

246

HTF

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



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*

This chapter contains the following topics:

- [Deprecated Subprograms](#)
- [Operational Notes](#)
- [Rules and Limits](#)
- [Examples](#)
- [Summary of Tags](#)
- [Summary of HTF Subprograms](#)

HTF Deprecated Subprograms

The `ESCAPE_URL` function is deprecated with Oracle Database 10g.



Note:

Oracle recommends that you do not use deprecated procedures in new applications. Support for deprecated features is for backward compatibility only.

Related Topics

- [ESCAPE_URL Function](#)
This deprecated function replaces characters that have special meaning in HTML and HTTP with their escape sequences.

HTF Operational Notes

These operational notes apply to the `HTF` package.

For every HTF function that generates one or more HTML tags, there is a corresponding HTP procedure 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 HTF.PRINT function. It then prints its parameters to the generated Web page.

Rules and Limits

If you use values of the LONG datatype in functions such as HTF.PRINT, HTF.PRN, HTF.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 function.

HTF Examples

The commands in this example generate a simple HTML document.

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

Summary of Tags

This list summarizes the HTML tags generated by the HTF package.

HTML, HEAD, and BODY Tags

[HTMLOPEN Function](#), [HTMLCLOSE Function](#) - generate <HTML> and </HTML>

[HEADOPEN Function](#), [HEADCLOSE Function](#) - generate <HEAD> and </HEAD>

[BODYOPEN Function](#), [BODYCLOSE Function](#) - generate <BODY> and </BODY>

Comment Tag

[COMMENT Function](#) - generates <!-- and -->

[http://www.w3.org.BASE Function](#) - generates <BASE>

[LINKREL Function](#) - generates <LINK> with the REL attribute

[LINKREV Function](#) - generates <LINK> with the REV attribute

[TITLE Function](#) - generates <TITLE>

[META Function](#) - generates <META>

[SCRIPT Function](#) - generates <SCRIPT>

[STYLE Function](#) - generates <STYLE>

[ISINDEX Function](#) - generates <ISINDEX>

Applet Tags

[APPLETOPEN Function](#), [APPLETCLOSE Function](#) - generate <APPLET> and </APPLET>

[PARAM Function](#) - generates <PARAM>

List Tags

[OLISTOPEN Function](#), [OLISTCLOSE Function](#) - generate and

[ULISTOPEN Function](#), [ULISTCLOSE Function](#) - generate and

[DLISTOPEN Function](#), [DLISTCLOSE Function](#) - generate <DL> and </DL>

[DLISTTERM Function](#) - generates <DT>

[DLISTDEF Function](#) - generates <DD>

[DIRLISTOPEN Function](#), [DIRLISTCLOSE Function](#) - generate <DIR> and </DIR>

[LISTHEADER Function](#) - generates <LH>

[LISTINGOPEN Function](#), [LISTINGCLOSE Function](#) - generate <LISTING> and </LISTING>

[MENULISTOPEN Function](#) - generate <MENU> and </MENU>

[LISTITEM Function](#) - generates

Form Tags

[FORMOPEN Function](#), [FORMCLOSE Function](#) - generate <FORM> and </FORM>

[FORMCHECKBOX Function](#) - generates <INPUT TYPE="CHECKBOX">

[FORMHIDDEN Function](#) - generates <INPUT TYPE="HIDDEN">

[FORMIMAGE Function](#) - generates <INPUT TYPE="IMAGE">

[FORMPASSWORD Function](#) - generates <INPUT TYPE="PASSWORD">

[FORMRADIO Function](#) - generates <INPUT TYPE="RADIO">

[FORMSELECTOPEN Function](#), [FORMSELECTCLOSE Function](#) - generate <SELECT> and </SELECT>

[FORMSELETOPTION Function](#) - generates <OPTION>

[FORMTEXT Function](#) - generates <INPUT TYPE="TEXT">

[FORMTEXTAREA Function](#) - generate <TEXTAREA>

[FORMTEXTAREAOPEN Function](#), [FORMTEXTAREACLOSE Function](#) - generate `<TEXTAREA>` and `</TEXTAREA>`

[FORMRESET Function](#) - generates `<INPUT TYPE="RESET">`

[FORMSUBMIT Function](#) - generates `<INPUT TYPE="SUBMIT">`

Table Tags

[TABLEOPEN Function](#), [TABLECLOSE Function](#) - generate `<TABLE>` and `</TABLE>`

[TABLECAPTION Function](#) - generates `<CAPTION>`

[TABLEROWOPEN Function](#), [TABLEROWCLOSE Function](#) - generate `<TR>` and `</TR>`

[TABLEHEADER Function](#) - generates `<TH>`

[TABLEDATA Function](#) - generates `<TD>`

IMG, HR, and A Tags

[HR Function](#), [LINE Function](#) - generate `<HR>`

[IMG Function](#), [IMG2 Function](#) - generate ``

[ANCHOR Function](#), [ANCHOR2 Function](#) - generate `<A>`

[MAPOPEN Function](#), [MAPCLOSE Function](#) - generate `<MAP>` and `</MAP>`

Paragraph Formatting Tags

[HEADER Function](#) - generates heading tags (`<H1>` to `<H6>`)

[PARA Function](#), [PARAGRAPH Function](#) - generate `<P>`

[PRN Functions](#), [PRINT Functions](#) - generate any text that is passed in

[PRN Functions](#), [S Function](#) - generate any text that is passed in; special characters in HTML are escaped

[PREOPEN Function](#), [PRECLOSE Function](#) - generate `<PRE>` and `</PRE>`

[BLOCKQUOTEOPEN Function](#), [BLOCKQUOTECLOSE Function](#) - generate `<BLOCKQUOTE>` and `</BLOCKQUOTE>`

[DIV Function](#) - generates `<DIV>`

[NL Function](#), [BR Function](#) - generate `
`

[NOBR Function](#) - generates `<NOBR>`

[WBR Function](#) - generates `<WBR>`

[PLAINTEXT Function](#) - generates `<PLAINTEXT>`

[ADDRESS Function](#) - generates `<ADDRESS>`

[MAILTO Function](#) - generates `<A>` with the `MAILTO` attribute

[AREA Function](#) - generates `<AREA>`

[BGSOUND Function](#) - generates `<BGSOUND>`

Character Formatting Tags

BASEFONT Function - generates <BASEFONT>

BIG Function - generates <BIG>

BOLD Function - generates

CENTER Function - generates <CENTER> and </CENTER>

CENTEROPEN Function, CENTERCLOSE Function - generate <CENTER> and </CENTER>

CITE Function - generates <CITE>

CODE Function - generates <CODE>

DFN Function - generates <DFN>

EM Function, EMPHASIS Function - generate

FONTOPEN Function, FONTCLOSE Function - generate and

ITALIC Function - generates <I>

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

S Function - generates <S>

SAMPLE Function - generates <SAMP>

SMALL Function - generates <SMALL>

STRIKE Function - generates <STRIKE>

STRONG Function - generates

SUB Function - generates <SUB>

SUP Function - generates <SUP>

TELETYPE Function - generates <TT>

UNDERLINE Function - generates <U>

VARIABLE Function - generates <VAR>

Frame Tags

FRAME Function - generates <FRAME>

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

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

Summary of HTF Subprograms

This table lists the HTF subprograms and briefly describes them.

Table 246-1 HTF Package Subprograms

Subprogram	Description
ADDRESS Function	Generates the <ADDRESS> and </ADDRESS> tags which specify the address, author and signature of a document
ANCHOR Function	Generates the <A> and tags which specify the source or destination of a hypertext link
ANCHOR2 Function	Generates the <A> and tags which specify the source or destination of a hypertext link
APPLETCLOSE Function	Closes the applet invocation with the </APPLET> tag
APPLETOPEN Function	Generates the <APPLET> tag which begins the invocation of a Java applet
AREA Function	Generates the <AREA> tag, which defines a client-side image map
BASE Function	Generates the <BASE> tag which records the URL of the document
BASEFONT Function	Generates the <BASEFONT> tag which specifies the base font size for a Web page
BGSOUND Function	Generates the <BGSOUND> tag which includes audio for a Web page
BIG Function	Generates the <BIG> and </BIG> tags which direct the browser to render the text in a bigger font
BLOCKQUOTECLOSE Function	Generates the </BLOCKQUOTE> tag which mark the end of a section of quoted text
BLOCKQUOTEOPEN Function	Generates the <BLOCKQUOTE> tag, which marks the beginning of a section of quoted text
BODYCLOSE Function	Generates the </BODY> tag which marks the end of a body section of an HTML document
BODYOPEN Function	Generates the <BODY> tag which marks the beginning of the body section of an HTML document
BOLD Function	Generates the and tags which direct the browser to display the text in boldface
BR Function	Generates the tag which begins a new line of text
CENTER Function	Generates the <CENTER> and </CENTER> tags which center a section of text within a Web page
CENTERCLOSE Function	Generates the </CENTER> tag which marks the end of a section of text to center
CENTEROPEN Function	Generates the <CENTER> tag which mark the beginning of a section of text to center
CITE Function	Generates the <CITE> and </CITE> tags which direct the browser to render the text as a citation
CODE Function	Generates the <CODE> and </CODE> tags which direct the browser to render the text in monospace font or however "code" is defined stylistically
COMMENT Function	Generates the comment tags <!-- ctext -->
DFN Function	Generates the <DFN> and </DFN> tags which direct the browser to mark the text as italics or however "definition" is defined stylistically
DIRLISTCLOSE Function	Generates the </DIR> tag which ends a directory list section
DIRLISTOPEN Function	Generates the <DIR> which starts a directory list section

Table 246-1 (Cont.) HTF Package Subprograms

Subprogram	Description
DIV Function	Generates the <DIV> tag which creates document divisions
DLISTCLOSE Function	Generates the </DL> tag which ends a definition list
DLISTDEF Function	Generates the <DD> tag, which inserts definitions of terms
DLISTOPEN Function	Generates the <DL> tag which starts a definition list
DLISTTERM Function	Generates the <DT> tag which defines a term in a definition list <DL>
EM Function	Generates the and tags, which define text to be emphasized
EMPHASIS Function	Generates the and tags, which define text to be emphasized
ESCAPE_SC Function	Replaces characters that have special meaning in HTML with their escape sequences
ESCAPE_URL Function	Replaces characters that have special meaning in HTML and HTTP with their escape sequences
FONTCLOSE Function	Generates the tag which marks the end of a section of text with the specified font characteristics
FONTOPEN Function	Generates the which marks the beginning of section of text with the specified font characteristics
FORMAT_CELL Function	formats column values inside an HTML table using the TABLEDATA Function
FORMCHECKBOX Function	Generates the <INPUT> tag with TYPE="checkbox" which inserts a checkbox element in a form
FORMCLOSE Function	Generates the </FORM> tag which marks the end of a form section in an HTML document
FORMFILE Function	Generates the <INPUT> tag with TYPE="file" which inserts a file form element, and is used for file uploading for a given page
FORMHIDDEN Function	Generates the <INPUT> tag with TYPE="hidden" which inserts a hidden form element
FORMIMAGE Function	Generates the <INPUT> tag with TYPE="image" which creates an image field that the user clicks to submit the form immediately
FORMOPEN Function	Generates the <FORM> tag which marks the beginning of a form section in an HTML document
FORMPASSWORD Function	Generates the <INPUT> tag with TYPE="password" which creates a single-line text entry field
FORMRADIO Function	Generates the <INPUT> tag with TYPE="radio", which creates a radio button on the HTML form
FORMRESET Function	Generates the <INPUT> tag with TYPE="reset" which creates a button that, when selected, resets the form fields to their initial values
FORMSELECTCLOSE Function	Generates the </SELECT> tag which marks the end of a Select form element
FORMSELECTOPEN Function	Generates the </SELECT> tag which marks the beginning of a Select form element

Table 246-1 (Cont.) HTF Package Subprograms

Subprogram	Description
FORMSELECTOPTION Function	Generates the <OPTION> tag which represents one choice in a Select element
FORMSUBMIT Function	Generates the <INPUT> tag with TYPE="submit" which creates a button that, when clicked, submits the form
FORMTEXT Function	Generates the <INPUT> tag with TYPE="text", which creates a field for a single line of text
FORMTEXTAREA Function	Generates the <TEXTAREA> tag, which creates a text field that has no predefined text in the text area
FORMTEXTAREA2 Function	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
FORMTEXTAREACLOSE Function	Generates the </TEXTAREA> tag which ends a text area form element
FORMTEXTAREAOPEN Function	Generates the <TEXTAREA> which marks the beginning of a text area form element
FORMTEXTAREAOPEN2 Function	Generates the <TEXTAREA> which marks the beginning of a text area form element with the ability to specify a wrap style
FRAME Function	Generates the <FRAME> tag which defines the characteristics of a frame created by a <FRAMESET> tag
FRAMESETCLOSE Function	Generates the </FRAMESET> tag which ends a frameset section
FRAMESETOPEN Function	Generates the </FRAMESET> tag which begins a frameset section
HEADCLOSE Function	Generates the </HEAD> tag which marks the end of an HTML document head section
HEADER Function	Generates opening heading tags (<H1> to <H6>) and their corresponding closing tags (</H1> to </H6>)
HEADOPEN Function	Generates the <HEAD> tag which marks the beginning of the HTML document head section
HR Function	Generates the <HR> tag, which generates a line in the HTML document
HTMLCLOSE Function	Generates the </HTML> tag which marks the end of an HTML document
HTMLOPEN Function	Generates the <HTML> tag which marks the beginning of an HTML document
IMG Function	Generates the tag which directs the browser to load an image onto the HTML page
IMG2 Function	Generates the tag which directs the browser to load an image onto the HTML page with the option of specifying values for the USEMAP attribute
ISINDEX Function	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 Function	Generates the <I> and </I> tags which direct the browser to render the text in italics
KBD Function	Generates the <KBD> and </KBD> tags which direct the browser to render the text in monospace font

Table 246-1 (Cont.) HTF Package Subprograms

Subprogram	Description
KEYBOARD Function	Generates the <KBD> and </KBD> tags, which direct the browser to render the text in monospace font
LINE Function	Generates the <HR> tag, which generates a line in the HTML document
LINKREL Function	Generates the <LINK> tag with the REL attribute which delineates the relationship described by the hypertext link from the anchor to the target
LINKREV Function	Generates the <LINK> tag with the REV attribute which delineates the relationship described by the hypertext link from the target to the anchor
LISTHEADER Function	Generates the <LH> and </LH> tags which print an HTML tag at the beginning of the list
LISTINGCLOSE Function	Generates the </LISTING> tags which marks the end of a section of fixed-width text in the body of an HTML page
LISTINGOPEN Function	Generates the <LISTING> tag which marks the beginning of a section of fixed-width text in the body of an HTML page
LISTITEM Function	Generates the tag, which indicates a list item
MAILTO Function	Generates the <A> tag with the HREF set to 'mailto' prepended to the mail address argument
MAPCLOSE Function	Generates the </MAP> tag which marks the end of a set of regions in a client-side image map
MAPOPEN Function	Generates the <MAP> tag which mark the beginning of a set of regions in a client-side image map
MENULISTCLOSE Function	Generates the </MENU> tag which ends a list that presents one line for each item
MENULISTOPEN Function	Generates the <MENU> tag which begins a list that presents one line for each item
META Function	Generates the <META> tag, which embeds meta-information about the document and also specifies values for HTTP headers
NL Function	Generates the tag which begins a new line of text
NOBR Function	Generates the <NOBR> and </NOBR> tags which turn off line-breaking in a section of text
NOFRAMESCLOSE Function	Generates the </NOFRAMES> tag which marks the end of a no-frames section
NOFRAMESOPEN Function	Generates the <NOFRAMES> tag which mark the beginning of a no-frames section
OLISTCLOSE Function	Generates the tag which defines the end of an ordered list
OLISTOPEN Function	Generates the tag which marks the beginning of an ordered list
PARA Function	Generates the <P> tag which indicates that the text that comes after the tag is to be formatted as a paragraph
PARAGRAPH Function	Adds attributes to the <P> tag
PARAM Function	Generates the <PARAM> tag which specifies parameter values for Java applets

Table 246-1 (Cont.) HTF Package Subprograms

Subprogram	Description
PLAINTEXT Function	Generates the <PLAINTEXT> and </PLAINTEXT> tags which direct the browser to render the text they surround in fixed-width type
PRECLOSE Function	Generates the </PRE> tag which marks the end of a section of preformatted text in the body of the HTML page
PREOPEN Function	Generates the <PRE> tag which marks the beginning of a section of preformatted text in the body of the HTML page
PRINT Functions	Generates the specified parameter as a string terminated with the \n newline character
PRN Functions	Generates the specified parameter as a string
S Function	Generates the <S> and </S> tags which direct the browser to render the text they surround in strikethrough type
SAMPLE Function	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 Function	Generates the <SCRIPT> and </SCRIPT> tags which contain a script written in languages such as JavaScript and VBScript
SMALL Function	Generates the <SMALL> and </SMALL> tags, which direct the browser to render the text they surround using a small font
STRIKE Function	Generates the <STRIKE> and </STRIKE> tags which direct the browser to render the text they surround in strikethrough type
STRONG Function	Generates the and tags which direct the browser to render the text they surround in bold or however "strong" is defined stylistically
STYLE Function	Generates the <STYLE> and </STYLE> tags which include a style sheet in a Web page
SUB Function	Generates the _{and} tags which direct the browser to render the text they surround as subscript
SUP Function	Generates the ^{and} tags which direct the browser to render the text they surround as superscript
TABLECAPTION Function	Generates the <CAPTION> and </CAPTION> tags which place a caption in an HTML table
TABLECLOSE Function	Generates the </TABLE> tag which marks the end of an HTML table
TABLEDATA Function	Generates the <TD> and </TD> tags which insert data into a cell of an HTML table
TABLEHEADER Function	Generates the <TH> and </TH> tags which insert a header cell in an HTML table.
TABLEOPEN Function	Generates the <TABLE> tag which marks the beginning of an HTML table
TABLEROWCLOSE Function	Generates the </TR> tag which marks the end of a new row in an HTML table
TABLEROWOPEN Function	Generates the <TR> tag which marks the beginning of a new row in an HTML table

Table 246-1 (Cont.) HTF Package Subprograms

Subprogram	Description
TELETYPE Function	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 Function	Generates the <TITLE> and </TITLE> tags which specify the text to display in the titlebar of the browser window
ULISTCLOSE Function	Generates the tag which marks the end of an unordered list
ULISTOPEN Function	Generates the tag which marks the beginning of an unordered list
UNDERLINE Function	Generates the <U> and </U> tags, which direct the browser to render the text they surround with an underline
VARIABLE Function	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 Function	Generates the <WBR> tag, which inserts a soft line break within a section of NOBR text

ADDRESS Function

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

Syntax

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

Parameters

Table 246-2 ADDRESS Function 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 function generates

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

ANCHOR Function

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

The difference between these subprograms is that the [ANCHOR2 Function](#) provides a target and therefore can be used for a frame.

Syntax

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

Parameters

Table 246-3 ANCHOR Function Parameters

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

Examples

This function 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 Function

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

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

Syntax

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

Parameters

Table 246-4 ANCHOR2 Function Parameters

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

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

APPLETCLOSE Function

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

You must first invoke the a Java applet using [APPLETOPEN Function](#)

Syntax

```
HTF.APPLETCLOSE  
RETURN VARCHAR2;
```

APPLETOPEN Function

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

You close the applet invocation with [APPLETCLOSE Function](#) which generates the </APPLET> tag.

Syntax

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

Parameters

Table 246-5 APPLETOPEN Function Parameters

Parameter	Description
ccode	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.

Table 246-5 (Cont.) APPLETOPEN Function Parameters

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

Examples

This function generates

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

so that, for example,

```
HTF.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 Function](#) function.
- 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 Function

This function 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

```
HTF.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)
RETURN VARCHAR2;
```

Parameters**Table 246-6 AREA Function Parameters**

Parameter	Description
ccords	The value for the <code>COORDS</code> attribute.
cshape	The value for the <code>SHAPE</code> attribute.
chref	The value for the <code>HREF</code> attribute.
cnohref	If the value for this parameter is not <code>NULL</code> , the <code>NOHREF</code> attribute is added to the tag.
ctarget	The value for the <code>TARGET</code> attribute.

Table 246-6 (Cont.) AREA Function Parameters

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

Examples

This function generates

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

BASE Function

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

Syntax

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

Parameters**Table 246-7 BASE Function 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 function generates

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

BASEFONT Function

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

Syntax

```
HTF.BASEFONT (  
    nsize  IN  INTEGER)  
RETURN VARCHAR2;
```

Parameters**Table 246-8 BASEFONT Function Parameters**

Parameter	Description
nsize	The value for the SIZE attribute.

Examples

This function generates

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

BGSOUND Function

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

Syntax

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

Parameters

Table 246-9 BGSOUND Function 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 function generates

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

BIG Function

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

Syntax

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

Parameters

Table 246-10 BIG Function 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 function generates

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

BLOCKQUOTECLOSE Function

This function 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 Function](#).

Syntax

```
HTF.BLOCKQUOTECLOSE  
RETURN VARCHAR2;
```

Examples

This function generates

```
</BLOCKQUOTE>
```

BLOCKQUOTEOPEN Function

This function 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 Function](#).

Syntax

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

Parameters

Table 246-11 BLOCKQUOTEOPEN Function Parameters

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

Examples

This function generates

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

BODYCLOSE Function

This function 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 Function](#).

Syntax

```
HTF.BODYCLOSE  
RETURN VARCHAR2;
```

Examples

This function generates

```
</BODY>
```

BODYOPEN Function

This function 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 Function](#).

Syntax

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

Parameters

Table 246-12 BODYOPEN Function Parameters

Parameter	Description
cbackground	The value for the <code>BACKGROUND</code> 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 function generates

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

so that

```
HTF.BODYOPEN('/img/background.gif')  
RETURN VARCHAR2;
```

generates:

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

BOLD Function

This function generates the `` and `` tags which direct the browser to display the text in boldface.

Syntax

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

Parameters

Table 246-13 BOLD Function 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 function generates

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

BR Function

This function generates the `
` tag which begins a new line of text.

It performs the same operation as the [NL Function](#).

Syntax

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

Parameters

Table 246-14 BR Function Parameters

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

Examples

This function generates

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

CENTER Function

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

Syntax

```
HTF.CENTER (  
    ctext          IN          VARCHAR2)  
    RETURN VARCHAR2;
```

Parameters

Table 246-15 CENTER Parameters

Parameter	Description
ctext	The text that goes between the tags.

Examples

This function generates

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

CENTERCLOSE Function

This function 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 Function](#).

Syntax

```
HTF.CENTERCLOSE  
    RETURN VARCHAR2;
```

Examples

This function generates

```
</CENTER>
```

CENTEROPEN Function

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

You mark the beginning of a section of text to center with the [CENTERCLOSE Function](#).

Syntax

```
HTF.CENTEROPEN  
    RETURN VARCHAR2;
```

Examples

This function generates

<CENTER>

CITE Function

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

Syntax

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

Parameters

Table 246-16 CITE Function Parameters

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

Examples

This function generates

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

CODE Function

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

Syntax

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

Parameters

Table 246-17 CODE Function Parameters

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

Examples

This function generates

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

COMMENT Function

This function generates the comment tags.

Syntax

```
HTF.COMMENT (  
    ctext          IN          VARCHAR2)  
    RETURN VARCHAR2;
```

Parameters

Table 246-18 COMMENT Function Parameters

Parameter	Description
ctext	The comment.

Examples

This function generates

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

DFN Function

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

Syntax

```
HTF.DFN (  
    ctext          IN          VARCHAR2)  
    RETURN VARCHAR2;
```

Parameters

Table 246-19 DFN Function Parameters

Parameter	Description
ctext	The text to render in italics.

Examples

This function generates

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

DIRLISTCLOSE Function

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

Syntax

```
HTF.DIRLISTCLOSE  
RETURN VARCHAR2;
```

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 `` tag directly or invoke the [LISTITEM Function](#) so that the `` tag appears directly after the `</DIR>` tag to define the items as a list.

Examples

This function generates

```
</DIR>
```

Related Topics

- [DIRLISTOPEN Function](#)
This function generates the `<DIR>` which starts a directory list section. You end a directory list section with the DIRLISTCLOSE Function.

DIRLISTOPEN Function

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

Syntax

```
HTF.DIRLISTOPEN  
RETURN VARCHAR2;
```

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 `` tag directly or invoke the [LISTITEM Function](#) so that the `` tag appears directly after the `</DIR>` tag to define the items as a list.

Examples

This function generates

```
<DIR>
```

Related Topics

- [DIRLISTCLOSE Function](#)
This function generates the `</DIR>` tag which ends a directory list section. You start a directory list section with the DIRLISTOPEN Function.

DIV Function

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

Syntax

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

Parameters

Table 246-20 DIV Function Parameters

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

Examples

This function generates

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

DLISTCLOSE Function

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

Syntax

```
HTF.DLISTCLOSE  
RETURN VARCHAR2;
```

Usage Notes

A definition list looks like a glossary: it contains terms and definitions. Terms are inserted using the [DLISTTERM Function](#) and definitions are inserted using the [DLISTDEF Function](#).

Examples

This function generates

```
</DL>
```

Related Topics

- [DLISTOPEN Function](#)
This function generates the `<DL>` tag which starts a definition list. You end a definition list by means of the `DLISTCLOSE` Function.

DLISTDEF Function

This function 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

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

Parameters

Table 246-21 DLISTDEF Function 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 function generates

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

DLISTOPEN Function

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

Syntax

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

Parameters

Table 246-22 DLISTOPEN Function 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 Function](#) and definitions are inserted using the [DLISTDEF Function](#).

Examples

This function generates

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

Related Topics

- [DLISTCLOSE Function](#)

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

DLISTTERM Function

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

Syntax

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

Parameters

Table 246-23 DLISTTERM Function Parameters

Parameter	Description
ctext	The term.
cclear	The value for the <code>CLEAR</code> attribute.
cattributes	The other attributes to be included as-is in the tag.

Examples

This function generates

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

EM Function

This function generates the `` and `` tags, which define text to be emphasized.

It performs the same task as the [EMPHASIS Function](#).

Syntax

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

Parameters

Table 246-24 EM Function Parameters

Parameter	Description
c <code>text</code>	The text to emphasize.
c <code>attributes</code>	The other attributes to be included as-is in the tag.

Examples

This function generates

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

EMPHASIS Function

This function generates the `` and `` tags, which define text to be emphasized.

It performs the same task as the [EM Function](#).

Syntax

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

Parameters

Table 246-25 EMPHASIS Function Parameters

Parameter	Description
c <code>text</code>	The text to emphasize.
c <code>attributes</code>	The other attributes to be included as-is in the tag.

Examples

This function generates

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

ESCAPE_SC Function

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

The following characters are converted:

- `&` to `&`;
- `"` to `"`;
- `<` to `<`;
- `>` to `>`;

This function performs the same operation as HTP. [PRINTS Procedure](#) and HTP. [PS Procedure](#).

Syntax

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

Parameters

Table 246-26 ESCAPE_SC Function Parameters

Parameter	Description
ctest	The text string to convert.

ESCAPE_URL Function

This deprecated function replaces characters that have special meaning in HTML and HTTP with their escape sequences.



Note:

This procedure, deprecated in Release 10g, and provided here only for reasons of backward compatibility, does not comply with the Internet Engineering Task Force (IETF) Request for Comments (RFC) standards of URL encoding. If you need to encode URLs, it is recommended you use the [ESCAPE Function](#) in the [UTL_URL](#) package.

The following characters are converted:

- & to &
- " to "
- < to <
- > to >
- % to &25

Syntax

```
HTF.ESCAPE_URL(  
    p_url          IN          VARCHAR2);
```

Parameters

Table 246-27 ESCAPE_URL Function Parameters

Parameter	Description
p_url	The string to convert.

Fontclose Function

This function generates the `` 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 Function](#).

Syntax

```
HTF.FONTCLOSE  
RETURN VARCHAR2;
```

Examples

This function generates

``

Fontopen Function

This function generates the `` 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 Function](#).

Syntax

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

Parameters

Table 246-28 FONTOPEN Function 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 function generates

``

FORMAT_CELL Function

This function formats column values inside an HTML table using the TABLEDATA Function. It allows for better control over the HTML tables.

Syntax

```
HTF.FORMAT_CELL(  
    columnValue      IN      VARCHAR2  
    format_numbers   IN      VARCHAR2  DEFAULT NULL  
    RETURN VARCHAR2;
```

Parameters

Table 246-29 FORMAT_CELL Function Parameters

Parameter	Description
columnValue	The value that needs to be formatted in an HTML table.
format_numbers	The format that numeric data is displayed in. If the value of this parameter is not NULL, the number fields are right-justified and rounded to two decimal places.

Examples

This function generates

```
<TD >columnValue</TD>
```

Related Topics

- [TABLEDATA Function](#)

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

FORMCHECKBOX Function

This function 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

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

Parameters

Table 246-30 FORMCHECKBOX Function Parameters

Parameter	Description
cname	The value for the NAME attribute.

Table 246-30 (Cont.) FORMCHECKBOX Function Parameters

Parameter	Description
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 function generates

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

FORMCLOSE Function

This function 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 Function](#).

Syntax

```
HTF.FORMCLOSE  
RETURN VARCHAR2;
```

Examples

This function generates

```
</FORM>
```

FORMFILE Function

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

Syntax

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

Parameters

Table 246-31 FORMFILE Function 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 function generates

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

FORMHIDDEN Function

This function 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

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

Parameters

Table 246-32 FORMHIDDEN Function 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 function generates

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

FORMIMAGE Function

This function 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

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


Parameters

Table 246-33 FORMIMAGE Function 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 function generates

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

FORMOPEN Function

This function 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 Function](#).

Syntax

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

Parameters

Table 246-34 FORMOPEN Function Parameters

Parameter	Description
curl	The URL of the Web Request Broker 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 function generates

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

FORMPASSWORD Function

This function 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

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

Parameters

Table 246-35 FORMPASSWORD Function 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 function generates

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

FORMRADIO Function

This function 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

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

Parameters

Table 246-36 FORMRADIO Function 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 function generates

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

FORMRESET Function

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

Syntax

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

Parameters

Table 246-37 FORMRESET Function Parameters

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

Examples

This function generates

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

FORMSELECTCLOSE Function

This function 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 Function](#). The values are inserted using [FORMSELETOPTION Function](#).

Syntax

```
HTF.FORMSELECTCLOSE  
    RETURN VARCHAR2;
```

Examples

This function generates

```
</SELECT>
```

as shown under [Examples](#) of the [FORMSELECTOPEN Function](#).

FORMSELECTOPEN Function

This function generates the `<SELECT>` tags which begins 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 Function](#). The values are inserted using [FORMSELECTOPTION Function](#).

Syntax

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

Parameters

Table 246-38 FORMSELECTOPEN Function 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 function generates

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

so that

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

generates

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

FORMSELECHOPTION Function

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

Syntax

```
HTF.FORMSELECHOPTION (
    cvalue          IN          VARCHAR2,
    cselected       IN          VARCHAR2  DEFAULT NULL,
    cattributes     IN          VARCHAR2  DEFAULT NULL)
RETURN VARCHAR2;
```

Parameters

Table 246-39 FORMSELECHOPTION Function Parameters

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

Examples

This function generates `<OPTION SELECTED cattributes>cvalue` as shown under the Examples section of the [FORMSELECHOPTION Function](#).

FORMSUBMIT Function

This function 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

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

Parameters

Table 246-40 FORMSUBMIT Function Parameters

Parameter	Description
cname	The value for the <code>NAME</code> attribute.

Table 246-40 (Cont.) FORMSUBMIT Function Parameters

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

Examples

This function generates

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

FORMTEXT Function

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

Syntax

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

Parameters

Table 246-41 FORMTEXT Function 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 function generates

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

FORMTEXTAREA Function

This function 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 Function](#) which in addition has the `cwrap` parameter that lets you specify a wrap style.

Syntax

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

Parameters

Table 246-42 FORMTEXTAREA Function 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 function generates

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

FORMTEXTAREA2 Function

This function 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 Function](#) except that in that case you cannot specify a wrap style.

Syntax

```
HTF.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)  
RETURN VARCHAR2;
```

Parameters

Table 246-43 FORMTEXTAREA2 Function Parameters

Parameter	Description
cname	The value for the NAME attribute.

Table 246-43 (Cont.) FORMTEXTAREA2 Function Parameters

Parameter	Description
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 function generates

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

FORMTEXTAREACLOSE Function

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

You open a text area element by means of either [FORMTEXTAREAOPEN Function](#) or [FORMTEXTAREAOPEN2 Function](#).

Syntax

```
HTF.FORMTEXTAREACLOSE
RETURN VARCHAR2;
```

Examples

This function generates

```
</TEXTAREA>
```

FORMTEXTAREAOPEN Function

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

The same operation is performed by the [FORMTEXTAREAOPEN2 Function](#) 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 Function](#).

Syntax

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


Parameters

Table 246-44 FORMTEXTAREAOPEN Function 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 function generates

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

FORMTEXTAREAOPEN2 Function

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

The same operation is performed by the [FORMTEXTAREAOPEN Function](#) 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 Function](#).

Syntax

```
HTF.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)  
RETURN VARCHAR2;
```

Parameters

Table 246-45 FORMTEXTAREAOPEN2 Function 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 function generates

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

FRAME Function

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

Syntax

```
HTF.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)
RETURN VARCHAR2;
```

Parameters

Table 246-46 FRAME Function 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 function generates

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

FRAMESETCLOSE Function

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

You mark the beginning of a frameset section by means of the [FRAMESETOPEN Function](#).

Syntax

```
HTF.FRAMESETCLOSE
RETURN VARCHAR2;
```

Examples

This function generates

```
</FRAMESET>
```

FRAMESETOPEN Function

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

You mark the end of a frameset section by means of the [FRAMESETCLOSE Function](#).

Syntax

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

Parameters

Table 246-47 FRAMESETOPEN Function 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 function generates

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

HEADCLOSE Function

This function 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 Function](#).

Syntax

```
HTF.HEADCLOSE  
RETURN VARCHAR2;
```

Examples

This function generates

```
</HEAD>
```

HEADER Function

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

Syntax

```
HTF.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)  
RETURN VARCHAR2;
```

Parameters

Table 246-48 HEADER Function Parameters

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

```
HTF.header (1,'Overview')  
RETURN VARCHAR2;
```

produces:

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

HEADOPEN Function

This function 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 Function](#).

Syntax

```
HTF.HEADOPEN  
RETURN VARCHAR2;
```

Examples

This function generates

```
<HEAD>
```

HR Function

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

This subprogram performs the same operation as the [LINE Function](#).

Syntax

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

Parameters

Table 246-49 HR Function 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 function generates

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

HTMLCLOSE Function

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

You use the [HTMLOPEN Function](#) to mark the beginning of an HTML document.

Syntax

```
HTF.HTMLCLOSE  
RETURN VARCHAR2;
```

Examples

This function generates

```
</HTML>
```

HTMLOPEN Function

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

You use the [HTMLCLOSE Function](#) to mark the end of the an HTML document.

Syntax

```
HTF.HTMLOPEN  
    RETURN VARCHAR2;
```

Examples

This function generates

```
<HTML>
```

IMG Function

This function generates the `` tag which directs the browser to load an image onto the HTML page.

The [IMG2 Function](#) performs the same operation but additionally uses the `cusemap` parameter.

Syntax

```
HTF.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)  
    RETURN VARCHAR2;
```

Parameters

Table 246-50 IMG Function Parameters

Parameter	Description
<code>curl</code>	The value for the SRC attribute.
<code>calign</code>	The value for the ALIGN attribute.
<code>calt</code>	The value for the ALT attribute which specifies alternative text to display if the browser does not support images.
<code>cismap</code>	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.
<code>cattributes</code>	The other attributes to be included as-is in the tag.

Examples

This function generates

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

IMG2 Function

This function generates the `` tag, which directs the browser to load an image onto the HTML page.

The [IMG Function](#) performs the same operation but does not use the `cusemap` parameter.

Syntax

```
HTF.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)  
RETURN VARCHAR2;
```

Parameters

Table 246-51 IMG2 Function 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.
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 function generates

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

ISINDEX Function

This function 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

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

Parameters

Table 246-52 ISINDEX Function Parameters

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

Examples

This function generates

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

ITALIC Function

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

Syntax

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

Parameters

Table 246-53 ITALIC Function 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 function generates

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

KBD Function

This function 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 Function](#).

Syntax

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

Parameters

Table 246-54 KBD Function 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 function generates

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

KEYBOARD Function

This function 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 Function](#).

Syntax

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

Parameters

Table 246-55 KEYBOARD Function 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 function generates

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

LINE Function

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

This subprogram performs the same operation as the [HR Function](#).

Syntax

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

Parameters

Table 246-56 LINE Function Parameters

Parameter	Description
cclear	The value for the <code>CLEAR</code> attribute.

Table 246-56 (Cont.) LINE Function Parameters

Parameter	Description
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 function generates

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

LINKREL Function

This function 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 Function](#). This tag indicates a relationship between documents but does not create a link. To create a link, use the [ANCHOR Function](#).

Syntax

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

Parameters

Table 246-57 LINKREL Function 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 function generates

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

LINKREV Function

This function 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 Function](#). This tag indicates a relationship between documents, but does not create a link. To create a link, use the [ANCHOR Function](#).

Syntax

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

Parameters

Table 246-58 LINKREV Function Parameters

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

Examples

This function generates

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

LISTHEADER Function

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

Syntax

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

Parameters

Table 246-59 LISTHEADER Function 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 function generates

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

LISTINGCLOSE Function

This function 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 Function](#).

Syntax

```
HTF.LISTINGCLOSE
RETURN VARCHAR2;
```

Examples

This function generates

```
</LISTING>
```

LISTINGOPEN Function

This function 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 Function](#).

Syntax

```
HTF.LISTINGOPEN
RETURN VARCHAR2;
```

Examples

This function generates

```
<LISTING>
```

LISTITEM Function

This function generates the `` tag, which indicates a list item.

Syntax

```
HTF.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)
RETURN VARCHAR2;
```

Parameters

Table 246-60 LISTITEM Function 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 function generates

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

MAILTO Function

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

Syntax

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

Parameters

Table 246-61 MAILTO Function 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 function generates

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

so that

```
HTF.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 Function

This function 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 Function](#).

Syntax

```
HTF.MAPCLOSE  
RETURN VARCHAR2;
```

Examples

This function generates

```
</MAP>
```

MAPOPEN Function

This function 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 Function](#).

Syntax

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

Parameters

Table 246-62 MAPOPEN Function Parameters

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

Examples

This function generates

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

MENULISTCLOSE Function

This function 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 Function](#). The items in the list appear more compact than an unordered list. The [LISTITEM Function](#) defines the list items in a menu list.

Syntax

```
HTF.MENULISTCLOSE  
RETURN VARCHAR2;
```

Examples

This function generates

```
</MENU>
```

MENULISTOPEN Function

This function 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 Function](#). The items in the list appear more compact than an unordered list. The [LISTITEM Function](#) defines the list items in a menu list.

Syntax

```
HTF.MENULISTOPEN  
RETURN VARCHAR2;
```

Examples

This function generates

```
<MENU>
```

META Function

This function 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

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

Parameters

Table 246-63 META Function Parameters

Parameter	Description
chttp_equiv	The value for the <code>CHTTP_EQUIV</code> attribute.
cname	The value for the <code>NAME</code> attribute.
ccontent	The value for the <code>CONTENT</code> attribute.

Examples

This function generates

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

so that

```
HTF.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 Function

This function generates the `
` tag which begins a new line of text.

It performs the same operation as the [BR Function](#).

Syntax

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

Parameters

Table 246-64 NL Function Parameters

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

Examples

This function generates

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

NOBR Function

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

Syntax

```
HTF.NOBR(  
    ctext          IN          VARCHAR2)  
RETURN VARCHAR2;
```

Parameters

Table 246-65 NOBR Function Parameters

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

Examples

This function generates

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

NOFRAMESCLOSE Function

This function 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 Function](#). See also [FRAME Function](#), [FRAMESETOPEN Function](#) and [FRAMESETCLOSE Function](#).

Syntax

```
HTF.NOFRAMESCLOSE  
RETURN VARCHAR2;
```

Examples

This function generates

```
</NOFRAMES>
```

NOFRAMESOPEN Function

This function 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 Function](#). See also [FRAME Function](#), [FRAMESETOPEN Function](#) and [FRAMESETCLOSE Function](#).

Syntax

```
HTF.NOFRAMESOPEN  
RETURN VARCHAR2;
```

Examples

This function generates

```
<NOFRAMES>
```

OLISTCLOSE Function

This function generates the `` 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 Function](#). Numbered items are added using [LISTITEM Function](#).

Syntax

```
HTF.OLISTCLOSE  
RETURN VARCHAR2;
```

Examples

This function generates

OLISTOPEN Function

This function generates the 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 Function](#). Numbered items are added using [LISTITEM Function](#).

Syntax

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

Parameters

Table 246-66 OLISTOPEN Function 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 function generates

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

PARA Function

This function 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 Function](#).

Syntax

```
HTF.PARA  
RETURN VARCHAR2;
```

Examples

This function generates

```
<P>
```

PARAGRAPH Function

You can use this function to add attributes to the <P> tag created by the PARA Function.

Syntax

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

Parameters

Table 246-67 PARAGRAPH Function 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.
cattributes	The other attributes to be included as-is in the tag.

Examples

This function generates

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

Related Topics

- [PARA Function](#)
This function generates the <P> tag which indicates that the text that comes after the tag is to be formatted as a paragraph.

PARAM Function

This function 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 Function](#) to begin the invocation. Use one [PARAM Function](#) for each desired name-value pair, and use [APPLETCLOSE Function](#) to end the applet invocation.

Syntax

```
HTF.PARAM(  
    cname          IN          VARCHAR2  
    cvalue         IN          VARCHAR2)  
RETURN VARCHAR2;
```

Parameters

Table 246-68 PARAM Function Parameters

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

Examples

This function generates

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

PLAINTEXT Function

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

Syntax

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

Parameters

Table 246-69 PLAINTEXT Function 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 function generates

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

PRECLOSE Function

This function 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 Function](#).

Syntax

```
HTF.PRECLOSE  
RETURN VARCHAR2;
```

Examples

This function generates

```
</PRE>
```

PREOPEN Function

This function 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 Function](#).

Syntax

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

Parameters

Table 246-70 PREOPEN Function Parameters

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

Examples

This function generates

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

PRINT Functions

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

The [PRN Functions](#) performs the same operation but does not terminate with a newline character.

Syntax

```
HTF.PRINT (
    cbuf      IN      VARCHAR2)
RETURN VARCHAR2;
```

```
HTF.PRINT (
    dbuf      IN      DATE)
RETURN VARCHAR2;
```

```
HTF.PRINT (
```

```
    nbuf      IN      NUMBER)
RETURN VARCHAR2;
```

Parameters

Table 246-71 PRINT Function 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 `
`. The `\n` character formats the HTML source but it does not affect how the browser renders the HTML source. Use `
` to control how the browser renders the HTML source.
- These functions do not have function equivalents.

PRN Functions

These functions generate the specified parameter as a string.

Unlike the [PRINT Functions](#) the string is not terminated with the `\n` newline character.

Syntax

```
HTF.PRN (
    cbuf      IN      VARCHAR2)
RETURN VARCHAR2;
```

```
HTF.PRN (
    dbuf      IN      DATE)
RETURN VARCHAR2;
```

```
HTF.PRN (
    nbuf      IN      NUMBER)
RETURN VARCHAR2;
```

Parameters

Table 246-72 PRN Function 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 functions do not have function equivalents.

S Function

This function 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 Function](#).

Syntax

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

Parameters

Table 246-73 S Function 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 function generates

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

SAMPLE Function

This function 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

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

Parameters

Table 246-74 SAMPLE Function 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 function generates

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

SCRIPT Function

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

Syntax

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

Parameters

Table 246-75 **SCRIPT Function 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 function generates

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

so that

```
HTF.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 Function

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

Syntax

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


Parameters

Table 246-76 SMALL Function Parameters

Parameter	Description
c <code>text</code>	The text to be rendered in small font.
c <code>attributes</code>	The other attributes to be included as-is in the tag.

Examples

This function generates

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

STRIKE Function

This function 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 Function](#).

Syntax

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

Parameters

Table 246-77 STRIKE Function Parameters

Parameter	Description
c <code>text</code>	The text to be rendered in strikethrough type.
c <code>attributes</code>	The other attributes to be included as-is in the tag.

Examples

This function generates

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

STRONG Function

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

Syntax

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

Parameters

Table 246-78 STRONG Function Parameters

Parameter	Description
c <code>text</code>	The text to be emphasized.
c <code>attributes</code>	The other attributes to be included as-is in the tag.

Examples

This function generates

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

STYLE Function

This function 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

```
HTF.STYLE(  
    cstyle          IN          VARCHAR2)  
    RETURN VARCHAR2;
```

Parameters

Table 246-79 STYLE Function Parameters

Parameter	Description
c <code>style</code>	The style information to include.

Examples

This function generates

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

SUB Function

This function generates the `_{` and `}` tags which direct the browser to render the text they surround as subscript.

Syntax

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

Parameters

Table 246-80 SUB Function 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 function generates

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

SUP Function

This function generates the ^{and} tags which direct the browser to render the text they surround as superscript.

Syntax

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

Parameters

Table 246-81 SUP Function 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 function generates

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

TABLECAPTION Function

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

Syntax

```
HTF.TABLECAPTION(  
    ccaption       IN          VARCHAR2,  
    calign         in          VARCHAR2  DEFAULT NULL,
```

```
    cattributes    IN          VARCHAR2    DEFAULT NULL)
RETURN VARCHAR2;
```

Parameters

Table 246-82 TABLECAPTION Function 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 function generates

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

TABLECLOSE Function

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

To define the beginning of an HTML table, use the [TABLEOPEN Function](#).

Syntax

```
HTF.TABLECLOSE
RETURN VARCHAR2;
```

Examples

This function generates

```
</TABLE>
```

TABLEDATA Function

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

Syntax

```
HTF.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)
RETURN VARCHAR2;
```

Parameters

Table 246-83 TABLEDATA Function 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.
nowrap	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 function generates

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

TABLEHEADER Function

This function 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

```
HTF.TABLEHEADER(  
  cvalue      IN      VARCHAR2  DEFAULT NULL,  
  calign      IN      VARCHAR2  DEFAULT NULL,  
  cdp         IN      VARCHAR2  DEFAULT NULL,  
  nowrap      IN      VARCHAR2  DEFAULT NULL,  
  crowspan    IN      VARCHAR2  DEFAULT NULL,  
  ccolspan    IN      VARCHAR2  DEFAULT NULL,  
  cattributes IN      VARCHAR2  DEFAULT NULL)  
RETURN VARCHAR2;
```

Parameters

Table 246-84 TABLEHEADER Function 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.
nowrap	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 function generates

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

TABLEOPEN Function

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

To define the end of an HTML table, use the [TABLECLOSE Function](#).

Syntax

```
HTF.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)
RETURN VARCHAR2;
```

Parameters

Table 246-85 TABLEOPEN Function 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 function generates

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

TBLEROWCLOSE Function

This function 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 [TBLEROWOPEN Function](#).

Syntax

```
HTF.TBLEROWCLOSE
RETURN VARCHAR2;
```

Examples

This function generates

```
</TABLE>
```

TABLEROWOPEN Function

This function 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 Function](#).

Syntax

```
HTF.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)  
RETURN VARCHAR2;
```

Parameters

Table 246-86 TABLEROWOPEN Function Parameters

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

Examples

This function generates

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

TELETYPE Function

This function 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

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

Parameters

Table 246-87 TELETYPE Function Parameters

Parameter	Description
ctext	The text to render in a fixed width typewriter font.

Table 246-87 (Cont.) TELETYPE Function Parameters

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

Examples

This function generates

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

TITLE Function

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

Syntax

```
HTF.TITLE(  
    ctitle          IN          VARCHAR2)  
    RETURN VARCHAR2;
```

Parameters**Table 246-88 TITLE Function Parameters**

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

Examples

This function generates

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

ULISTCLOSE Function

This function generates the `` 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 Function](#). Add list items with [LISTITEM Function](#).

Syntax

```
HTF.ULISTCLOSE  
    RETURN VARCHAR2;
```

Examples

This function generates

```
</UL>
```


ULISTOPEN Function

This function generates the 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 Function](#). Add list items with [LISTITEM Function](#).

Syntax

```
HTF.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)  
RETURN VARCHAR2;
```

Parameters

Table 246-89 ULISTOPEN Function Parameters

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

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

UNDERLINE Function

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

Syntax

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

Parameters

Table 246-90 UNDERLINE Function Parameters

Parameter	Description
ctext	The text to render with an underline.

Table 246-90 (Cont.) UNDERLINE Function Parameters

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

Examples

This function generates

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

VARIABLE Function

This function 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

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

Parameters**Table 246-91 VARIABLE Function Parameters**

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

Examples

This function generates

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

WBR Function

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

Syntax

```
HTF.WBR  
RETURN VARCHAR2;
```

Examples

This function generates

```
<WBR>
```