - generate `<LISTING>` and `</LISTING>`

MENULISTOPEN Procedure - generate `<MENU>` and `</MENU>`

LISTITEM Procedure - generates `<LI>`

Form Tags

FORMOPEN Procedure, FORMCLOSE Procedure - generate `<FORM>` and `</FORM>`

FORMCHECKBOX Procedure - generates `<INPUT TYPE="CHECKBOX">`

FORMHIDDEN Procedure - generates `<INPUT TYPE="HIDDEN">`

FORMIMAGE Procedure - generates `<INPUT TYPE="IMAGE">`

FORMPASSWORD Procedure - generates `<INPUT TYPE="PASSWORD">`

FORMRADIO Procedure - generates `<INPUT TYPE="RADIO">`

FORMSELECTOPEN Procedure, FORMSELECTCLOSE Procedure - generate `<SELECT>` and `</SELECT>`

FORMSELECTOPTION Procedure - generates `<OPTION>`

FORMTEXT Procedure - generates `<INPUT TYPE="TEXT">`

FORMTEXTAREA Procedure - generate `<TEXTAREA>`

FORMTEXTAREAOPEN Procedure, FORMTEXTAREACLOSE Procedure - generate `<TEXTAREA>` and `</TEXTAREA>`

FORMRESET Procedure - generates `<INPUT TYPE="RESET">`

FORMSUBMIT Procedure - generates `<INPUT TYPE="SUBMIT">`

Table Tags

TABLEOPEN Procedure, TABLECLOSE Procedure - generate `<TABLE>` and `</TABLE>`

TABLECAPTION Procedure - generates `<CAPTION>`

TABLEROWOPEN Procedure, TABLEROWCLOSE Procedure - generate `<TR>` and `</TR>`

TABLEHEADER Procedure - generates `<TH>`

TABLEDATA Procedure - generates `<TD>`

IMG, HR, and A Tags

HR Procedure, LINE Procedure - generate `<HR>`

IMG Procedure, IMG2 Procedure - generate `<IMG>`

ANCHOR Procedure, ANCHOR2 Procedure - generate `<A>`

MAPOPEN Procedure, MAPCLOSE Procedure - generate `<MAP>` and `</MAP>`

Paragraph Formatting Tags

HEADER Procedure - generates heading tags (`<H1>` to `<H6>`)

PARA Procedure, PARAGRAPH Procedure - generate `<P>`

PRN Procedures, PRINT Procedures - generate any text that is passed in

PRINTS Procedure, PS Procedure - generate any text that is passed in; special characters in HTML are escaped

PREOPEN Procedure, PRECLOSE Procedure - generate `<PRE>` and `</PRE>`

BLOCKQUOTEOPEN Procedure, BLOCKQUOTECLOSE Procedure - generate `<BLOCKQUOTE>` and `</BLOCKQUOTE>`

DIV Procedure - generates `<DIV>`

NL Procedure, BR Procedure - generate `<BR>`

NOBR Procedure - generates `<NOBR>`

WBR Procedure - generates `<WBR>`

PLAINTEXT Procedure - generates `<PLAINTEXT>`

ADDRESS Procedure - generates `<ADDRESS>`

MAILTO Procedure - generates `<A>` with the `MAILTO` attribute

AREA Procedure - generates `<AREA>`

BGSOUND Procedure - generates `<BGSOUND>`

Character Formatting Tags

BASEFONT Procedure - generates `<BASEFONT>`

BIG Procedure - generates `<BIG>`

BOLD Procedure - generates `<B>`

CENTER Procedure - generates `<CENTER>` and `</CENTER>`

CENTEROPEN Procedure, CENTERCLOSE Procedure - generate `<CENTER>` and `</CENTER>`

CITE Procedure - generates <CITE>

CODE Procedure - generates <CODE>

DFN Procedure - generates <DFN>

EM Procedure, EMPHASIS Procedure - generate <EM>

FONTOPEN Procedure, FONTCLOSE Procedure - generate <FONT> and </FONT>

ITALIC Procedure - generates <I>

KBD Procedure, KEYBOARD Procedure - generate <KBD> and </KBD>

S Procedure - generates <S>

SAMPLE Procedure - generates <SAMP>

SMALL Procedure - generates <SMALL>

STRIKE Procedure - generates <STRIKE>

STRONG Procedure - generates <STRONG>

SUB Procedure - generates <SUB>

SUP Procedure - generates <SUP>

TELETYPE Procedure - generates <TT>

UNDERLINE Procedure - generates <U>

VARIABLE Procedure - generates <VAR>

Frame Tags

FRAME Procedure - generates <FRAME>

FRAMESETOPEN Procedure, FRAMESETCLOSE Procedure - generate <FRAMESET> and </FRAMESET>

NOFRAMESOPEN Procedure, NOFRAMESCLOSE Procedure - generate <NOFRAMES> and </NOFRAMES>

Summary of HTP Subprograms

This table lists the HTP subprograms and briefly describes them.

Table 247-1 HTP Package Subprograms

Subprogram	Description
ADDRESS Procedure	Generates the <ADDRESS> and </ADDRESS> tags which specify the address, author and signature of a document
ANCHOR Procedure	Generates the <A> and </A> tags which specify the source or destination of a hypertext link
ANCHOR2 Procedure	Generates the <A> and </A> tags which specify the source or destination of a hypertext link
APPLETCLOSE Procedure	Closes the applet invocation with the </APPLET> tag

Table 247-1 (Cont.) HTP Package Subprograms

Subprogram	Description
APPLETOPEN Procedure	Generates the `<APPLET>` tag which begins the invocation of a Java applet
AREA Procedure	Generates the `<AREA>` tag, which defines a client-side image map
BASE Procedure	Generates the `<BASE>` tag which records the URL of the document
BASEFONT Procedure	Generates the `<BASEFONT>` tag which specifies the base font size for a Web page
BGSOUND Procedure	Generates the `<BGSOUND>` tag which includes audio for a Web page
BIG Procedure	Generates the `<BIG>` and `</BIG>` tags which direct the browser to render the text in a bigger font
BLOCKQUOTECLOSE Procedure	Generates the `</BLOCKQUOTE>` tag which mark the end of a section of quoted text
BLOCKQUOTEOPEN Procedure	Generates the `<BLOCKQUOTE>` tag, which marks the beginning of a section of quoted text
BODYCLOSE Procedure	Generates the `</BODY>` tag which marks the end of a body section of an HTML document
BODYOPEN Procedure	Generates the `<BODY>` tag which marks the beginning of the body section of an HTML document
BOLD Procedure	Generates the `<B>` and `</B>` tags which direct the browser to display the text in boldface
BR Procedure	Generates the `<BR>` tag which begins a new line of text
CENTER Procedure	Generates the `<CENTER>` and `</CENTER>` tags which center a section of text within a Web page
CENTERCLOSE Procedure	Generates the `</CENTER>` tag which marks the end of a section of text to center
CENTEROPEN Procedure	Generates the `<CENTER>` tag which mark the beginning of a section of text to center
CITE Procedure	Generates the `<CITE>` and `</CITE>` tags which direct the browser to render the text as a citation
CODE Procedure	Generates the `<CODE>` and `</CODE>` tags which direct the browser to render the text in monospace font or however "code" is defined stylistically
COMMENT Procedure	`Generates procedure generates the comment tags <!-- ctext -->`
DFN Procedure	Generates the `<DFN>` and `</DFN>` tags which direct the browser to mark the text as italics or however "definition" is defined stylistically
DIRLISTCLOSE Procedure	Generates the `</DIR>` tag which ends a directory list section
DIRLISTOPEN Procedure	Generates the `<DIR>` which starts a directory list section
DIV Procedure	Generates the `<DIV>` tag which creates document divisions
DLISTCLOSE Procedure	Generates the `</DL>` tag which ends a definition list
DLISTDEF Procedure	Generates the `<DD>` tag, which inserts definitions of terms
DLISTOPEN Procedure	Generates the `<DL>` tag which starts a definition list
DLISTTERM Procedure	Generates the `<DT>` tag which defines a term in a definition list `<DL>`

Table 247-1 (Cont.) HTP Package Subprograms

Subprogram	Description
EM Procedure	Generates the `<EM>` and `</EM>` tags, which define text to be emphasized
EMPHASIS Procedure	Generates the `<EM>` and `</EM>` tags, which define text to be emphasized
ESCAPE_SC Procedure	Replaces characters that have special meaning in HTML with their escape sequences
FONTCLOSE Procedure	Generates the `</FONT>` tag which marks the end of a section of text with the specified font characteristics
FONTOPEN Procedure	Generates the `<FONT>` which marks the beginning of section of text with the specified font characteristics
FORMCHECKBOX Procedure	Generates the `<INPUT>` tag with `TYPE="checkbox"` which inserts a checkbox element in a form
FORMCLOSE Procedure	Generates the `</FORM>` tag which marks the end of a form section in an HTML document
FORMOPEN Procedure	Generates the `<FORM>` tag which marks the beginning of a form section in an HTML document
FORMFILE Procedure	Generates the `<INPUT>` tag with `TYPE="file"` which inserts a file form element, and is used for file uploading for a given page
FORMHIDDEN Procedure	Generates the `<INPUT>` tag with `TYPE="hidden"`which inserts a hidden form element
FORMIMAGE Procedure	Generates the `<INPUT>` tag with `TYPE="image"` which creates an image field that the user clicks to submit the form immediately
FORMPASSWORD Procedure	Generates the `<INPUT>` tag with `TYPE="password"` which creates a single-line text entry field
FORMRADIO Procedure	Generates the `<INPUT>` tag with `TYPE="radio"`, which creates a radio button on the HTML form
FORMRESET Procedure	`Generates the <INPUT> tag with TYPE="reset"` which creates a button that, when selected, resets the form fields to their initial values
FORMSELECTCLOSE Procedure	Generates the `</SELECT>` tag which marks the end of a Select form element
FORMSELECTOPEN Procedure	Generates the `</SELECT>` tag which marks the beginning of a Select form element
FORMSELECTOPTION Procedure	Generates the `<OPTION>` tag which represents one choice in a Select element
FORMSUBMIT Procedure	Generates the `<INPUT>` tag with `TYPE="submit"` which creates a button that, when clicked, submits the form
FORMTEXT Procedure	Generates the `<INPUT>` tag with `TYPE="text"`, which creates a field for a single line of text
FORMTEXTAREA Procedure	Generates the `<TEXTAREA>` tag, which creates a text field that has no predefined text in the text area
FORMTEXTAREA2 Procedure	Generates the `<TEXTAREA>` tag, which creates a text field that has no predefined text in the text area with the ability to specify a wrap style

Table 247-1 (Cont.) HTP Package Subprograms

Subprogram	Description
FORMTEXTAREACLOSE Procedure	Generates the `</TEXTAREA>` tag which ends a text area form element
FORMTEXTAREAOPEN Procedure	Generates the `<TEXTAREA>` which marks the beginning of a text area form element
FORMTEXTAREAOPEN2 Procedure	Generates the `<TEXTAREA>` which marks the beginning of a text area form element with the ability to specify a wrap style
FRAME Procedure	Generates the `<FRAME>` tag which begins the characteristics of a frame created by a `<FRAMESET>` tag
FRAMESETCLOSE Procedure	Generates the `</FRAMESET>` tag which ends a frameset section
FRAMESETOPEN Procedure	Generates the `</FRAMESET>` tag which begins a frameset section
HEADCLOSE Procedure	Generates the `</HEAD>` tag which marks the end of an HTML document head section
HEADER Procedure	Generates opening heading tags `(<H1>` to `<H6>)` and their corresponding closing tags (`</H1>` to `</H6>)`
HEADOPEN Procedure	Generates the `<HEAD>` tag which marks the beginning of the HTML document head section
HR Procedure	Generates the `<HR>` tag, which generates a line in the HTML document
HTMLCLOSE Procedure	Generates the `</HTML>` tag which marks the end of an HTML document
HTMLOPEN Procedure	Generates the `<HTML>` tag which marks the beginning of an HTML document
IMG Procedure	Generates the `<IMG>` tag which directs the browser to load an image onto the HTML page
IMG2 Procedure	`Generates the <IMG>` tag which directs the browser to load an image onto the HTML page with the option of specifying values for the `USEMAP` attribute
ISINDEX Procedure	Creates a single entry field with a prompting text, such as "*enter value,*" then sends that value to the URL of the page or program
ITALIC Procedure	Generates the `<I>` and `</I>` tags which direct the browser to render the text in italics
KBD Procedure	Generates the `<KBD>` and `</KBD>` tags which direct the browser to render the text in monospace font
KEYBOARD Procedure	Generates the <KBD> and </KBD> tags, which direct the browser to render the text in monospace font
LINE Procedure	Generates the `<HR>` tag, which generates a line in the HTML document
LINKREL Procedure	Generates the `<LINK>` tag with the `REL` attribute which delineates the relationship described by the hypertext link from the anchor to the target
LINKREV Procedure	Generates the `<LINK>` tag with the `REV` attribute which delineates the relationship described by the hypertext link from the target to the anchor
LISTHEADER Procedure	Generates the `<LH>` and `</LH>` tags which print an HTML tag at the beginning of the list

Table 247-1 (Cont.) HTP Package Subprograms

Subprogram	Description
LISTINGCLOSE Procedure	Generates the `</LISTING>` tags which marks the end of a section of fixed-width text in the body of an HTML page
LISTINGOPEN Procedure	Generates the `<LISTING>` tag which marks the beginning of a section of fixed-width text in the body of an HTML page
LISTITEM Procedure	Generates the `<LI>` tag, which indicates a list item
MAILTO Procedure	Generates the `<A>` tag with the `HREF` set to 'mailto' prepended to the mail address argument
MAPCLOSE Procedure	Generates the `</MAP>` tag which marks the end of a set of regions in a client-side image map
MAPOPEN Procedure	Generates the `<MAP>` tag which mark the beginning of a set of regions in a client-side image map
MENULISTCLOSE Procedure	Generates the `</MENU>` tag which ends a list that presents one line for each item
MENULISTOPEN Procedure	Generates the `<MENU>` tag which begins a list that presents one line for each item
META Procedure	Generates the `<META>` tag, which embeds meta-information about the document and also specifies values for HTTP headers
NL Procedure	Generates the `<BR>` tag which begins a new line of text
NOBR Procedure	Generates the `<NOBR>` and `</NOBR>` tags which turn off line-breaking in a section of text
NOFRAMESCLOSE Procedure	Generates the `</NOFRAMES>` tag which marks the end of a no-frames section
NOFRAMESOPEN Procedure	Generates the `<NOFRAMES>` tag which mark the beginning of a no-frames section
OLISTCLOSE Procedure	Generates the `</OL>` tag which defines the end of an ordered list
OLISTOPEN Procedure	Generates the `<OL>` tag which marks the beginning of an ordered list
PARA Procedure	Generates the `<P>` tag which indicates that the text that comes after the tag is to be formatted as a paragraph
PARAGRAPH Procedure	Adds attributes to the `<P>` tag
PARAM Procedure	Generates the `<PARAM>` tag which specifies parameter values for Java applets
PLAINTEXT Procedure	Generates the `<PLAINTEXT>` and `</PLAINTEXT>` tags which direct the browser to render the text they surround in fixed-width type
PRECLOSE Procedure	Generates the `</PRE>` tag which marks the end of a section of preformatted text in the body of the HTML page
PREOPEN Procedure	Generates the `<PRE>` tag which marks the beginning of a section of preformatted text in the body of the HTML page
PRINT Procedures	Generates the specified parameter as a string terminated with the `\n` newline character
PRINTS Procedure	Generates a string and replaces the following characters with the corresponding escape sequence
PRN Procedures	Generates the specified parameter as a string

Table 247-1 (Cont.) HTP Package Subprograms

Subprogram	Description
PS Procedure	Generates a string and replaces the following characters with the corresponding escape sequence.
S Procedure	Generates the `<S>` and `</S>` tags which direct the browser to render the text they surround in strikethrough type
SAMPLE Procedure	Generates the `<SAMP>` and `</SAMP>` tags which direct the browser to render the text they surround in monospace font or however "sample" is defined stylistically
SCRIPT Procedure	Generates the `<SCRIPT>` and `</SCRIPT>` tags which contain a script written in languages such as JavaScript and VBscript
SMALL Procedure	Generates the `<SMALL>` and `</SMALL>` tags, which direct the browser to render the text they surround using a small font
STRIKE Procedure	Generates the `<STRIKE>` and `</STRIKE>` tags which direct the browser to render the text they surround in strikethrough type
STRONG Procedure	Generates the `<STRONG>` and `</STRONG>` tags which direct the browser to render the text they surround in bold or however "strong" is defined stylistically
STYLE Procedure	Generates the `<STYLE>` and `</STYLE>` tags which include a style sheet in a Web page
SUB Procedure	Generates the `<SUB>` and `</SUB>` tags which direct the browser to render the text they surround as subscript
SUP Procedure	Generates the `<SUP>` and `</SUP>` tags which direct the browser to render the text they surround as superscript
TABLECAPTION Procedure	Generates the `<CAPTION>` and `</CAPTION>` tags which place a caption in an HTML table
TABLECLOSE Procedure	Generates the `</TABLE>` tag which marks the end of an HTML table
TABLEDATA Procedure	Generates the `<TD>` and `</TD>` tags which insert data into a cell of an HTML table
TABLEHEADER Procedure	Generates the `<TH>` and `</TH>` tags which insert a header cell in an HTML table.
TABLEOPEN Procedure	Generates the <TABLE> tag which marks the beginning of an HTML table
TABLEROWCLOSE Procedure	Generates the `</TR>` tag which marks the end of a new row in an HTML table
TABLEROWOPEN Procedure	Generates the `<TR>` tag which marks the beginning of a new row in an HTML table
TELETYPE Procedure	Generates the `<TT>` and `</TT>` tags which direct the browser to render the text they surround in a fixed width typewriter font, for example, the courier font
TITLE Procedure	Generates the `<TITLE>` and `</TITLE>` tags which specify the text to display in the titlebar of the browser window
ULISTCLOSE Procedure	Generates the `</UL>` tag which marks the end of an unordered list
ULISTOPEN Procedure	Generates the `<UL>` tag which marks the beginning of an unordered list

Table 247-1 (Cont.) HTP Package Subprograms

Subprogram	Description
UNDERLINE Procedure	Generates the `<U>` and `</U>` tags, which direct the browser to render the text they surround with an underline
VARIABLE Procedure	Generates the `<VAR>` and `</VAR>` tags which direct the browser to render the text they surround in italics or however "variable" is defined stylistically.
WBR Procedure	Generates the `<WBR>` tag, which inserts a soft line break within a section of `NOBR` text

ADDRESS Procedure

This procedure generates the `<ADDRESS>` and `</ADDRESS>` tags which specify the address, author and signature of a document.

Syntax

```
HTP.ADDRESS (
   cvalue         IN       VARCHAR2
   cnowrap        IN       VARCHAR2    DEFAULT NULL
   cclear         IN       VARCHAR2    DEFAULT NULL
   cattributes    IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-2 ADDRESS Procedure Parameters

Parameter	Description
`cvalue`	The string that goes between the `<ADDRESS>` and `</ADDRESS>` tags.
`cnowrap`	If the value for this parameter is not `NULL`, the `NOWRAP` attribute is included in the tag
`cclear`	The value for the `CLEAR` attribute.
`cattributes`	The other attributes to be included as-is in the tag

Examples

This procedure generates

```
<ADDRESS CLEAR="cclear" NOWRAP cattributes>cvalue</ADDRESS>
```

ANCHOR Procedure

Like the ANCHOR2 procedure, this procedure generates the `<A>` and `</A>` HTML tags which specify the source or destination of a hypertext link.

The difference between these subprograms is that the ANCHOR2 Procedure provides a target and therefore can be used for a frame.

Syntax

```
HTP.ANCHOR (
   curl            IN       VARCHAR2,
   ctext           IN       VARCHAR2,
   cname           IN       VARCHAR2   DEFAULT NULL,
   cattributes     IN       VARCHAR2   DEFAULT NULL);
```

Parameters

Table 247-3   ANCHOR Procedure Parameters

Parameter	Description
`curl`	The value for the `HREF` attribute.
`ctext`	The string that goes between the `<A>` and `</A>` tags.
`cname`	The value for the `NAME` attribute.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<A HREF="curl" NAME="cname" cattributes>ctext</A>
```

Usage Notes

This tag accepts several attributes, but either `HREF` or `NAME` is required. `HREF` specifies to where to link. `NAME` allows this tag to be a target of a hypertext link.

ANCHOR2 Procedure

Like the ANCHOR procedure, this procedure generates the `<A>` and `</A>` HTML tags which specify the source or destination of a hypertext link.

The difference between this procedure and the ANCHOR procedure is that this procedure provides a target and therefore can be used for a frame.

Syntax

```
HTP.ANCHOR2 (
   curl            IN       VARCHAR2,
   ctext           IN       VARCHAR2,
   cname           IN       VARCHAR2   DEFAULT NULL,
   ctarget         in       varchar2   DEFAULT NULL,
   cattributes     IN       VARCHAR2   DEFAULT NULL);
```

Parameters

Table 247-4   ANCHOR2 Procedure Parameters

Parameter	Description
`curl`	The value for the `HREF` attribute.
`ctext`	The string that goes between the `<A>` and `</A>` tags.

Table 247-4 (Cont.) ANCHOR2 Procedure Parameters

Parameter	Description
`cname`	The value for the `NAME` attribute
`ctarget`	The value for the `TARGET` attribute.
`cattributes`	The other attributes to be included as-is in the tag

Examples

This procedure generates

```
<A HREF="curl" NAME="cname" TARGET = "ctarget" cattributes>ctext</A>
```

APPLETCLOSE Procedure

This procedure closes the applet invocation with the `</APPLET>` tag.

You must first invoke the a Java applet using APPLETOPEN Procedure.

Syntax

```
HTP.APPLETCLOSE;
```

APPLETOPEN Procedure

This procedure generates the `<APPLET>` tag which begins the invocation of a Java applet.

You close the applet invocation with APPLETCLOSE Procedure which generates the </APPLET> tag.

Syntax

```
HTP.APPLETOPEN (
   ccode          IN       VARCHAR2,
   cheight        IN       NUMBER,
   cwidth         IN       NUMBER,
   cattributes    IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-5 APPLETOPEN Procedure Parameters

Parameter	Description
`ccode`	The the value for the `CODE` attribute which specifies the name of the applet class.
`cheight`	The value for the `HEIGHT` attribute.
`cwidth`	The value for the `WIDTH` attribute.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<APPLET CODE=ccode HEIGHT=cheight WIDTH=cwidth cattributes>
```

so that, for example,

```
HTP.appletopen('testclass.class', 100, 200, 'CODEBASE="/ows-applets"')
```

generates

```
<APPLET CODE="testclass.class" height=100 width=200 CODEBASE="/ows-applets">
```

Usage Notes

- Specify parameters to the Java applet using the PARAM Procedure.
- Use the `cattributes` parameter to specify the `CODEBASE` attribute since the PL/SQL cartridge does not know where to find the class files. The `CODEBASE` attribute specifies the virtual path containing the class files.

AREA Procedure

This procedure generates the `<AREA>` tag, which defines a client-side image map. The `<AREA>` tag defines areas within the image and destinations for the areas.

Syntax

```
HTP.AREA (
   ccoords        IN       VARCHAR2
   cshape         IN       VARCHAR2   DEFAULT NULL,
   chref          IN       VARCHAR2   DEFAULT NULL,
   cnohref        IN       VARCHAR2   DEFAULT NULL,
   ctarget        IN       VARCHAR2   DEFAULT NULL,
   cattributes    IN       VARCHAR2   DEFAULT NULL);
```

Parameters

Table 247-6 AREA Procedure Parameters

Parameter	Description
`ccords`	The the value for the `COORDS` attribute.
`cshape`	The value for the `SHAPE` attribute.
`chref`	The value for the `HREF` attribute.
`cnohref`	If the value for this parameter is not `NULL`, the `NOHREF` attribute is added to the tag.
`ctarget`	The value for the `TARGET` attribute.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<AREA COORDS="ccoords" SHAPE="cshape" HREF="chref" NOHREF TARGET="ctarget" cattributes>
```

BASE Procedure

This procedure generates the `<BASE>` tag which records the URL of the document.

Syntax

```
HTP.BASE (
   ctarget         IN       VARCHAR2   DEFAULT NULL,
   cattributes     IN       VARCHAR2   DEFAULT NULL);
```

Parameters

Table 247-7   BASE Procedure Parameters

Parameter	Description
`ctarget`	The value for the `TARGET` attribute which establishes a window name to which all links in this document are targeted.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<BASE HREF="<current URL>" TARGET="ctarget" cattributes>
```

BASEFONT Procedure

This procedure generates the `<BASEFONT>` tag which specifies the base font size for a Web page.

Syntax

```
HTP.BASEFONT (
   nsize     IN     INTEGER);
```

Parameters

Table 247-8   BASEFONT Procedure Parameters

Parameter	Description
`nsize`	The value for the `SIZE` attribute.

Examples

This procedure generates

```
<BASEFONT SIZE="nsize">
```

BGSOUND Procedure

This procedure generates the `<BGSOUND>` tag which includes audio for a Web page.

Syntax

```
HTP.BGSOUND (
   csrc           IN       VARCHAR2,
   cloop          IN       VARCHAR2   DEFAULT NULL,
   cattributes    IN       VARCHAR2   DEFAULT NULL);
```

Parameters

Table 247-9 BGSOUND Procedure Parameters

Parameter	Description
`csrc`	The value for the `SRC` attribute.
`cloop`	The value for the `LOOP` attribute.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<BGSOUND SRC="csrc" LOOP="cloop" cattributes>
```

BIG Procedure

This procedure generates the `<BIG>` and `</BIG>` tags which direct the browser to render the text in a bigger font.

Syntax

```
HTP.BIG (
   ctext          IN       VARCHAR2,
   cattributes    IN       VARCHAR2   DEFAULT NULL);
```

Parameters

Table 247-10 BIG Procedure Parameters

Parameter	Description
`ctext`	The the text that goes between the tags.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<BIG cattributes>ctext</BIG>
```

BLOCKQUOTECLOSE Procedure

This procedure generates the `</BLOCKQUOTE>` tag which mark the end of a section of quoted text.

You mark the beginning of a section of text by means of the BLOCKQUOTEOPEN Procedure.

Syntax

```
HTP.BLOCKQUOTECLOSE;
```

Examples

This procedure generates

```
</BLOCKQUOTE>
```

BLOCKQUOTEOPEN Procedure

This procedure generates the `<BLOCKQUOTE>` tag, which marks the beginning of a section of quoted text.

You mark the end of a section of text by means of the BLOCKQUOTECLOSE Procedure.

Syntax

```
HTP.BLOCKQUOTEOPEN (
   cnowrap        IN       VARCHAR2    DEFAULT NULL,
   cclear         IN       VARCHAR2    DEFAULT NULL,
   cattributes    IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-11 BLOCKQUOTEOPEN Procedure Parameters

Parameter	Description
`cnowrap`	If the value for this parameter is not `NULL`, the `NOWRAP` attribute is added to the tag.
`cclear`	The value for the `CLEAR` attribute.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<BLOCKQUOTE CLEAR="cclear" NOWRAP cattributes>
```

BODYCLOSE Procedure

This procedure generates the `</BODY>` tag which marks the end of a body section of an HTML document.

You mark the beginning of a body section by means of the BODYOPEN Procedure.

Syntax

```
HTP.BODYCLOSE;
```

Examples

This procedure generates

```
</BODY>
```

BODYOPEN Procedure

This procedure generates the `<BODY>` tag which marks the beginning of the body section of an HTML document.

You mark the end of a body section by means of the BODYCLOSE Procedure.

Syntax

```
HTP.BODYOPEN (
   cbackground    IN       VARCHAR2   DEFAULT NULL,
   cattributes    IN       VARCHAR2   DEFAULT NULL);
```

Parameters

Table 247-12   BODYOPEN Procedure Parameters

Parameter	Description
`cbackground`	The value for the `BACKGROUND` attribute which specifies a graphic file to use for the background of the document.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<BODY background="cbackground" cattributes>
```

so that

```
HTP.BODYOPEN('/img/background.gif');
```

generates:

```
<BODY background="/img/background.gif">
```

BOLD Procedure

This procedure generates the `<B>` and `</B>` tags which direct the browser to display the text in boldface.

Syntax

```
HTP.BOLD (
   ctext          IN       VARCHAR2,
   cattributes    IN       VARCHAR2   DEFAULT NULL);
```

Parameters

Table 247-13 BOLD Procedure Parameters

Parameter	Description
`ctext`	The text that goes between the tags.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<B cattributes>ctext</B>
```

BR Procedure

This procedure generates the `<BR>` tag which begins a new line of text.

It performs the same operation as the NL Procedure.

Syntax

```
HTP.BR(
   cclear         IN       VARCHAR2   DEFAULT NULL,
   cattributes    IN       VARCHAR2   DEFAULT NULL);
```

Parameters

Table 247-14 BR Procedure Parameters

Parameter	Description
`cclear`	The value for the `CLEAR` attribute.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<BR CLEAR="cclear" cattributes>
```

CENTER Procedure

This procedure generates the `<CENTER>` and `</CENTER>` tags which center a section of text within a Web page.

Syntax

```
HTP.CENTER (
   ctext          IN       VARCHAR2);
```

Parameters

Table 247-15   CENTER Parameters

Parameter	Description
`ctext`	The text that goes between the tags.

Examples

This procedure generates

```
<CENTER>ctext</CENTER>
```

CENTERCLOSE Procedure

This procedure generates the `</CENTER>` tag which marks the end of a section of text to center.

You mark the beginning of a section of text to center with the CENTEROPEN Procedure.

Syntax

```
HTP.CENTERCLOSE;
```

Examples

This procedure generates

```
</CENTER>
```

CENTEROPEN Procedure

This procedure generates the `<CENTER>` tag which mark the beginning of a section of text to center.

You mark the beginning of a of a section of text to center by means of the CENTERCLOSE Procedure.

Syntax

```
HTP.CENTEROPEN;
```

Examples

This procedure generates

```
<CENTER>
```

CITE Procedure

This procedure generates the `<CITE>` and `</CITE>` tags which direct the browser to render the text as a citation.

Syntax

```
HTP.CITE (
   ctext           IN       VARCHAR2,
   cattributes     IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-16 CITE Procedure Parameters

Parameter	Description
`ctext`	The text to render as citation.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<CITE cattributes>ctext</CITE>
```

CODE Procedure

This procedure generates the `<CODE>` and `</CODE>` tags which direct the browser to render the text in monospace font or however "code" is defined stylistically.

Syntax

```
HTP.CODE (
   ctext          IN       VARCHAR2,
   cattributes    IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-17 CODE Procedure Parameters

Parameter	Description
`ctext`	The text to render as code.
`cattributes`	The other attributes to be included as-is in the tag

Examples

This procedure generates

```
<CODE cattributes>ctext</CODE>
```

COMMENT Procedure

This procedure generates the comment tags.

Syntax

```
HTP.COMMENT (
   ctext          IN       VARCHAR2);
```

Parameters

Table 247-18 COMMENT Procedure Parameters

Parameter	Description
`ctext`	The comment.

Examples

This procedure generates

```
<!-- ctext -->
```

DFN Procedure

This procedure generates the `<DFN>` and `</DFN>` tags which direct the browser to mark the text in italics or however "definition" is described stylistically.

Syntax

```
HTP.DFN (
   ctext          IN       VARCHAR2);
```

Parameters

Table 247-19 DFN Procedure Parameters

Parameter	Description
`ctext`	The text to render in italics.

Examples

This procedure generates

```
<DFN>ctext</DFN>
```

DIRLISTCLOSE Procedure

This procedure generates the `</DIR>` tag which ends a directory list section. You start a directory list section with the DIRLISTOPEN Procedure.

Syntax

```
HTP.DIRLISTCLOSE;
```

Usage Notes

A directory list presents a list of items that contains up to 20 characters. Items in this list are typically arranged in columns, 24 characters wide. Insert the <LI> tag directly or invoke the LISTITEM Procedure so that the `<LI>` tag appears directly after the `</DIR>` tag to define the items as a list.

Examples

This procedure generates

```
</DIR>
```

Related Topics

- DIRLISTOPEN Procedure
  This procedure generates the `<DIR>` which starts a directory list section. You end a directory list section with the DIRLISTCLOSE Procedure.

DIRLISTOPEN Procedure

This procedure generates the `<DIR>` which starts a directory list section. You end a directory list section with the DIRLISTCLOSE Procedure.

Syntax

```
HTP.DIRLISTOPEN;
```

Usage Notes

A directory list presents a list of items that contains up to 20 characters. Items in this list are typically arranged in columns, 24 characters wide. Insert the <LI> tag directly or invoke the LISTITEM Procedure so that the `<LI>` tag appears directly after the `</DIR>` tag to define the items as a list.

Examples

This procedure generates

```
<DIR>
```

Related Topics

- DIRLISTCLOSE Procedure
  This procedure generates the `</DIR>` tag which ends a directory list section. You start a directory list section with the DIRLISTOPEN Procedure.

DIV Procedure

This procedure generates the `<DIV>` tag which creates document divisions.

Syntax

```
HTP.DIV (
   calign         IN       VARCHAR2    DEFAULT NULL,
   cattributes    IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-20 DIV Procedure Parameters

Parameter	Description
`calign`	The value for the `ALIGN` attribute.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<DIV ALIGN="calign" cattributes>
```

DLISTCLOSE Procedure

This procedure generates the `</DL>` tag which ends a definition list. You start a definition list by means of the DLISTOPEN Procedure.

Syntax

```
HTP.DLISTCLOSE;
```

Usage Notes

A definition list looks like a glossary: it contains terms and definitions. Terms are inserted using the DLISTTERM Procedure and definitions are inserted using the DLISTDEF Procedure.

Examples

This procedure generates

```
</DL>
```

Related Topics

- DLISTOPEN Procedure
  This procedure generates the `<DL>` tag which starts a definition list. You end a definition list by means of the DLISTCLOSE Procedure.

DLISTDEF Procedure

This procedure generates the `<DD>` tag, which inserts definitions of terms. Use this tag for a definition list `<DL>`. Terms are tagged `<DT>` and definitions are tagged `<DD>`.

Syntax

```
HTP.DLISTDEF (
   ctext          IN       VARCHAR2   DEFAULT NULL,
   cclear         IN       VARCHAR2   DEFAULT NULL,
   cattributes    IN       VARCHAR2   DEFAULT NULL);
```

Parameters

Table 247-21 DLISTDEF Procedure Parameters

Parameter	Description
`ctext`	The definition of the term.
`cclear`	The value for the `CLEAR` attribute.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<DD CLEAR="cclear" cattributes>ctext
```

DLISTOPEN Procedure

This procedure generates the <DL> tag which starts a definition list. You end a definition list by means of the DLISTCLOSE Procedure.

Syntax

```
HTP.DLISTOPEN (
   cclear         IN       VARCHAR2    DEFAULT NULL,
   cattributes    IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-22   DLISTOPEN Procedure Parameters

Parameter	Description
cclear	The value for the CLEAR attribute.
cattributes	The other attributes to be included as-is in the tag.

Usage Notes

A definition list looks like a glossary: it contains terms and definitions. Terms are inserted using the DLISTTERM Procedure and definitions are inserted using the DLISTDEF Procedure.

Examples

This procedure generates

```
<DL CLEAR="cclear" cattributes>
```

Related Topics

- DLISTCLOSE Procedure
  This procedure generates the </DL> tag which ends a definition list. You start a definition list by means of the DLISTOPEN Procedure.

DLISTTERM Procedure

This procedure generates the <DT> tag which defines a term in a definition list <DL>.

Syntax

```
HTP.DLISTTERM (
   ctext          IN       VARCHAR2    DEFAULT NULL,
   cclear         IN       VARCHAR2    DEFAULT NULL,
   cattributes    IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-23   DLISTTERM Procedure Parameters

Parameter	Description
ctext	The term.
cclear	The value for the CLEAR attribute.

Table 247-23 (Cont.) DLISTTERM Procedure Parameters

Parameter	Description
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<DT CLEAR="cclear" cattributes>ctext
```

EM Procedure

This procedure generates the `<EM>` and `</EM>` tags, which define text to be emphasized.

It performs the same task as the EMPHASIS Procedure.

Syntax

```
HTP.EM(
     ctext           IN       VARCHAR2,
     cattributes     IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-24 EM Procedure Parameters

Parameter	Description
`ctext`	The text to emphasize.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<EM cattributes>ctext</EM>
```

EMPHASIS Procedure

This procedure generates the `<EM>` and `</EM>` tags, which define text to be emphasized.

It performs the same task as the EM Procedure.

Syntax

```
HTP.EMPHASIS(
     ctext           IN       VARCHAR2,
     cattributes     IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-25 EMPHASIS Procedure Parameters

Parameter	Description
`ctext`	The text to emphasize.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<EM cattributes>ctext</EM>
```

ESCAPE_SC Procedure

This procedure replaces characters that have special meaning in HTML with their escape sequences.

The following characters are converted:

- `&` to `&amp;`
- " to `&quot:`
- < to `&lt;`
- > to `&gt;`

This procedure performs the same operation as PRINTS Procedures and PS Procedure.

Syntax

```
HTP.ESCAPE_SC(
     ctext          IN       VARCHAR2);
```

Parameters

Table 247-26 ESCAPE_SC Procedure Parameters

Parameter	Description
`ctext`	The text string to convert.

FONTCLOSE Procedure

This procedure generates the`</FONT>` tag which marks the end of a section of text with the specified font characteristics.

You mark the beginning of the section text by means of the FONTOPEN Procedure.

Syntax

```
HTP.FONTCLOSE;
```

Examples

This procedure generates

```
</FONT>
```

FONTOPEN Procedure

This procedure generates the `<FONT>` which marks the beginning of section of text with the specified font characteristics.

You mark the end of the section text by means of the FONTCLOSE Procedure.

Syntax

```
HTP.FONTOPEN(
   ccolor          IN       VARCHAR2    DEFAULT NULL,
   cface           IN       VARCHAR2    DEFAULT NULL,
   csize           IN       VARCHAR2    DEFAULT NULL,
   cattributes     IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-27 FONTOPEN Procedure Parameters

Parameter	Description
`ccolor`	The value for the `COLOR` attribute.
`cface`	The value for the `FACE` attribute
`csize`	The value for the `SIZE` attribute
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<FONT COLOR="ccolor" FACE="cface" SIZE="csize" cattributes>
```

FORMCHECKBOX Procedure

This procedure generates the `<INPUT>` tag with `TYPE="checkbox"` which inserts a checkbox element in a form.

A checkbox element is a button that the user toggles on or off.

Syntax

```
HTP.FORMCHECKBOX(
   cname           IN       VARCHAR2,
   cvalue          IN       VARCHAR2    DEFAULT 'ON',
   cchecked        IN       VARCHAR2    DEFAULT NULL,
   cattributes     IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-28 FORMCHECKBOX Procedure Parameters

Parameter	Description
`cname`	The value for the `NAME` attribute.
`cvalue`	The value for the `VALUE` attribute.
`cchecked`	If the value for this parameter is not `NULL`, the `CHECKED` attribute is added to the tag.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<INPUT TYPE="checkbox" NAME="cname" VALUE="cvalue" CHECKED cattributes>
```

FORMCLOSE Procedure

This procedure generates the `</FORM>` tag which marks the end of a form section in an HTML document.

You mark the beginning of the form section by means of the FORMOPEN Procedure.

Syntax

```
HTP.FORMCLOSE;
```

Examples

This procedure generates

```
</FORM>
```

FORMOPEN Procedure

This procedure generates the `<FORM>` tag which marks the beginning of a form section in an HTML document.

You mark the end of the form section by means of the FORMCLOSE Procedure.

Syntax

```
HTP.FORMOPEN(
   curl           IN       VARCHAR2,
   cmethod        IN       VARCHAR2   DEFAULT 'POST',
   ctarget        IN       VARCHAR2   DEFAULT NULL,
   cenctype       IN       VARCHAR2   DEFAULT NULL,
   cattributes    IN       VARCHAR2   DEFAULT NULL);
```

Parameters

Table 247-29 FORMOPEN Procedure Parameters

Parameter	Description
`curl`	The URL of the WRB or CGI script where the contents of the form is sent. This parameter is required.
`cmethod`	The value for the `METHOD` attribute. The value can be "`GET`" or "`POST`".
`ctarget`	The value for the `TARGET` attribute.
`cenctype`	The value for the `ENCTYPE` attribute.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<FORM ACTION="curl" METHOD="cmethod" TARGET="ctarget" ENCTYPE="cenctype" cattributes>
```

FORMFILE Procedure

This procedure generates the `<INPUT>` tag with `TYPE="file"` which inserts a file form element. This is used for file uploading for a given page.

Syntax

```
HTP.FORMFILE(
   cname           IN       VARCHAR2,
   caccept         IN       VARCHAR2    DEFAULT NULL,
   cattributes     IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-30 FORMFILE Procedure Parameters

Parameter	Description
`cname`	The value for the `NAME` attribute.
`caccept`	A comma-delimited list of MIME types for upload.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<INPUT TYPE="file" NAME="cname" ACCEPT="caccept" cattributes>
```

FORMHIDDEN Procedure

This procedure generates the `<INPUT>` tag with TYPE="`hidden`", which inserts a hidden form element.

This element is not seen by the user. It submits additional values to the script.

Syntax

```
HTP.FORMHIDDEN(
   cname          IN       VARCHAR2,
   cvalue         IN       VARCHAR2    DEFAULT NULL,
   cattributes    IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-31   FORMHIDDEN Procedure Parameters

Parameter	Description
`cname`	The value for the `NAME` attribute.
`cvalue`	The value for the `VALUE` attribute.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<INPUT TYPE="hidden" NAME="cname" VALUE="cvalue" cattributes>
```

FORMIMAGE Procedure

This procedure generates the `<INPUT>` tag with `TYPE="image"` which creates an image field that the user clicks to submit the form immediately.

The coordinates of the selected point are measured in pixels, and returned (along with other contents of the form) in two name/value pairs. The x coordinate is submitted under the name of the field with .x appended, and the y coordinate with .y appended. Any `VALUE` attribute is ignored.

Syntax

```
HTP.FORMIMAGE(
   cname          IN       VARCHAR2,
   csrc           IN       VARCHAR2,
   calign         IN       VARCHAR2    DEFAULT NULL,
   cattributes    IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-32   FORMIMAGE Procedure Parameters

Parameter	Description
`cname`	The value for the `NAME` attribute.
`csrc`	The value for the `SRC` attribute that specifies the image file.
`calign`	The value for the `ALIGN` attribute.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<INPUT TYPE="image" NAME="cname" SRC="csrc" ALIGN="calign" cattributes>
```

FORMPASSWORD Procedure

This procedure generates the `<INPUT>` tag with `TYPE="password"` which creates a single-line text entry field. When the user enters text in the field, each character is represented by one asterisk. This is used for entering passwords.

Syntax

```
HTP.FORMPASSWORD(
   cname          IN       VARCHAR2,
   csize          IN       VARCHAR2,
   cmaxlength     IN       VARCHAR2   DEFAULT NULL,
   cvalue         IN       VARCHAR2   DEFAULT NULL,
   cattributes    IN       VARCHAR2   DEFAULT NULL);
```

Parameters

Table 247-33 FORMPASSWORD Procedure Parameters

Parameter	Description
`cname`	The value for the `NAME` attribute.
`csize`	The value for the `SIZE` attribute.
`cmaxlength`	The value for the `MAXLENGTH` attribute.
`cvalue`	The value for the `VALUE` attribute.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<INPUT TYPE="password" NAME="cname" SIZE="csize" MAXLENGTH="cmaxlength" VALUE="cvalue"
cattributes>
```

FORMRADIO Procedure

This procedure generates the `<INPUT>` tag with `TYPE="radio"`, which creates a radio button on the HTML form. Within a set of radio buttons, the user selects only one. Each radio button in the same set has the same name, but different values. The selected radio button generates a name/value pair.

Syntax

```
HTP.FORMRADIO(
   cname          IN       VARCHAR2,
   cvalue         IN       VARCHAR2,
   cchecked       IN       VARCHAR2   DEFAULT NULL,
   cattributes    IN       VARCHAR2   DEFAULT NULL);
```

Parameters

Table 247-34   FORMRADIO Procedure Parameters

Parameter	Description
`cname`	The value for the `NAME` attribute.
`cvalue`	The value for the `VALUE` attribute.
`cchecked`	If the value for this parameter is not `NULL`, the `CHECKED` attribute is added to the tag.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<INPUT TYPE="radio" NAME="cname" VALUE="cvalue" CHECKED cattributes>
```

FORMRESET Procedure

This procedure generates the `<INPUT>` tag with `TYPE="reset"` which creates a button that, when selected, resets the form fields to their initial values.

Syntax

```
HTP.FORMRESET(
   cvalue         IN       VARCHAR2   DEFAULT 'Reset',
   cattributes    IN       VARCHAR2   DEFAULT NULL);
```

Parameters

Table 247-35   FORMRESET Procedure Parameters

Parameter	Description
`cvalue`	The value for the `VALUE` attribute.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<INPUT TYPE="reset" VALUE="cvalue" cattributes>
```

FORMSELECTCLOSE Procedure

This procedure generates the `</SELECT>` tag which marks the end of a Select form element.

A Select form element is a listbox where the user selects one or more values. You mark the beginning of Select form element by means of the FORMSELECTOPEN Procedure.The values are inserted using FORMSELECTOPTION Procedure.

Syntax

```
HTP.FORMSELECTCLOSE;
```

Examples

This procedure generates

```
</SELECT>
```

as shown under Examples of the FORMSELECTOPEN Procedure.

FORMSELECTOPEN Procedure

This procedure generates the `<SELECT>` tags which creates a Select form element.

A Select form element is a listbox where the user selects one or more values. You mark the end of Select form element by means of the FORMSELECTCLOSE Procedure.The values are inserted using FORMSELECTOPTION Procedure.

Syntax

```
FORMSELECTOPEN(
   cname           IN       VARCHAR2,
   cprompt         IN       VARCHAR2    DEFAULT NULL,
   nsize           IN       INTEGER     DEFAULT NULL,
   cattributes     IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-36 FORMSELECTOPEN Procedure Parameters

Parameter	Description
`cname`	The value for the `NAME` attribute.
`cprompt`	The string preceding the list box.
`nsize`	The value for the `SIZE` attribute.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
cprompt <SELECT NAME="cname" SIZE="nsize" cattributes>
</SELECT>
```

so that

```
HTP.FORMSELECTOPEN('greatest_player';
    'Pick the greatest player:');
HTP.FORMSELECTOPTION('Messier');
HTP.FORMSELECTOPTION('Howe');
HTP.FORMSELECTOPTION('Gretzky');.
HTP.FORMSELECTCLOSE;
```

generates

```
Pick the greatest player:
<SELECT NAME="greatest_player">
<OPTION>Messier
<OPTION>Howe
```

```
<OPTION>Gretzky
</SELECT>
```

FORMSELECTOPTION Procedure

This procedure generates the `<OPTION>` tag which represents one choice in a Select element.

Syntax

```
HTP.FORMSELECTOPTION(
   cvalue          IN       VARCHAR2,
   cselected       IN       VARCHAR2    DEFAULT NULL,
   cattributes     IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-37   FORMSELECTOPTION Procedure Parameters

Parameter	Description
`cvalue`	The text for the option.
`cvalue`	If the value for this parameter is not `NULL`, the `SELECTED` attribute is added to the tag.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<OPTION SELECTED cattributes>cvalue
```

as shown under Examples of the FORMSELECTOPEN Procedure.

FORMSUBMIT Procedure

This procedure generates the `<INPUT>` tag with `TYPE="submit"` which creates a button that, when clicked, submits the form. If the button has a `NAME` attribute, the button contributes a name/value pair to the submitted data.

Syntax

```
HTP.FORMSUBMIT(
   cname           IN       VARCHAR2    DEFAULT NULL,
   cvalue         IN       VARCHAR2    DEFAULT 'Submit',
   cattributes     IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-38   FORMSUBMIT Procedure Parameters

Parameter	Description
`cname`	The value for the `NAME` attribute.
`cvalue`	The value for the `VALUE` attribute.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<INPUT TYPE="submit" NAME="cname" VALUE="cvalue" cattributes>
```

FORMTEXT Procedure

This procedure generates the `<INPUT>` tag with `TYPE="text"`, which creates a field for a single line of text.

Syntax

```
HTP.FORMTEXT(
   cname          IN       VARCHAR2,
   csize          IN       VARCHAR2   DEFAULT NULL,
   cmaxlength     IN       VARCHAR2   DEFAULT NULL,
   cvalue         IN       VARCHAR2   DEFAULT NULL,
   cattributes    IN       VARCHAR2   DEFAULT NULL);
```

Parameters

Table 247-39 FORMTEXT Procedure Parameters

Parameter	Description
`cname`	The value for the `NAME` attribute.
`csize`	The value for the `SIZE` attribute.
`cmaxlength`	The value for the `MAXLENGTH` attribute.
`cvalue`	The value for the `VALUE` attribute.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<INPUT TYPE="text" NAME="cname" SIZE="csize" MAXLENGTH="cmaxlength" VALUE="cvalue"
cattributes>
```

FORMTEXTAREA Procedure

This procedure generates the `<TEXTAREA>` tag, which creates a text field that has no predefined text in the text area. This field enables entering several lines of text.

The same operation is performed by the FORMTEXTAREA2 Procedure which in addition has the `cwrap` parameter that lets you specify a wrap style.

Syntax

```
HTP.FORMTEXTAREA(
   cname          IN       VARCHAR2,
   nrows          IN       INTEGER,
   ncolumns       IN       INTEGER,
   calign       , IN       VARCHAR2   DEFAULT NULL,
   cattributes    IN       VARCHAR2   DEFAULT NULL);
```

Parameters

Table 247-40 FORMTEXTAREA Procedure Parameters

Parameter	Description
`cname`	The value for the `NAME` attribute.
`nrows`	The value for the `ROWS` attribute.This is an integer.
`ncolumns`	The value for the `COLS` attribute.This is an integer.
`calign`	The value for the `ALIGN` attribute.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<TEXTAREA NAME="cname" ROWS="nrows" COLS="ncolumns" ALIGN="calign" cattributes></
TEXTAREA>
```

FORMTEXTAREA2 Procedure

This procedure generates the `<TEXTAREA>` tag, which creates a text field that has no predefined text in the text area. This field enables entering several lines of text.

The same operation is performed by the FORMTEXTAREA Procedure except that in that case you cannot specify a wrap style.

Syntax

```
HTP.FORMTEXTAREA2(
   cname          IN       VARCHAR2,
   nrows          IN       INTEGER,
   ncolumns       IN       INTEGER,
   calign         IN       VARCHAR2   DEFAULT NULL,
   cwrap          IN       VARCHAR2   DEFAULT NULL,
   cattributes    IN       VARCHAR2   DEFAULT NULL);
```

Parameters

Table 247-41 FORMTEXTAREA2 Procedure Parameters

Parameter	Description
`cname`	The value for the `NAME` attribute.
`nrows`	The value for the `ROWS` attribute.This is an integer.
`ncolumns`	The value for the `COLS` attribute.This is an integer.
`calign`	The value for the `ALIGN` attribute.
`cwrap`	The value for the `WRAP` attribute.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<TEXTAREA NAME="cname" ROWS="nrows" COLS="ncolumns" ALIGN="calign" WRAP="cwrap"
cattributes></TEXTAREA>
```

FORMTEXTAREACLOSE Procedure

This procedure generates the `</TEXTAREA>` tag which ends a text area form element.

You open a text area element by means of either FORMTEXTAREAOPEN Procedure or FORMTEXTAREAOPEN2 Procedure.

Syntax

```
HTP.FORMTEXTAREACLOSE;
```

Examples

This procedure generates

```
</TEXTAREA>
```

FORMTEXTAREAOPEN Procedure

This procedure generates the `<TEXTAREA>` which marks the beginning of a text area form element.

The same operation is performed by the FORMTEXTAREAOPEN2 Procedure which in addition has the `cwrap` parameter that lets you specify a wrap style. You mark the end of a text area form element by means of the FORMTEXTAREACLOSE Procedure.

Syntax

```
HTP.FORMTEXTAREAOPEN(
   cname          IN       VARCHAR2,
   nrows          IN       INTEGER,
   ncolumns       IN       INTEGER,
   calign         IN       VARCHAR2    DEFAULT NULL,
   cattributes    IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-42 FORMTEXTAREAOPEN Procedure Parameters

Parameter	Description
`cname`	The value for the `NAME` attribute.
`nrows`	The value for the `ROWS` attribute.This is an integer.
`ncolumns`	The value for the `COLS` attribute.This is an integer.
`calign`	The value for the `ALIGN` attribute.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<TEXTAREA NAME="cname" ROWS="nrows" COLS="ncolumns" ALIGN="calign" cattributes>
```

FORMTEXTAREAOPEN2 Procedure

This procedure generates the <TEXTAREA> which marks the beginning of a text area form element.

The same operation is performed by the FORMTEXTAREAOPEN Procedure except that in that case you cannot specify a wrap style. You mark the end of a text area form element by means of the FORMTEXTAREACLOSE Procedure.

Syntax

```
HTP.FORMTEXTAREAOPEN2(
   cname          IN       VARCHAR2,
   nrows          IN       INTEGER,
   ncolumns       IN       INTEGER,
   calign         IN       VARCHAR2   DEFAULT NULL,
   cwrap          IN       VARCHAR2   DEFAULT NULL,
   cattributes    IN       VARCHAR2   DEFAULT NULL);
```

Parameters

Table 247-43   FORMTEXTAREAOPEN2 Procedure Parameters

Parameter	Description
`cname`	The value for the `NAME` attribute.
`nrows`	The value for the `ROWS` attribute.This is an integer.
`ncolumns`	The value for the `COLS` attribute.This is an integer.
`calign`	The value for the `ALIGN` attribute.
`cwrap`	The value for the `WRAP` attribute.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<TEXTAREA NAME="cname" ROWS="nrows" COLS="ncolumns" ALIGN="calign" WRAP = "cwrap"
cattributes>
```

FRAME Procedure

This procedure generates the <FRAME> tag which begins the characteristics of a frame created by a <FRAMESET> tag.

Syntax

```
HTP.FRAME(
   csrc           IN       VARCHAR2,
   cname          IN       VARCHAR2   DEFAULT NULL,
   cmarginwidth   IN       VARCHAR2   DEFAULT NULL,
   cmarginheight  IN       VARCHAR2   DEFAULT NULL,
   cscrolling     IN       VARCHAR2   DEFAULT NULL,
   cnoresize      IN       VARCHAR2   DEFAULT NULL,
   cattributes    IN       VARCHAR2   DEFAULT NULL);
```

Parameters

Table 247-44   FRAME Procedure Parameters

Parameter	Description
csrc	The URL to display in the frame.
cname	The value for the NAME attribute.
cmarginwidth	The value for the MARGINWIDTH attribute.
cscrolling	The value for the SCROLLING attribute.
cnoresize	If the value for this parameter is not NULL, the NORESIZE attribute is added to the tag.
cattributes	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<FRAME SRC="csrc" NAME="cname" MARGINWIDTH="cmarginwidth" MARGINHEIGHT="cmarginheight"
SCROLLING="cscrolling" NORESIZE cattributes>
```

FRAMESETCLOSE Procedure

This procedure generates the `</FRAMESET>` tag which ends a frameset section.

You mark the beginning of a frameset section by means of the FRAMESETOPEN Procedure.

Syntax

```
HTP.FRAMESETCLOSE;
```

Examples

This procedure generates

```
</FRAMESET>
```

FRAMESETOPEN Procedure

This procedure generates the `<FRAMESET>` tag which define a frameset section.

You mark the end of a frameset section by means of the FRAMESETCLOSE Procedure.

Syntax

```
HTP.FRAMESETOPEN(
   crows          IN       VARCHAR2   DEFAULT NULL,
   ccols          IN       VARCHAR2   DEFAULT NULL,
   cattributes    IN       VARCHAR2   DEFAULT NULL);
```

Parameters

Table 247-45 FRAMESETOPEN Procedure Parameters

Parameter	Description
`crows`	The value for the `ROWS` attribute.
`ccols`	The value for the `COLS` attribute.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<FRAMESET ROWS="crows" COLS="ccols" cattributes>
```

HEADCLOSE Procedure

This procedure generates the `</HEAD>` tag which marks the end of an HTML document head section.

You mark the beginning of an HTML document head section by means of the HEADOPEN Procedure.

Syntax

```
HTP.HEADCLOSE;
```

Examples

This procedure generates

```
</HEAD>
```

HEADER Procedure

This procedure generates opening heading tags `(<H1>` to `<H6>)` and their corresponding closing tags (`</H1>` to `</H6>)`.

Syntax

```
HTP.HEADER(
   nsize          IN       INTEGER,
   cheader        IN       VARCHAR2,
   calign         IN       VARCHAR2   DEFAULT NULL,
   cnowrap        IN       VARCHAR2   DEFAULT NULL,
   cclear         IN       VARCHAR2   DEFAULT NULL,
   cattributes    IN       VARCHAR2   DEFAULT NULL);
```

Parameters

Table 247-46 HEADER Procedure Parameters

Parameter	Description
`nsize`	The the heading level. This is an integer between 1 and 6.

Table 247-46 (Cont.) HEADER Procedure Parameters

Parameter	Description
`cheader`	The text to display in the heading.
`calign`	The value for the `ALIGN` attribute.
`cnowrap`	The value for the `NOWRAP` attribute.
`cclear`	The value for the `CLEAR` attribute.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

```
HTP.header (1,'Overview');
```

produces:

```
<H1>Overview</H1>
```

HEADOPEN Procedure

This procedure generates the `<HEAD>` tag which marks the beginning of the HTML document head section.

You mark the end of an HTML document head section by means of the HEADCLOSE Procedure.

Syntax

```
HTP.HEADOPEN;
```

Examples

This procedure generates

```
<HEAD>
```

HR Procedure

This procedure generates the `<HR>` tag, which generates a line in the HTML document.

This subprogram performs the same operation as the LINE Procedure.

Syntax

```
HTP.HR(
   cclear          IN       VARCHAR2    DEFAULT NULL,
   csrc            IN       VARCHAR2    DEFAULT NULL,
   cattributes     IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-47 HR Procedure Parameters

Parameter	Description
`cclear`	The value for the `CLEAR` attribute.
`csrc`	The value for the `SRC` attribute which specifies a custom image as the source of the line.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<HR CLEAR="cclear" SRC="csrc" cattributes>
```

HTMLCLOSE Procedure

This procedure generates the `</HTML>` tag which marks the end of an HTML document.

You use the HTMLOPEN Procedure to mark the beginning of an HTML document.

Syntax

```
HTP.HTMLCLOSE;
```

Examples

This procedure generates

```
</HTML>
```

HTMLOPEN Procedure

This procedure generates the <HTML> tag which marks the beginning of an HTML document.

You use the HTMLCLOSE Procedure to mark the end of the an HTML document.

Syntax

```
HTP.HTMLOPEN;
```

Examples

This procedure generates

```
<HTML>
```

IMG Procedure

This procedure generates the `<IMG>` tag which directs the browser to load an image onto the HTML page.

The IMG2 Procedure performs the same operation but additionally uses the `cusemap` parameter.

Syntax

```
HTP.IMG(
   curl           IN       VARCHAR2   DEFAULT NULL,
   calign         IN       VARCHAR2   DEFAULT NULL,
   calt           IN       VARCHAR2   DEFAULT NULL,
   cismap         IN       VARCHAR2  DEFAULT NULL,
   cattributes    IN       VARCHAR2   DEFAULT NULL);
```

Parameters

Table 247-48 IMG Procedure Parameters

Parameter	Description
`curl`	The value for the SRC attribute.
`calign`	The value for the `ALIGN` attribute.
`calt`	The value for the `ALT` attribute which specifies alternative text to display if the browser does not support images.
`cismap`	If the value for this parameter is not `NULL`, the `ISMAP` attribute is added to the tag. The attribute indicates that the image is an imagemap.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<IMG SRC="curl" ALIGN="calign" ALT="calt" ISMAP cattributes>
```

IMG2 Procedure

This procedure generates the <IMG> tag, which directs the browser to load an image onto the HTML page.

The IMG Procedure performs the same operation but does not use the `cusemap` parameter.

Syntax

```
HTP.IMG2(
   curl           IN       VARCHAR2   DEFAULT NULL,
   calign         IN       VARCHAR2   DEFAULT NULL,
   calt           IN       VARCHAR2   DEFAULT NULL,
   cismap         IN       VARCHAR2  DEFAULT NULL,
   cusemap        IN       VARCHAR2   DEFAULT NULL,
   cattributes    IN       VARCHAR2   DEFAULT NULL);
```

Parameters

Table 247-49 IMG2 Procedure Parameters

Parameter	Description
`curl`	The value for the SRC attribute.
`calign`	The value for the `ALIGN` attribute.

Table 247-49 (Cont.) IMG2 Procedure Parameters

Parameter	Description
`calt`	The value for the `ALT` attribute which specifies alternative text to display if the browser does not support images.
`cismap`	If the value for this parameter is not `NULL`, the `ISMAP` attribute is added to the tag. The attribute indicates that the image is an imagemap.
`cusemap`	The value for the `USEMAP` attribute which specifies a client-side image map.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<IMG SRC="curl" ALIGN="calign" ALT="calt" ISMAP USEMAP="cusemap" cattributes>
```

ISINDEX Procedure

This procedure creates a single entry field with a prompting text, such as "*enter value,*" then sends that value to the URL of the page or program.

Syntax

```
HTP.ISINDEX(
   cprompt        IN       VARCHAR2     DEFAULT NULL,
   curl           IN       VARCHAR2     DEFAULT NULL);
```

Parameters

Table 247-50 ISINDEX Procedure Parameters

Parameter	Description
`cprompt`	The value for the `PROMPT` attribute.
`curl`	The value for the `HREF` attribute.

Examples

This procedure generates

```
<ISINDEX PROMPT="cprompt" HREF="curl">
```

ITALIC Procedure

This procedure generates the `<I>` and `</I>` tags which direct the browser to render the text in italics.

Syntax

```
HTP.ITALIC(
   ctext          IN       VARCHAR2,
   cattributes    IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-51 ITALIC Procedure Parameters

Parameter	Description
`ctext`	The text to be rendered in italics.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<I cattributes>ctext</I>
```

KBD Procedure

This procedure generates the `<KBD>` and `</KBD>` tags which direct the browser to render the text in monospace font.

This subprogram performs the same operation as the KEYBOARD Procedure.

Syntax

```
HTP.KBD(
   ctext           IN       VARCHAR2,
   cattributes     IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-52 KBD Procedure Parameters

Parameter	Description
`ctext`	The text to be rendered in monospace.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<KBD cattributes>ctext</KBD>
```

KEYBOARD Procedure

This procedure generates the <KBD> and </KBD> tags, which direct the browser to render the text in monospace font.

This subprogram performs the same operation as the KBD Procedure.

Syntax

```
HTP.KEYBOARD(
   ctext           IN       VARCHAR2,
   cattributes     IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-53 KEYBOARD Procedure Parameters

Parameter	Description
`ctext`	The text to be rendered in monospace.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<KBD cattributes>ctext</KBD>
```

LINE Procedure

This procedure generates the `<HR>` tag, which generates a line in the HTML document.

This subprogram performs the same operation as the HR Procedure.

Syntax

```
HTP.LINE(
   cclear         IN       VARCHAR2   DEFAULT NULL,
   csrc           IN       VARCHAR2   DEFAULT NULL,
   cattributes    IN       VARCHAR2   DEFAULT NULL);
```

Parameters

Table 247-54 LINE Procedure Parameters

Parameter	Description
`cclear`	The value for the `CLEAR` attribute.
`csrc`	The value for the `SRC` attribute which specifies a custom image as the source of the line.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<HR CLEAR="cclear" SRC="csrc" cattributes>
```

LINKREL Procedure

This procedure generates the `<LINK>` tag with the `REL` attribute which delineates the relationship described by the hypertext link from the anchor to the target. This is only used when the `HREF` attribute is present.

This is the opposite of LINKREV Procedure. This tag indicates a relationship between documents but does not create a link. To create a link, use the ANCHOR Procedure.

Syntax

```
HTP.LINKREL(
   crel           IN       VARCHAR2,
   curl           IN       VARCHAR2,
   ctitle         IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-55 LINKREL Procedure Parameters

Parameter	Description
`crel`	The value for the `REL` attribute.
`curl`	The value for the `URL` attribute.
`ctitle`	The value for the `TITLE` attribute.

Examples

This procedure generates

```
<LINK REL="crel" HREF="curl" TITLE="ctitle">
```

LINKREV Procedure

This procedure generates the `<LINK>` tag with the `REV` attribute which delineates the relationship described by the hypertext link from the target to the anchor.

This is the opposite of the LINKREL Procedure. This tag indicates a relationship between documents, but does not create a link. To create a link, use the ANCHOR Procedure.

Syntax

```
HTP.LINKREV(
   crev           IN       VARCHAR2,
   curl           IN       VARCHAR2,
   ctitle         IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-56 LINKREV Procedure Parameters

Parameter	Description
`crel`	The value for the `REV` attribute.
`curl`	The value for the `URL` attribute.
`ctitle`	The value for the `TITLE` attribute.

Examples

This procedure generates

```
<LINK REV="crev" HREF="curl" TITLE="ctitle">
```

LISTHEADER Procedure

This procedure generates the `<LH>` and `</LH>` tags which print an HTML tag at the beginning of the list.

Syntax

```
HTP.LISTHEADER(
   ctext          IN       VARCHAR2,
   cattributes    IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-57 LISTHEADER Procedure Parameters

Parameter	Description
`ctext`	The text to place between `<LH>` and `</LH>`.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<LH cattributes>ctext</LH>
```

LISTINGCLOSE Procedure

This procedure generates the `</LISTING>` tags which marks the end of a section of fixed-width text in the body of an HTML page.

To mark the beginning of a section of fixed-width text in the body of an HTML page, use the LISTINGOPEN Procedure.

Syntax

```
HTP.LISTINGCLOSE;
```

Examples

This procedure generates

```
</LISTING>
```

LISTINGOPEN Procedure

This procedure generates the `<LISTING>` tag which marks the beginning of a section of fixed-width text in the body of an HTML page.

To mark the end of a section of fixed-width text in the body of an HTML page, use the LISTINGCLOSE Procedure.

Syntax

```
HTP.LISTINGOPEN;
```

Examples

This procedure generates

```
<LISTING>
```

LISTITEM Procedure

This procedure generates the `<LI>` tag, which indicates a list item.

Syntax

```
HTP.LISTITEM(
   ctext          IN       VARCHAR2   DEFAULT NULL,
   cclear         IN       VARCHAR2   DEFAULT NULL,
   cdingbat       IN       VARCHAR2   DEFAULT NULL,
   csrc           IN       VARCHAR2   DEFAULT NULL,
   cattributes    IN       VARCHAR2   DEFAULT NULL);
```

Parameters

Table 247-58 LISTITEM Procedure Parameters

Parameter	Description
`ctext`	The text for the list item.
`cclear`	The value for the `CLEAR` attribute.
`cdingbat`	The value for the `DINGBAT` attribute.
`csrc`	The value for the `SRC` attribute.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<LI CLEAR="cclear" DINGBAT="cdingbat" SRC="csrc" cattributes>ctext
```

MAILTO Procedure

This procedure generates the `<A>` tag with the `HREF` set to 'mailto' prepended to the mail address argument.

Syntax

```
HTP.MAILTO(
   caddress       IN       VARCHAR2,
   ctext          IN       VARCHAR2,
   cname          IN       VARCHAR2,
   cattributes    IN       VARCHAR2   DEFAULT NULL);
```

Parameters

Table 247-59 MAILTO Procedure Parameters

Parameter	Description
caddress	The email address of the recipient.
ctext	The clickable portion of the link.
cname	The value for the NAME attribute.
cattributes	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<A HREF="mailto:caddress" NAME="cname" cattributes>ctext</A>
```

so that

```
HTP.mailto('pres@white_house.gov','Send Email to the President');
```

generates:

```
<A HREF="mailto:pres@white_house.gov">Send Email to the President</A>
```

MAPCLOSE Procedure

This procedure generates the `</MAP>` tag which marks the end of a set of regions in a client-side image map.

To mark the beginning of a set of regions in a client-side image map, use the MAPOPEN Procedure.

Syntax

```
HTP.MAPCLOSE;
```

Examples

This procedure generates

```
</MAP>
```

MAPOPEN Procedure

This procedure generates the `<MAP>` tag which mark the beginning of a set of regions in a client-side image map.

To mark the end of a set of regions in a client-side image map, use the MAPCLOSE Procedure.

Syntax

```
HTP.MAPOPEN(
   cname           IN        VARCHAR2    DEFAULT NULL,
   cattributes     IN        VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-60 MAPOPEN Procedure Parameters

Parameter	Description
`cname`	The value for the `NAME` attribute.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<MAP NAME="cname" cattributes>
```

MENULISTCLOSE Procedure

This procedure generates the `</MENU>` tag which ends a list that presents one line for each item.

To begin a list of this kind, use the MENULISTOPEN Procedure. The items in the list appear more compact than an unordered list. The LISTITEM Procedure defines the list items in a menu list.

Syntax

```
HTP.MENULISTCLOSE;
```

Examples

This procedure generates

```
</MENU>
```

MENULISTOPEN Procedure

This procedure generates the `<MENU>` tag which begins a list that presents one line for each item.

To end a list of this kind, use the MENULISTCLOSE Procedure.The items in the list appear more compact than an unordered list. The LISTITEM Procedure defines the list items in a menu list.

Syntax

```
HTP.MENULISTOPEN;
```

Examples

This procedure generates

```
<MENU>
```

META Procedure

This procedure generates the `<META>` tag, which embeds meta-information about the document and also specifies values for HTTP headers. For example, you can specify the expiration date, keywords, and author name.

Syntax

```
HTP.META(
   chttp_equiv    IN       VARCHAR2,
   cname          IN       VARCHAR2,
   ccontent       IN       VARCHAR2);
```

Parameters

Table 247-61 META Procedure Parameters

Parameter	Description
`chttp_equiv`	The value for the `CHTTP_EQUIV` attribute.
`cname`	The value for the `NAME` attribute.
`ccontent`	The value for the `CONTENT` attribute.

Examples

This procedure generates

```
<META HTTP-EQUIV="chttp_equiv" NAME ="cname" CONTENT="ccontent">
```

so that

```
HTP.meta ('Refresh', NULL, 120);
```

generates

```
<META HTTP-EQUIV="Refresh" CONTENT=120>
```

On some Web browsers, this causes the current URL to be reloaded automatically every 120 seconds.

NL Procedure

This procedure generates the `<BR>` tag which begins a new line of text.

It performs the same operation as the BR Procedure.

Syntax

```
HTP.NL(
   cclear         IN       VARCHAR2   DEFAULT NULL,
   cattributes    IN       VARCHAR2   DEFAULT NULL);
```

Parameters

Table 247-62 NL Procedure Parameters

Parameter	Description
`cclear`	The value for the `CLEAR` attribute.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<BR CLEAR="cclear" cattributes>
```

NOBR Procedure

This procedure generates the `<NOBR>` and `</NOBR>` tags which turn off line-breaking in a section of text.

Syntax

```
HTP.NOBR(
ctext        IN        VARCHAR2);
```

Parameters

Table 247-63 NOBR Procedure Parameters

Parameter	Description
`ctext`	The text that is to be rendered on one line.

Examples

This procedure generates

```
<NOBR>ctext</NOBR>
```

NOFRAMESCLOSE Procedure

This procedure generates the `</NOFRAMES>` tag which marks the end of a no-frames section.

To mark the beginning of a no-frames section, use the FRAMESETOPEN Procedure. See also FRAME Procedure, FRAMESETOPEN Procedure and FRAMESETCLOSE Procedure.

Syntax

```
HTP.NOFRAMESCLOSE;
```

Examples

This procedure generates

```
</NOFRAMES>
```

NOFRAMESOPEN Procedure

This procedure generates the `<NOFRAMES>` tag which mark the beginning of a no-frames section.

To mark the end of a no-frames section, use the FRAMESETCLOSE Procedure. See also FRAME Procedure, FRAMESETOPEN Procedure and FRAMESETCLOSE Procedure.

Syntax

```
HTP.NOFRAMESOPEN;
```

Examples

This procedure generates

```
<NOFRAMES>
```

OLISTCLOSE Procedure

This procedure generates the `</OL>` tag which defines the end of an ordered list. An ordered list presents a list of numbered items.

To mark the beginning of a list of this kind, use the OLISTOPEN Procedure. Numbered items are added using LISTITEM Procedure.

Syntax

```
HTP.OLISTCLOSE;
```

Examples

This procedure generates

```
</OL>
```

OLISTOPEN Procedure

This procedure generates the `<OL>` tag which marks the beginning of an ordered list. An ordered list presents a list of numbered items.

To mark the end of a list of this kind, use the OLISTCLOSE Procedure. Numbered items are added using LISTITEM Procedure.

Syntax

```
HTP.OLISTOPEN(
   cclear         IN       VARCHAR2    DEFAULT NULL,
   cwrap          IN       VARCHAR2    DEFAULT NULL,
   cattributes    IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-64   OLISTOPEN Procedure Parameters

Parameter	Description
`cclear`	The value for the `CLEAR` attribute.
`cwrap`	The value for the `WRAP` attribute.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<OL CLEAR="cclear" WRAP="cwrap" cattributes>
```

PARA Procedure

This procedure generates the `<P>` tag which indicates that the text that comes after the tag is to be formatted as a paragraph.

You can add attributes to the tag by means of the PARAGRAPH Procedure.

Syntax

```
HTP.PARA;
```

Examples

This procedure generates

```
<P>
```

PARAGRAPH Procedure

You can use this procedure to add attributes to the `<P>` tag created by the PARA Procedure.

Syntax

```
HTP.PARAGRAPH(
   calign         IN       VARCHAR2    DEFAULT NULL,
   cnowrap        IN       VARCHAR2    DEFAULT NULL,
   cclear         IN       VARCHAR2    DEFAULT NULL,
   cattributes    IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-65   PARAGRAPH Procedure Parameters

Parameter	Description
`calign`	The value for the `ALIGN` attribute.
`cnowrap`	If the value for this parameter is not `NULL`, the `NOWRAP` attribute is added to the tag.
`cclear`	The value for the `CLEAR` attribute.

Table 247-65 (Cont.) PARAGRAPH Procedure Parameters

Parameter	Description
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<P ALIGN="calign" NOWRAP CLEAR="cclear" cattributes>
```

Related Topics

- PARA Procedure
  This procedure generates the `<P>` tag which indicates that the text that comes after the tag is to be formatted as a paragraph.

PARAM Procedure

This procedure generates the `<PARAM>` tag which specifies parameter values for Java applets.

The values can reference HTML variables. To invoke a Java applet from a Web page, use APPLETOPEN Procedure to begin the invocation. Use one PARAM Procedure for each desired name-value pair, and use APPLETCLOSE Procedure to end the applet invocation.

Syntax

```
HTP.PARAM(
   cname          IN       VARCHAR2
   cvalue         IN       VARCHAR2);
```

Parameters

Table 247-66 PARAM Procedure Parameters

Parameter	Description
`cname`	The value for the `NAME` attribute.
`cvalue`	The value for the `VALUE` attribute.

Examples

This procedure generates

```
<PARAM NAME=cname VALUE="cvalue">
```

PLAINTEXT Procedure

This procedure generates the `<PLAINTEXT>` and `</PLAINTEXT>` tags which direct the browser to render the text they surround in fixed-width type.

Syntax

```
HTP.PLAINTEXT(
   ctext          IN       VARCHAR2,
   cattributes    IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-67 PLAINTEXT Procedure Parameters

Parameter	Description
`ctext`	The text to be rendered in fixed-width font.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<PLAINTEXT cattributes>ctext</PLAINTEXT>
```

PRECLOSE Procedure

This procedure generates the `</PRE>` tag which marks the end of a section of preformatted text in the body of the HTML page.

To mark the beginning of a section of preformatted text in the body of the HTML page, use the PREOPEN Procedure.

Syntax

```
HTP.PRECLOSE;
```

Examples

This procedure generates

```
</PRE>
```

PREOPEN Procedure

This procedure generates the `<PRE>` tag which marks the beginning of a section of preformatted text in the body of the HTML page.

To mark the end of a section of preformatted text in the body of the HTML page, use the PRECLOSE Procedure.

Syntax

```
HTP.PREOPEN(
   cclear          IN        VARCHAR2    DEFAULT NULL,
   cwidth          IN        VARCHAR2    DEFAULT NULL,
   cattributes     IN        VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-68 PREOPEN Procedure Parameters

Parameter	Description
`cclear`	The value for the `CLEAR` attribute.
`cwidth`	The value for the `WIDTH` attribute.

Table 247-68 (Cont.) PREOPEN Procedure Parameters

Parameter	Description
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<PRE CLEAR="cclear" WIDTH="cwidth" cattributes>
```

PRINT Procedures

These procedures generate the specified parameter as a string terminated with the `\n` newline character.

The PRN Procedures performs the same operation but does not terminate with a newline character.

Syntax

```
HTP.PRINT (
   cbuf      IN       VARCHAR2);

HTP.PRINT (
   dbuf      IN       DATE);

HTP.PRINT (
   nbuf      IN       NUMBER);
```

Parameters

Table 247-69 PRINT Procedure Parameters

Parameter	Description
`cbuf`	The string to generate terminated by a newline.
`dbuf`	The string to generate terminated by a newline.
`nbuf`	The string to generate terminated by a newline.

Usage Notes

- The `\n` character is not the same as `<BR>`. The `\n` character formats the HTML source but it does not affect how the browser renders the HTML source. Use `<BR>` to control how the browser renders the HTML source.
- These procedures do not have function equivalents.

PRINTS Procedure

This procedure generates a string and replaces certain characters with a corresponding escape sequence.

The following characters are replaced with the corresponding escape sequence.

- < to &lt;
- > to &gt;
- " to &quot;
- & to &amp;

If not replaced, the special characters are interpreted as HTML control characters and produce garbled output. This procedure an the PS Procedure perform the same operation as the PRN Procedures but with character substitution.

Syntax

```
HTP.PRINTS (
   ctext       IN        VARCHAR2);
```

Parameters

Table 247-70 PRINTS Procedure Parameters

Parameter	Description
ctext	The string where to perform character substitution.

Usage Notes

This procedure does not have an HTF function equivalent (see Operational Notes for the HTF implementation).

PRN Procedures

These procedures generate the specified parameter as a string.

Unlike the PRINT Procedures the string is not terminated with the \n newline character.

Syntax

```
HTP.PRN (
   cbuf       IN        VARCHAR2);

HTP.PRN (
   dbuf       IN        DATE);

HTP.PRN (
   nbuf       IN        NUMBER);
```

Parameters

Table 247-71 PRN Procedure Parameters

Parameter	Description
cbuf	The string to generate (not terminated by a newline).
dbuf	The string to generate (not terminated by a newline).
nbuf	The string to generate (not terminated by a newline).

Usage Notes

These procedures do not have function equivalents.

PS Procedure

This procedure generates a string and replaces certain characters with the corresponding escape sequence.

This procedure replaces the following characters with the corresponding escape sequence.

- < to &lt;
- > to &gt;
- " to &quot;
- & to &amp;

If not replaced, the special characters are interpreted as HTML control characters and produce garbled output. This procedure and the PRINTS Procedure perform the same operation as the PRN Procedures but with character substitution.

Syntax

```
HTP.PS (
   ctext      IN       VARCHAR2);
```

Parameters

Table 247-72 PS Procedure Parameters

Parameter	Description
ctext	The string where to perform character substitution.

Usage Notes

This procedure does not have an HTF function equivalent (see Operational Notes for the HTF implementation).

S Procedure

This procedure generates the <S> and </S> tags which direct the browser to render the text they surround in strikethrough type.

This performs the same operation as STRIKE Procedure.

Syntax

```
HTP.S (
   ctext          IN       VARCHAR2,
   cattributes    IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-73   S Procedure Parameters

Parameter	Description
`ctext`	The text to be rendered in strikethrough type.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<S cattributes>ctext</S>
```

SAMPLE Procedure

This procedure generates the `<SAMP>` and `</SAMP>` tags which direct the browser to render the text they surround in monospace font or however "sample" is defined stylistically.

Syntax

```
HTP.SAMPLE (
   ctext          IN       VARCHAR2,
   cattributes    IN       VARCHAR2   DEFAULT NULL);
```

Parameters

Table 247-74   SAMPLE Procedure Parameters

Parameter	Description
`ctext`	The text to be rendered in monospace font.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<SAMP cattributes>ctext</SAMP>
```

SCRIPT Procedure

This procedure generates the `<SCRIPT>` and `</SCRIPT>` tags which contain a script written in languages such as JavaScript and VBscript.

Syntax

```
HTP.SCRIPT (
   cscript        IN       VARCHAR2,
   clanguage      IN       VARCHAR2   DEFAULT NULL);
```

Parameters

Table 247-75 SCRIPT Procedure Parameters

Parameter	Description
`cscript`	The text of the script. This is the text that makes up the script itself, not the name of a file containing the script.
`clanguage`	The language in which the script is written. If this parameter is omitted, the user's browser determines the scripting language.

Examples

This procedure generates

```
<SCRIPT LANGUAGE=clanguage>cscript</SCRIPT>
```

so that

```
HTP.script ('Erupting_Volcano', 'Javascript');
```

generates

```
<SCRIPT LANGUAGE=Javascript>"script text here"</SCRIPT>
```

This causes the browser to run the script enclosed in the tags.

SMALL Procedure

This procedure generates the `<SMALL>` and `</SMALL>` tags, which direct the browser to render the text they surround using a small font.

Syntax

```
HTP.SMALL (
   ctext          IN       VARCHAR2,
   cattributes    IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-76 SMALL Procedure Parameters

Parameter	Description
`ctext`	The text to be rendered in small font.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<SMALL cattributes>ctext</SMALL>
```

STRIKE Procedure

This procedure generates the `<STRIKE>` and `</STRIKE>` tags which direct the browser to render the text they surround in strikethrough type.

This performs the same operation as S Procedure.

Syntax

```
HTP.STRIKE (
   ctext           IN       VARCHAR2,
   cattributes     IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-77 STRIKE Procedure Parameters

Parameter	Description
`ctext`	The text to be rendered in strikethrough type.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<STRIKE cattributes>ctext</STRIKE>
```

STRONG Procedure

This procedure generates the `<STRONG>` and `</STRONG>` tags which direct the browser to render the text they surround in bold, or however "strong" is defined.

Syntax

```
HTP.STRONG(
   ctext           IN       VARCHAR2,
   cattributes     IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-78 STRONG Procedure Parameters

Parameter	Description
`ctext`	The text to be emphasized.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<STRONG cattributes>ctext</STRONG>
```

STYLE Procedure

This procedure generates the <STYLE> and </STYLE> tags which include a style sheet in a Web page.

You can get more information about style sheets at http://www.w3.org. This feature is not compatible with browsers that support only HTML versions 2.0 or earlier. Such browsers will ignore this tag.

Syntax

```
HTP.STYLE(
   cstyle          IN        VARCHAR2);
```

Parameters

Table 247-79 STYLE Procedure Parameters

Parameter	Description
cstyle	The the style information to include.

Examples

This procedure generates

```
<STYLE>cstyle</STYLE>
```

SUB Procedure

This procedure generates the <SUB> and </SUB> tags which direct the browser to render the text they surround as subscript.

Syntax

```
HTP.SUB(
   ctext          IN        VARCHAR2,
   calign         in        VARCHAR2    DEFAULT NULL,
   cattributes    IN        VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-80 SUB Procedure Parameters

Parameter	Description
ctext	The text to render in subscript.
calign	The value for the ALIGN attribute.
cattributes	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<SUB ALIGN="calign" cattributes>ctext</SUB>
```

SUP Procedure

This procedure generates the `<SUP>` and `</SUP>` tags which direct the browser to render the text they surround as superscript.

Syntax

```
HTP.SUP(
   ctext          IN        VARCHAR2,
   calign         in        VARCHAR2    DEFAULT NULL,
   cattributes    IN        VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-81   SUP Procedure Parameters

Parameter	Description
`ctext`	The text to render in superscript.
`calign`	The value for the `ALIGN` attribute.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<SUP ALIGN="calign" cattributes>ctext</SUP>
```

TABLECAPTION Procedure

This procedure generates the `<CAPTION>` and `</CAPTION>` tags which place a caption in an HTML table.

Syntax

```
HTP.TABLECAPTION(
   ccaption       IN        VARCHAR2,
   calign         in        VARCHAR2    DEFAULT NULL,
   cattributes    IN        VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-82   TABLECAPTION Procedure Parameters

Parameter	Description
`ctext`	The text for the caption.
`calign`	The value for the `ALIGN` attribute.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<CAPTION ALIGN="calign" cattributes>ccaption</CAPTION>
```

TABLECLOSE Procedure

This procedure generates the `</TABLE>` tag which marks the end of an HTML table.

To define the beginning of an HTML table, use the TABLEOPEN Procedure.

Syntax

```
HTP.TABLECLOSE;
```

Examples

This procedure generates

```
</TABLE>
```

TABLEDATA Procedure

This procedure generates the `<TD>` and `</TD>` tags which insert data into a cell of an HTML table.

Syntax

```
HTP.TABLEDATA(
   cvalue          IN       VARCHAR2   DEFAULT NULL,
   calign          IN       VARCHAR2   DEFAULT NULL,
   cdp             IN       VARCHAR2   DEFAULT NULL,
   cnowrap         IN       VARCHAR2   DEFAULT NULL,
   crowspan        IN       VARCHAR2   DEFAULT NULL,
   ccolspan        IN       VARCHAR2   DEFAULT NULL,
   cattributes     IN       VARCHAR2   DEFAULT NULL);
```

Parameters

Table 247-83 TABLEDATA Procedure Parameters

Parameter	Description
`cvalue`	The data for the cell in the table.
`calign`	The value for the `ALIGN` attribute.
`cdp`	The value for the `DP` attribute.
`cnowrap`	If the value of this parameter is not `NULL`, the `NOWRAP` attribute is added to the tag.
`ccolspan`	The value for the `COLSPAN` attribute.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<TD ALIGN="calign" DP="cdp" ROWSPAN="crowspan" COLSPAN="ccolspan" NOWRAP
cattributes>cvalue</TD>
```

TABLEHEADER Procedure

This procedure generates the <TH> and </TH> tags which insert a header cell in an HTML table.

The <TH> tag is similar to the <TD> tag except that the text in this case the rows are usually rendered in bold type.

Syntax

```
HTP.TABLEHEADER(
   cvalue         IN       VARCHAR2   DEFAULT NULL,
   calign         IN       VARCHAR2   DEFAULT NULL,
   cdp            IN       VARCHAR2   DEFAULT NULL,
   cnowrap        IN       VARCHAR2   DEFAULT NULL,
   crowspan       IN       VARCHAR2   DEFAULT NULL,
   ccolspan       IN       VARCHAR2   DEFAULT NULL,
   cattributes    IN       VARCHAR2   DEFAULT NULL);
```

Parameters

Table 247-84 TABLEHEADER Procedure Parameters

Parameter	Description
cvalue	The data for the cell in the table.
calign	The value for the ALIGN attribute.
cdp	The value for the DP attribute.
cnowrap	If the value of this parameter is not NULL, the NOWRAP attribute is added to the tag.
crispen	The value for the ROWSPAN attribute.
ccolspan	The value for the COLSPAN attribute.
cattributes	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<TH ALIGN="calign" DP="cdp" ROWSPAN="crowspan" COLSPAN="ccolspan" NOWRAP
cattributes>cvalue</TH>
```

TABLEOPEN Procedure

This procedure generates the <TABLE> tag which marks the beginning of an HTML table.

To define the end of an HTML table, use the TABLECLOSE Procedure.

Syntax

```
HTP.TABLEOPEN(
   cborder        IN       VARCHAR2   DEFAULT NULL
   calign         IN       VARCHAR2   DEFAULT NULL,
   cnowrap        IN       VARCHAR2   DEFAULT NULL,
   cclear         IN       VARCHAR2   DEFAULT NULL
   cattributes    IN       VARCHAR2   DEFAULT NULL);
```

Parameters

Table 247-85 TABLEOPEN Procedure Parameters

Parameter	Description
`border`	The value for the `BORDER` attribute.
`calign`	The value for the `ALIGN` attribute.
`cnowrap`	If the value of this parameter is not `NULL`, the `NOWRAP` attribute is added to the tag.
`cclear`	The value for the `CLEAR` attribute.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<TABLE "cborder" NOWRAP ALIGN="calign" CLEAR="cclear" cattributes>
```

TABLEROWCLOSE Procedure

This procedure generates the `</TR>` tag which marks the end of a new row in an HTML table.

To mark the beginning of a new row, use the TABLEROWOPEN Procedure.

Syntax

```
HTP.TABLEROWCLOSE;
```

Examples

This procedure generates

```
</TABLE>
```

TABLEROWOPEN Procedure

This procedure generates the `<TR>` tag which marks the beginning of a new row in an HTML table.

To mark the end of a new row, use the TABLEROWCLOSE Procedure.

Syntax

```
HTP.TABLEROWOPEN(
   calign         IN       VARCHAR2   DEFAULT NULL,
   cvalign        IN       VARCHAR2   DEFAULT NULL,
   cdp            IN       VARCHAR2   DEFAULT NULL,
   cnowrap        IN       VARCHAR2   DEFAULT NULL,
   cattributes    IN       VARCHAR2   DEFAULT NULL);
```

Parameters

Table 247-86 TABLEROWOPEN Procedure Parameters

Parameter	Description
`calign`	The value for the `ALIGN` attribute.
`cvalign`	The value for the `VALIGN` attribute.
`cdp`	The value for the `DP` attribute.
`cnowrap`	If the value of this parameter is not `NULL`, the `NOWRAP` attribute is added to the tag.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<<TR ALIGN="calign" VALIGN="cvalign" DP="cdp" NOWRAP cattributes>
```

TELETYPE Procedure

This procedure generates the `<TT>` and `</TT>` tags which direct the browser to render the text they surround in a fixed width typewriter font, for example, the courier font.

Syntax

```
HTP.TELETYPE(
   ctext          IN       VARCHAR2,
   cattributes    IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-87 TELETYPE Procedure Parameters

Parameter	Description
`ctext`	The text to render in a fixed width typewriter font.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<TT cattributes>ctext</TT>
```

TITLE Procedure

This procedure generates the `<TITLE>` and `</TITLE>` tags which specify the text to display in the titlebar of the browser window.

Syntax

```
HTP.TITLE(
   ctitle           IN       VARCHAR2);
```

Parameters

Table 247-88 TITLE Procedure Parameters

Parameter	Description
`ctitle`	The text to display in the titlebar of the browser window.

Examples

This procedure generates

```
<TITLE>ctitle</TITLE>
```

ULISTCLOSE Procedure

This procedure generates the `</UL>` tag which marks the end of an unordered list. An unordered list presents items with bullets.

To mark the beginning of an unordered list, use the ULISTOPEN Procedure. Add list items with LISTITEM Procedure.

Syntax

```
HTP.ULISTCLOSE;
```

Examples

This procedure generates

```
</TABLE>
```

ULISTOPEN Procedure

This procedure generates the `<UL>` tag which marks the beginning of an unordered list. An unordered list presents items with bullets.

To mark the end of an unordered list, use the ULISTCLOSE Procedure. Add list items with LISTITEM Procedure.

Syntax

```
HTP.ULISTOPEN(
   cclear        IN       VARCHAR2    DEFAULT NULL,
   cwrap         IN       VARCHAR2    DEFAULT NULL,
   cdingbat      IN       VARCHAR2    DEFAULT NULL,
   csrc          IN       VARCHAR2    DEFAULT NULL,
   cattributes   IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-89 ULISTOPEN Procedure Parameters

Parameter	Description
`cclear`	The value for the `CLEAR` attribute.
`cwrap`	The value for the `WRAP` attribute.

Table 247-89 (Cont.) ULISTOPEN Procedure Parameters

Parameter	Description
`cdingbat`	The value for the `DINGBAT` attribute.
`csrc`	The value for the `SRC` attribute.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<UL CLEAR="cclear" WRAP="cwrap" DINGBAT="cdingbat" SRC="csrc" cattributes>
```

UNDERLINE Procedure

This procedure generates the `<U>` and `</U>` tags, which direct the browser to render the text they surround with an underline.

Syntax

```
HTP.UNDERLINE(
   ctext          IN       VARCHAR2,
   cattributes    IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-90 UNDERLINE Procedure Parameters

Parameter	Description
`ctext`	The text to render with an underline.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<U cattributes>ctext</U>
```

VARIABLE Procedure

This procedure generates the `<VAR>` and `</VAR>` tags which direct the browser to render the text they surround in italics or however "variable" is defined stylistically.

Syntax

```
HTP.VARIABLE(
   ctext          IN       VARCHAR2,
   cattributes    IN       VARCHAR2    DEFAULT NULL);
```

Parameters

Table 247-91 VARIABLE Procedure Parameters

Parameter	Description
`ctext`	The text to render in italics.
`cattributes`	The other attributes to be included as-is in the tag.

Examples

This procedure generates

```
<VAR cattributes>ctext</VAR>
```

WBR Procedure

This procedure generates the `<WBR>` tag, which inserts a soft line break within a section of `NOBR` text.

Syntax

```
HTP.WBR;
```

Examples

This procedure generates

```
<WBR>
```