

Oracle[®] Application Server

PL/SQL Web Toolkit Quick Reference

Release 4.0.8.2

June 2000

Part No. A60119-05

ORACLE[®]

Oracle Application Server Release 4.0.8.2 PL/SQL Web Toolkit Quick Reference

Part No. A60119-05

Copyright © 1996, 2000, Oracle Corporation. All rights reserved.

Primary Author: Sanjay Singh

Contributors: Sanjay Patil

The Programs (which include both the software and documentation) contain proprietary information of Oracle Corporation; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property law. Reverse engineering, disassembly, or decompilation of the Programs is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Oracle Corporation.

If the Programs are delivered to the U.S. Government or anyone licensing or using the Programs on behalf of the U.S. Government, the following notice is applicable:

Restricted Rights Notice Programs delivered subject to the DOD FAR Supplement are “commercial computer software” and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are “restricted computer software” and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

The programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be licensee's responsibility to take all appropriate fail-safe, back up, redundancy, and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and Oracle Corporation disclaims liability for any damages caused by such use of the Programs.

Oracle is a registered trademark, and the Oracle logo, Oracle8, Oracle8i, SQL*Plus, Net8, Oracle Advanced Security, Oracle Application Server, JDeveloper, Oracle Enterprise Manager, Oracle Internet Directory, Oracle Message Broker SQL*Net, Oracle Wallet Manager, and PL/SQL are trademarks or registered trademarks of Oracle Corporation. Other names may be trademarks of their respective owners.

Contents

The http and htf Packages	1
The owa_content Package	23
The owa_cookie Package	28
The owa_image Package	29
The owa_opt_lock Package	30
The owa_pattern Package	31
The owa_sec Package	35
The owa_text Package	36
The owa_util Package	38

Send Us Your Comments

PL/SQL Web Toolkit Quick Reference Part No. A60119-05

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- What features did you like most about this manual?

If you find any errors or have suggestions for improvement, please indicate the topic, package, and page number below:

Please send your comments to:

- E-mail - oasdocs@us.oracle.com
- FAX - 650-654-6206. Attn: Oracle Application Server Documentation Manger
- Postal Service:

Oracle Corporation
Oracle Application Server Documentation
500 Oracle Parkway, M/S 6op4
Redwood Shores, CA 94065
USA

If you would like a reply, please provide your name, address, and telephone number below:

Thank you for helping us improve our documentation.

The htp and htf Packages

Name	Syntax	Generates>Returns
htp.address	<pre>htp.address(cvalue in varchar2 cnowrap in varchar2 DEFAULT NULL cclear in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.address(cvalue, cnowrap, cclear, cattributes) return varchar2;</pre>	<pre><ADDRESS CLEAR="cclear" NOWRAP cattributes>cvalue </ADDRESS></pre>
htp.anchor	<pre>htp.anchor(curl in varchar2 ctext in varchar2 cname in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.anchor(curl, ctext, cname, cattributes) return varchar2;</pre>	<pre>ctext</pre>
htp.anchor2	<pre>htp.anchor2(curl in varchar2 ctext in varchar2 cname in varchar2 DEFAULT NULL ctargget in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.anchor2(curl, ctext, cname, ctargget, cattributes) return varchar2;</pre>	<pre>ctext</pre>

Name	Syntax	Generates/Returns
http.appletopen	<pre> http.appletopen(ccode in varchar2 cheight in number cwidth in number cattributes in varchar2 DEFAULT NULL); htf.appletopen(ccode, cheight, cwidth, cattributes) return varchar2;</pre>	<pre><APPLET CODE=<i>ccode</i> HEIGHT=<i>cheight</i> WIDTH=<i>cwidth</i> <i>cattributes</i>></pre>
http.appletclose	<pre> http.appletclose; htf.appletclose return varchar2;</pre>	<pre></APPLET></pre>
http.area	<pre> http.area(ccoords in varchar2 cshape in varchar2 DEFAULT NULL chref in varchar2 DEFAULT NULL cnohref in varchar2 DEFAULT NULL ctargt in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.area(ccoords, cshape, chref, cnohref, ctargt, cattributes) return varchar2;</pre>	<pre><AREA COORDS="<i>ccoords</i>" SHAPE="<i>cshape</i>" HREF="<i>chref</i>" NOHREF TARGET="<i>ctargt</i>" <i>cattributes</i>></pre>
http.base	<pre> http.base(ctargt in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.base(ctargt, cattributes) return varchar2;</pre>	<pre><BASE HREF="<i><current URL></i>" TARGET="<i>ctargt</i>" <i>cattributes</i>></pre>
http.basefont	<pre> http.basefont(nsize in integer); htf.basefont(nsize) return varchar2;</pre>	<pre><BASEFONT SIZE="<i>nsize</i>"></pre>

Name	Syntax	Generates/Returns
htp.bgsound	<pre>htp.bgsound(csrc in varchar2 cloop in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.bgsound(csrc, cloop, cattributes) return varchar2;</pre>	<BGSOUND SRC="csrc" LOOP="clloop" cattributes>
htp.big	<pre>htp.big(ctext in varchar2 cattributes in varchar2 DEFAULT NULL); htf.big(ctext, cattributes) return varchar2;</pre>	<BIG cattributes>ctext</BIG>
htp.blockquote Open	<pre>htp.blockquoteOpen(cnowrap in varchar2 DEFAULT NULL cclear in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.blockquoteOpen(cnowrap, cclear, cattributes) return varchar2;</pre>	<BLOCKQUOTE CLEAR="cclear" NOWRAP cattributes>
htp.blockquote Close	<pre>htp.blockquoteClose; htf.blockquoteClose return varchar2;</pre>	</BLOCKQUOTE>
htp.bodyClose	<pre>htp.bodyClose; htf.bodyClose return varchar2;</pre>	</BODY>
htp.bodyOpen	<pre>htp.bodyOpen(cbackground in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.bodyOpen(cbackground, cattributes) return varchar2;</pre>	<BODY background="cbackground" cattributes>
htp.bold	<pre>htp.bold(ctext in varchar2 cattributes in varchar2 DEFAULT NULL); htf.bold(ctext, cattributes) return varchar2;</pre>	<B cattributes>ctext

Name	Syntax	Generates/Returns
http.br	<pre> http.br(cclear in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.br(cclear, cattributes) return varchar2; </pre>	<BR CLEAR="cclear" cattributes>
http.center	<pre> http.center(ctext in varchar2); htf.center(ctext in varchar2) return varchar2; </pre>	<CENTER>ctext</CENTER>
http.centerClose	<pre> http.centerClose; htf.centerClose return varchar2; </pre>	</CENTER>
http.centerOpen	<pre> http.centerOpen; htf.centerOpen return varchar2; </pre>	<CENTER>
http.cite	<pre> http.cite(ctext in varchar2 cattributes in varchar2 DEFAULT NULL); htf.cite(ctext, cattributes) return varchar2; </pre>	<CITE cattributes>ctext</CITE>
http.code	<pre> http.code(ctext in varchar2 cattributes in varchar2 DEFAULT NULL); htf.code(ctext, cattributes) return varchar2; </pre>	<CODE cattributes>ctext</CODE>
http.comment	<pre> http.comment(ctext in varchar2); htf.comment(ctext in varchar2) return varchar2; </pre>	<!-- ctext -->
http.dfn	<pre> http.dfn(ctext in varchar2); htf.dfn(ctext in varchar2) return varchar2; </pre>	<DFN>ctext</DFN>
http.dirlistClose	<pre> http.dirlistClose; htf.dirlistClose return varchar2; </pre>	</DIR>

Name	Syntax	Generates/Returns
http.dirlistOpen	http.dirlistOpen; htf.dirlistOpen return varchar2;	<DIR>
http.div	http.div(calign in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.div(calign, cattributes) return varchar2;	<DIV ALIGN="calign" cattributes>
http.dlistClose	http.dlistClose; htf.dlistClose return varchar2;	</DL>
http.dlistDef	http.dlistDef(ctext in varchar2 DEFAULT NULL cclear in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.dlistDef(ctext, cclear, cattributes) return varchar2;	<DD CLEAR="cclear" cattributes>ctext
http.dlistOpen	http.dlistOpen(cclear in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.dlistOpen(cclear, cattributes) return varchar2;	<DL CLEAR="cclear" cattributes>
http.dlistTerm	http.dlistTerm(ctext in varchar2 DEFAULT NULL cclear in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.dlistTerm(ctext, cclear, cattributes) return varchar2;	<DT CLEAR="cclear" cattributes>ctext
http.download_file	http.download_file(sFileName in varchar2) http.download_file(bCompress in boolean DEFAULT FALSE);	The downloaded file.

Name	Syntax	Generates/Returns
htp.em	<pre>htp.em(ctext in varchar2 cattributes in varchar2 DEFAULT NULL); htf.em(ctext, cattributes) return varchar2;</pre>	<EM <i>cattributes</i> >ctext
htp.emphasis	<pre>htp.emphasis(ctext in varchar2 cattributes in varchar2 DEFAULT NULL); htf.emphasis(ctext, cattributes) return varchar2;</pre>	<EM <i>cattributes</i> >ctext
htf.escape_sc	<pre>htf.escape_sc(ctext in varchar2) return varchar2; htf.escape_sc(ctext in varchar2);</pre>	The converted string.
htf.escape_url	<pre>htf.escape_url(p_url in varchar2) return varchar2;</pre>	The converted string.
htp.fontClose	<pre>htp.fontClose; htf.fontClose return varchar2;</pre>	
htp.fontOpen	<pre>htp.fontOpen(ccolor in varchar2 DEFAULT NULL cface in varchar2 DEFAULT NULL csize in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.fontOpen(ccolor, cface, csize, cattributes) return varchar2;</pre>	<FONT COLOR=" <i>ccolor</i> " FACE=" <i>cface</i> " SIZE=" <i>csize</i> " <i>cattributes</i> >
htf.format_cell	<pre>htf.format_cell (columnValue in varchar2 format_numbers in varchar2 DEFAULT NULL) return varchar2;</pre>	<TD > <i>columnValue</i> </TD>

Name	Syntax	Generates/Returns
http.formCheckbox	<pre> http.formCheckbox(cname in varchar2 cvalue in varchar2 DEFAULT 'on' cchecked in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.formCheckbox(cname, cvalue, cchecked, cattributes) return varchar2; </pre>	<pre> <INPUT TYPE="checkbox" NAME="cname" VALUE="cvalue" CHECKED cattributes> </pre>
http.formClose	<pre> http.formClose; htf.formClose return varchar2; </pre>	</FORM>
http.formFile	<pre> function formFile(cname in varchar2 caccept in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.formFile (cname, caccept, cattributes) return varchar2; </pre>	<pre> <INPUT TYPE="file" NAME="cname" ACCEPT="caccept" cattributes> </pre>
http.formHidden	<pre> http.formHidden(cname in varchar2 cvalue in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.formHidden(cname, cvalue, cattributes) return varchar2; </pre>	<pre> <INPUT TYPE="hidden" NAME="cname" VALUE="cvalue" cattributes> </pre>
http.formImage	<pre> http.formImage(cname in varchar2 csrc in varchar2 calign in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.formImage(cname, csrc, calign, cattributes) return varchar2; </pre>	<pre> <INPUT TYPE="image" NAME="cname" SRC="csrc" ALIGN="calign" cattributes> </pre>

Name	Syntax	Generates/Returns
http.formOpen	<pre> http.formOpen(curl in varchar2 cmethod in varchar2 DEFAULT 'POST' ctargt in varchar2 DEFAULT NULL cencatype in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.formOpen(curl, cmethod, ctargt, cencatype, cattributes) return varchar2; </pre>	<pre> <FORM ACTION="curl" METHOD="cmethod" TARGET="ctargt" ENCTYPE="cencatype" cattributes> </pre>
http.formPassword	<pre> http.formPassword(cname in varchar2 csize in varchar2 cmaxlength in varchar2 DEFAULT NULL cvalue in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.formPassword(cname, csize, cmaxlength, cvalue, cattributes) return varchar2; </pre>	<pre> <INPUT TYPE="password" NAME="cname" SIZE="csize" MAXLENGTH="cmmaxlength" VALUE="cvalue" cattributes> </pre>
http.formRadio	<pre> http.formRadio(cname in varchar2 cvalue in varchar2 cchecked in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.formRadio(cname, cvalue, cchecked, cattributes) return varchar2; </pre>	<pre> <INPUT TYPE="radio" NAME="cname" VALUE="cvalue" CHECKED cattributes> </pre>
http.formReset	<pre> http.formReset(cvalue in varchar2 DEFAULT 'Reset' cattributes in varchar2 DEFAULT NULL); htf.formReset(cvalue, cattributes) return varchar2; </pre>	<pre> <INPUT TYPE="reset" VALUE="cvalue" cattributes> </pre>
http.formSelect Close	<pre> http.formSelectClose; htf.formSelectClose return varchar2; </pre>	<pre> </SELECT> </pre>

Name	Syntax	Generates/Returns
htp.formSelect Open	<pre>htp.formSelectOpen(cname in varchar2 cprompt in varchar2 DEFAULT NULL nsize in integer DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.formSelectOpen(cname, cprompt, nsize, cattributes) return varchar2;</pre>	<pre><i>cprompt</i> <SELECT NAME="<i>cname</i>" SIZE="<i>nsize</i>" <i>cattributes</i>></pre>
htp.formSelect Option	<pre>htp.formSelectOption(cvalue in varchar2 cselected in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.formSelectOption(cvalue, cselected, cattributes) return varchar2;</pre>	<pre><OPTION SELECTED <i>cattributes</i>><i>cvalue</i></pre>
htp.formSubmit	<pre>htp.formSubmit(cname in varchar2 DEFAULT NULL cvalue in varchar2 DEFAULT 'Submit' cattributes in varchar2 DEFAULT NULL); htf.formSubmit(cname, cvalue, cattributes) return varchar2;</pre>	<pre><INPUT TYPE="submit" NAME="<i>cname</i>" VALUE="<i>cvalue</i>" <i>cattributes</i>></pre>
htp.formText	<pre>htp.formText(cname in varchar2 csize in varchar2 DEFAULT NULL cmaxlength in varchar2 DEFAULT NULL cvalue in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.formText(cname, csize, cmaxlength, cvalue, cattributes) return varchar2;</pre>	<pre><INPUT TYPE="text" NAME="<i>cname</i>" SIZE="<i>csize</i>" MAXLENGTH="<i>cmaxlength</i>" VALUE="<i>cvalue</i>" <i>cattributes</i>></pre>

Name	Syntax	Generates/Returns
http.formTextarea	<pre> http.formTextarea(cname in varchar2 nrows in integer ncolumns in integer calign in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.formTextarea(cname, nrows, ncolumns, calign, cattributes) return varchar2;</pre>	<pre> <TEXTAREA NAME="cname" ROWS="nrows" COLS="ncolumns" ALIGN="calign" cattributes></TEXTAREA></pre>
http.formTextarea2	<pre> http.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); htf.formTextarea2(cname, nrows, ncolumns, calign, cwrap, cattributes) return varchar2;</pre>	<pre> <TEXTAREA NAME="cname" ROWS="nrows" COLS="ncolumns" ALIGN="calign" WRAP="cwrap" cattributes></TEXTAREA></pre>
http.formTextarea Close	<pre> http.formTextareaClose; htf.formTextareaClose return varchar2;</pre>	<pre> </TEXTAREA></pre>
http.formTextarea Open	<pre> http.formTextareaOpen(cname in varchar2 nrows in integer ncolumns in integer calign in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.formTextareaOpen(cname, nrows, ncolumns, calign, cattributes) return varchar2;</pre>	<pre> <TEXTAREA NAME="cname" ROWS="nrows" COLS="ncolumns" ALIGN="calign" cattributes></pre>

Name	Syntax	Generates/Returns
http.formTextareaOpen2	<pre>http.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); htf.formTextareaOpen2(cname, nrows, ncolumns, calign, cwrap, cattributes) return varchar2;</pre>	<pre><TEXTAREA NAME="cname" ROWS="nrows" COLS="ncolumns" ALIGN="calign" WRAP = "cwrap" cattributes></pre>
http.frame	<pre>http.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); htf.frame(csrc, cname, cmarginwidth, cmarginheight, cscrolling, cnoresize, cattributes) return varchar2;</pre>	<pre><FRAME SRC="csrc" NAME="cname" MARGINWIDTH="cmarginwidth" MARGINHEIGHT="cmarginheight" SCROLLING="cscrolling" NOESIZE cattributes></pre>
http.framesetClose	<pre>http.framesetClose; htf.framesetClose return varchar2;</pre>	</FRAMESET>
http.framesetOpen	<pre>http.framesetOpen(crows in varchar2 DEFAULT NULL ccols in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.framesetOpen(crows, ccols, cattributes) return varchar2;</pre>	<FRAMESET ROWS="crows" COLS="ccols" cattributes>

Name	Syntax	Generates/Returns
http.get_download_files_list	<pre>http.get_download_files_list(sFileName out varchar2) http.get_download_files_list(bCompress out binary_integer);</pre>	The downloaded file.
http.headClose	<pre>http.headClose; htf.headClose return varchar2;</pre>	</HEAD>
http.headOpen	<pre>http.headOpen; htf.headOpen return varchar2;</pre>	<HEAD>
http.header	<pre>http.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); htf.header(nsize, cheader, calign, cnowrap, cclear, cattributes) return varchar2;</pre>	<Hnsize ALIGN="calign" NOWRAP CLEAR="cclear" cattributes>cheader</Hnsize>
http.hr	<pre>http.hr(cclear in varchar2 DEFAULT NULL csrc in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.hr(cclear, csrc, cattributes) return varchar2;</pre>	<HR CLEAR="cclear" SRC="csrc" cattributes>
http.htmlClose	<pre>http.htmlClose; htf.htmlClose return varchar2;</pre>	</HTML>
http.htmlOpen	<pre>http.htmlOpen; htf.htmlOpen return varchar2;</pre>	<HTML>

Name	Syntax	Generates/Returns
htp.img	<pre>htp.img(curl in varchar2 DEFAULT NULL calign in varchar2 DEFAULT NULL calt in varchar2 DEFAULT NULL cismap in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.img(curl, calign, calt, cismap, cattributes) return varchar2;</pre>	<pre></pre>
htp.img2	<pre>htp.img2(curl in varchar2 DEFAULT NULL calign in varchar2 DEFAULT NULL calt in varchar2 DEFAULT NULL cismap in varchar2 DEFAULT NULL cusemap in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.img2(curl, calign, calt, cismap, cusemap, cattributes) return varchar2;</pre>	<pre></pre>
htp.isindex	<pre>htp.isindex(cprompt in varchar2 DEFAULT NULL curl in varchar2 DEFAULT NULL); htf.isindex(cprompt, curl) return varchar2;</pre>	<pre><ISINDEX PROMPT="cprompt" HREF="curl"></pre>
htp.italic	<pre>htp.italic(ctext in varchar2 cattributes in varchar2 DEFAULT NULL); htf.italic(ctext, cattributes) return varchar2;</pre>	<pre><I cattributes>ctext</I></pre>
htp.kbd	<pre>htp.kbd(ctext in varchar2 cattributes in varchar2 DEFAULT NULL); htf.kbd(ctext, cattributes) return varchar2;</pre>	<pre><KBD cattributes>ctext</KBD></pre>

Name	Syntax	Generates/Returns
http.keyboard	<pre> http.keyboard(ctext in varchar2 cattributes in varchar2 DEFAULT NULL); htf.keyboard(ctext, cattributes) return varchar2; </pre>	<KBD <i>cattributes</i> >ctext</KBD>
http.line	<pre> http.line(cclear in varchar2 DEFAULT NULL csrc in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.line(cclear, csrc, cattributes) return varchar2; </pre>	<HR CLEAR=" <i>cclear</i> " SRC=" <i>csrc</i> " <i>cattributes</i> >
http.linkRel	<pre> http.linkRel(crel in varchar2 curl in varchar2 ctitle in varchar2 DEFAULT NULL); htf.linkRel(crel, curl, ctitle) return varchar2; </pre>	<LINK REL=" <i>crel</i> " HREF=" <i>curl</i> " TITLE=" <i>ctitle</i> ">
http.linkRev	<pre> http.linkRev(crev in varchar2 curl in varchar2 ctitle in varchar2 DEFAULT NULL); htf.linkRev(crev, curl, ctitle) return varchar2; </pre>	<LINK REV=" <i>crev</i> " HREF=" <i>curl</i> " TITLE=" <i>ctitle</i> ">
http.listHeader	<pre> http.listHeader(ctext in varchar2 cattributes in varchar2 DEFAULT NULL); htf.listHeader(ctext, cattributes) return varchar2; </pre>	<LH <i>cattributes</i> >ctext</LH>

Name	Syntax	Generates/Returns
htp.listItem	<pre>htp.listItem(ctext in varchar2 DEFAULT NULL cclear in varchar2 DEFAULT NULL cdingbat in varchar2 DEFAULT NULL csrc in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.listItem(ctext, cclear, cdingbat, csrc, cattributes) return varchar2;</pre>	<pre><LI CLEAR="cclear" DINGBAT="cdingbat" SRC="csrc" cattributes>ctext</pre>
htp.listingClose	<pre>htp.listingClose; htf.listingClose return varchar2;</pre>	</LISTING>
htp.listingOpen	<pre>htp.listingOpen; htf.listingOpen return varchar2;</pre>	<LISTING>
htp.mailto	<pre>htp.mailto(caddress in varchar2 ctext in varchar2 cname in varchar2 cattributes in varchar2 DEFAULT NULL); htf.mailto(caddress, ctext, cname, cattributes) return varchar2;</pre>	<pre>ctext</pre>
htp.mapClose	<pre>htp.mapClose; htf.mapClose return varchar2;</pre>	</MAP>
htp.mapOpen	<pre>htp.mapOpen(cname in varchar2 cattributes in varchar2 DEFAULT NULL); htf.mapOpen(cname, cattributes) return varchar2;</pre>	<MAP NAME="cname" cattributes>
htp.menulistClose	<pre>htp.menulistClose; htf.menulistClose return varchar2;</pre>	</MENU>

Name	Syntax	Generates/Returns
http.menuListOpen	<pre>http.menuListOpen; htf.menuListOpen return varchar2;</pre>	<MENU>
http.meta	<pre>http.meta(chttp_equiv in varchar2 cname in varchar2 ccontent in varchar2); htf.meta(chnhttp_equiv, cname, ccontent) return varchar2;</pre>	<META HTTP-EQUIV="chnhttp_equiv" NAME="cname" CONTENT="ccontent">
http.nl	<pre>http.nl(cclear in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.nl(cclear, cattributes) return varchar2;</pre>	<BR CLEAR="cclear" cattributes>
http.nobr	<pre>http.nobr(ctext in varchar2); htf.nobr(ctext) return varchar2;</pre>	<NOBR>ctext</NOBR>
http.noFramesClose	<pre>http.noFramesClose htf.noFramesClose return varchar2;</pre>	</NOFRAMES>
http.noFramesOpen	<pre>http.noFramesOpen htf.noFramesOpen return varchar2;</pre>	<NOFRAMES>
http.oListClose	<pre>http.oListClose; htf.oListClose return varchar2;</pre>	
http.oListOpen	<pre>http.oListOpen(cclear in varchar2 DEFAULT NULL cwrap in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.oListOpen(cclear, cwrap, cattributes) return varchar2;</pre>	<OL CLEAR="cclear" WRAP="cwrap" cattributes>

Name	Syntax	Generates/Returns
htp.para	htp.para; htf.para return varcahr2;	<P>
htp.paragraph	htp.paragraph(calign in varchar2 DEFAULT NULL cnowrap in varchar2 DEFAULT NULL cclear in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.paragraph(calign, cnowrap, cclear, cattributes) return varchar2;	<P ALIGN="calign" NOWRAP CLEAR="cclear" cattributes>
htp.param	htp.param(cname in varchar2 cvalue in varchar2); htf.param(cname, cvalue) return varchar2;	<PARAM NAME=cname VALUE="cvalue">
htp.plaintext	htp.plaintext(ctext in varchar2 cattributes in varchar2 DEFAULT NULL); htf.plaintext(ctext, cattributes) return varchar2;	<PLAINTEXT cattributes>ctext</PLAINTEXT>
htp.preClose	htp.preClose; htf.preClose return varchar2;	</PRE>
htp.preOpen	htp.preOpen(cclear in varchar2 DEFAULT NULL cwidth in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.preOpen(cclear, cwidth, cattributes) return varchar2;	<PRE CLEAR="cclear" WIDTH="cwidth" cattributes>

Name	Syntax	Generates/Returns
http.print	<pre>http.print(cbuf in varchar2); http.print(dbuf in date); http.print(nbuf in number);</pre>	A string terminated with a newline.
http.prints	<pre>http.prints(ctext in varchar2);</pre>	A string.
http.prn	<pre>http.prn(cbuf in varchar2); http.prn(dbuf in date); http.prn(nbuf in number);</pre>	The specified string, not terminated with a newline.
http.ps	<pre>http.ps(ctext in varchar2);</pre>	A string.
http.s	<pre>http.s(ctext in varchar2 cattributes in varchar2 DEFAULT NULL); htf.s(ctext, cattributes) return varchar2;</pre>	<S <i>cattributes</i> >ctext</S>
http.sample	<pre>http.sample(ctext in varchar2 cattributes in varchar2 DEFAULT NULL); htf.sample(ctext, cattributes) return varchar2;</pre>	<SAMP <i>cattributes</i> >ctext</SAMP>
http.script	<pre>http.script(cscript in varchar2 clanguage in varchar2 DEFAULT NULL); htf.script(cscript, clanguage) return varchar2;</pre>	<SCRIPT LANGUAGE= <i>clanguage</i> >cscript</SCRIPT>
http.small	<pre>http.small(ctext in varchar2 cattributes in varchar2 DEFAULT NULL); htf.small(ctext, cattributes) return varchar2;</pre>	<SMALL <i>cattributes</i> >ctext</SMALL>

Name	Syntax	Generates/Returns
htp.strike	<pre>htp.strike(ctext in varchar2 cattributes in varchar2 DEFAULT NULL); htf.strike(ctext, cattributes) return varchar2;</pre>	<STRIKE <i>cattributes</i> >ctext</STRIKE>
htp.strong	<pre>htp.strong(ctext in varchar2 cattributes in varchar2 DEFAULT NULL); htf.strong(ctext, cattributes) return varchar2;</pre>	<STRONG <i>cattributes</i> >ctext
htp.style	<pre>htp.style(cstyle in varchar2); htf.style(cstyle) return varchar2;</pre>	<STYLE>cstyle</STYLE>
htp.sub	<pre>htp.sub(ctext in varchar2 calign in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.sub(ctext, calign, cattributes) return varchar2;</pre>	<SUB ALIGN="calign" <i>cattributes</i> >ctext</SUB>
htp.sup	<pre>htp.sup(ctext in varchar2 calign in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.sup(ctext, calign, cattributes) return varchar2;</pre>	<SUP ALIGN="calign" <i>cattributes</i> >ctext</SUP>
htp.tableCaption	<pre>htp.tableCaption(ccaption in varchar2 calign in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.tableCaption(ccaption, calign, cattributes) return varchar2;</pre>	<CAPTION ALIGN="calign" <i>cattributes</i> >ccaption </CAPTION>

Name	Syntax	Generates/Returns
http.tableClose	<pre>http.tableClose; htf.tableClose return varchar2;</pre>	</TABLE>
http.tableData	<pre>http.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); htf.tableData(cvalue, calign, cdp, cnowrap, crowspan, ccolspan, cattributes) return varchar2;</pre>	<pre><TD ALIGN="calign" DP="cdp" ROWSPAN="crowspan" COLSPAN="ccolspan" NOWRAP cattributes>cvalue</TD></pre>
http.tableHeader	<pre>http.tableHeader(cvalue in varchar2 DEFAULT NULL calign in varchar2 DEFAULT NULL cdp in varchar2 DEFAULT NULL cnowrap in varchar2 DEFAULT NULL crowspan in varchar2 DEFAULT NULL ccolspan in varchar2 DEFAULT NULL cattributes in varchar2 DEFAULT NULL); htf.tableHeader(cvalue, calign, cdp, cnowrap, crowspan, ccolspan, cattributes) return varchar2;</pre>	<pre><TH ALIGN="calign" DP="cdp" ROWSPAN="crowspan" COLSPAN="ccolspan" NOWRAP cattributes>cvalue</TH></pre>
http.tableOpen	<pre>http.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); htf.tableOpen(cborder, calign, cnowrap, cclear, cattributes) return varchar2;</pre>	<pre><TABLE "cborder" NOWRAP ALIGN="calign" CLEAR="cclear" cattributes></pre>

Name	Syntax	Generates/Returns
htp.tableRowClose	htp.tableRowClose; htf.tableRowClose return varchar2;	</TR>
htp.tableRowOpen	htp.tableRowOpen(calign in varchar2 DEFAULT NULL cvalign in varchar2 DEFAULT NULL cdp in varchar2 DEFAULT NULL nowrap in varchar2 DEFAULT NULL attributes in varchar2 DEFAULT NULL); htf.tableRowOpen(calign, cvalign,cdp, nowrap, attributes) return varchar2;	<TR ALIGN=" <i>calign</i> " VALIGN=" <i>cvalign</i> " DP=" <i>cdp</i> " NOWRAP cattributes>
htp.teletype	htp.teletype(ctext in varchar2 attributes in varchar2 DEFAULT NULL); htf.teletype(ctext, attributes) return varchar2;	<TT cattributes>ctext</TT>
htp.title	htp.title(ctitle in varchar2); htf.title(ctitle) return varchar2;	<TITLE>ctitle</TITLE>
htp.ulistClose	htp.ulistClose; htf.ulistClose return varhar2;	
htp.ulistOpen	htp.ulistOpen(clear in varchar2 DEFAULT NULL cwrap in varchar2 DEFAULT NULL cdingbat in varchar2 DEFAULT NULL csrc in varchar2 DEFAULT NULL attributes in varchar2 DEFAULT NULL htf.ulistOpen(cclear, cwrap, cdingbat, csrc, attributes) return varchar2;	<UL CLEAR=" <i>cclear</i> " WRAP=" <i>cwrap</i> " DINGBAT=" <i>cdingbat</i> " SRC=" <i>csrc</i> " cattributes>

Name	Syntax	Generates/Returns
htp.underline	<pre>htp.underline(ctext in varchar2 cattributes in varchar2 DEFAULT NULL); htf.underline(ctext, cattributes) return varchar2;</pre>	<U <i>cattributes</i> >ctext</U>
htp.variable	<pre>htp.variable(ctext in varchar2 cattributes in varchar2 DEFAULT NULL); htf.variable(ctext, cattributes) return varchar2;</pre>	<VAR <i>cattributes</i> >ctext</VAR>
htp.wbr	<pre>htp.wbr; htf.wbr return wbr;</pre>	<WBR>

The owa_content Package

Name	Syntax	Generates/Returns
owa_content.bigstring_arr data type	type bigstring_arr is table of varchar2 (2000) index by binary_integer;	Not applicable.
owa_content.number_arr data type	number_arr is table of number index by binary_integer;	Not applicable.
owa_content.string_arr data type	string_arr is table of varchar2 (256) index by binary_integer;	Not applicable.
owa_content.delete_attribute procedure	owa_content.delete_attribute(doc_fullname in varchar2, attrib_name in varchar2);	None, but if a document does not exist, a NO_DATA_FOUND exception is raised.
owa_content.delete_document procedure	owa_content.delete_document(doc_fullname in varchar2);	None, but if a document does not exist, a NO_DATA_FOUND exception is raised.
owa_content.delete_documents procedure	owa_content.delete_documents(doc_fullname_list in string_arr);	None, but a NO_DATA_FOUND exception is raised if files do not exist. The delete operation stops at the same element, so make sure all files exist before using this procedure.
owa_content.document_exists function	owa_content.document_exists(doc_fullname in varchar2) return boolean;	Return true if document exists.

Name	Syntax	Generates/Returns
owa_content.get_attribute procedure	<pre>owa_content.get_attribute(doc_fullname in varchar2, attrib_name in varchar2, attrib_type out number, attrib_val in out varchar2);</pre>	None.
owa_content.get_attributes function	<pre>owa_content.get_attributes(doc_fullname in varchar2, attrib_name out string_arr, attrib_type out number_arr, attrib_val out bigstring_arr) return number;</pre>	The number of attributes is given and a NO_DATA_FOUND exception is raised if the document does not exist.
owa_content.get_author function	<pre>owa_content.get_author(doc_fullname in varchar2) return varchar2;</pre>	A document author value is returned and a NO_DATA_FOUND exception is raised when a document does not exist.
owa_content.get_char_set function	<pre>owa_content.char_set(doc_fullname in varchar2) return varchar2;</pre>	A character set value is returned and a NO_DATA_FOUND exception is raised when a document does not exist.
owa_content.get_content_type function	<pre>owa_content.get_content_type(doc_fullname in varchar2) return varchar2;</pre>	A content type value is returned and a NO_DATA_FOUND exception is raised if the document does not exist.
owa_content.get_description function	<pre>owa_content.get_description(doc_fullname in varchar2) return varchar2;</pre>	A description value is returned and a NO_DATA_FOUND exception is raised if the document does not exist.
owa_content.get_encoding function	<pre>owa_content.get_encoding(doc_fullname in varchar2) return varchar2;</pre>	An encoding value is returned and a NO_DATA_FOUND exception is raised if the document does not exist.

Name	Syntax	Generates/Returns
owa_content.get_expiration function	owa_content.get_expiration(doc_fullname in varchar2) return date;	An expiration date value is returned and a NO_DATA_FOUND exception is raised if the document does not exist.
owa_content.get_language function	owa_content.get_language(doc_fullname in varchar2) return varchar2;	A language value is returned and a NO_DATA_FOUND exception is raised if the document does not exist.
owa_content.get_length function	owa_content.get_length(doc_fullname in varchar2) return number;	The size of the document is returned and a NO_DATA_FOUND exception is raised if the document does not exist.
owa_content.get_title function	owa_content.get_title(doc_fullname in varchar2) return varchar2;	The document title is returned and a NO_DATA_FOUND exception is raised if the document does not exist.
owa_content.list_attributes procedure	owa_content.list_attributes(doc_fullname in varchar2, attrib_name out string_arr);	None, but a NO_DATA_FOUND exception is raised if the document does not exist.
owa_content.list_documents function	owa_content.list_documents(doc_list out string_arr) return binary_integer;	The number of documents in the doc_list is returned.
owa_content.list_system_attributes procedure	owa_content.list_system_attributes(doc_fullname in varchar2, attrib_name out string_arr);	None, but a NO_DATA_FOUND exception is raised if the document does not exist.

Name	Syntax	Generates/Returns
owa_content.list_user_attributes procedure	<pre>owa_content.list_user_attributes(doc_fullname in varchar2, attrib_name out string_arr);</pre>	None, but a NO_DATA_FOUND exception is raised if the document does not exist.
owa_content.rename_document procedure	<pre>owa_content.rename_document(old_fullname in varchar2, new_fullname in varchar2);</pre>	A NO_DATA_FOUND exception is raised if the old_fullname does not exist.
owa_content.set_author procedure	<pre>owa_set_author(doc_fullname in varchar2, author_val in varchar2);</pre>	None, but a NO_DATA_FOUND exception is raised if the document does not exist.
owa_content.set_attribute procedure	<pre>owa_content.set_attribute(doc_fullname in varchar2, attrib_name in varchar2, attrib_type in number, attrib_val in varchar2);</pre>	None, but a NO_DATA_FOUND exception is raised if the document does not exist.
owa_content.set_char_set procedure	<pre>owa_set_char_set(doc_fullname in varchar2, char_set_val in varchar2);</pre>	None, but a NO_DATA_FOUND exception is raised if the document does not exist.
owa_content.set_content_type procedure	<pre>owa_set_content_type(doc_fullname in varchar2, content_type_val in varchar2);</pre>	None, but a NO_DATA_FOUND exception is raised if the document does not exist.
owa_content.set_description procedure	<pre>owa_set_description(doc_fullname in varchar2, desc_val in varchar2);</pre>	None, but a NO_DATA_FOUND exception is raised if the document does not exist.

Name	Syntax	Generates/Returns
owa_content.set_encoding procedure	<pre>owa_set_encoding(doc_fullname in varchar2, encode_val in varchar2);</pre>	None, but a NO_DATA_FOUND exception is raised if the document does not exist.
owa_content.set_expiration procedure	<pre>owa_set_expiration(doc_fullname in varchar2, expire_val in date);</pre>	None, but a NO_DATA_FOUND exception is raised if the document does not exist.
owa_content.set_language procedure	<pre>owa_set_expiration(doc_fullname in varchar2, language_val in varchar2);</pre>	None, but a NO_DATA_FOUND exception is raised if the document does not exist.
owa_content.set_title procedure	<pre>owa_set_expiration(doc_fullname in varchar2, title_val in varchar2);</pre>	None, but a NO_DATA_FOUND exception is raised if the document does not exist.

The owa_cookie Package

Name	Syntax	Generates/Returns
owa_cookie.cookie data type	<pre>type cookie is RECORD(name varchar2(4000), vals vc_arr, num_vals integer);</pre>	Not applicable.
owa_cookie.get function	<pre>owa_cookie.get(name in varchar2) return cookie;</pre>	An owa_cookie.cookie data type.
owa_cookie.get_all procedure	<pre>owa_cookie.get_all(names out vc_arr, vals out vc_arr, num_vals out integer);</pre>	Arrays of the names and of the values in the order received, and the count of the combinations.
owa_cookie.remove procedure	<pre>owa_cookie.remove(name in varchar2, val in varchar2, path in varchar2 DEFAULT NULL);</pre>	Set-Cookie: <name>=<value> expires=01-JAN-1990
owa_cookie.send procedure	<pre>owa_cookie.send(name in varchar2, value in varchar2, expires in date DEFAULT NULL, path in varchar2 DEFAULT NULL, domain in varchar2 DEFAULT NULL, secure in varchar2 DEFAULT NULL);</pre>	Set-Cookie: <name>=<value> expires=<expires> path=<path> domain=<domain> secure

The owa_image Package

Name	Syntax	Generates/Returns
owa_image.NULL_POINT package variable	<code>null_point</code> - package variable	Not applicable.
owa_image.point data type	<code>point</code> - data type	Not applicable.
owa_image.get_x function	<code>owa_image.get_x(p in point) return integer;</code>	The X coordinate as an integer.
owa_image.get_y function	<code>owa_image.get_y(p in point) return integer;</code>	The Y coordinate as an integer.

The owa_opt_lock Package

Name	Syntax	Generates/Returns
owa_opt_lock.vcArray data type	owa_opt_lock.vcArray - data type	Not applicable.
owa_opt_lock.checksum function	owa_opt_lock.checksum(p_buff in varchar2) return number; owa_opt_lock.checksum(p_owner in varchar2 p_tname in varchar2 p_rowid in rowid) return number;	A checksum value.
owa_opt_lock.get_rowid function	owa_opt_lock.get_rowid(p_old_values in vcArray) return rowid;	A ROWID.
owa_opt_lock.store_values procedure	owa_opt_lock.store_values(p_owner in varchar2 p_tname in varchar2 p_rowid in rowid);	Stores the column values of the row that you want to update later. The values are stored in hidden HTML form elements.
owa_opt_lock.verify_values function	owa_opt_lock.verify_values(p_old_values in vcArray) return boolean;	TRUE if no other update has been performed; FALSE otherwise.

The owa_pattern Package

Name	Syntax	Generates/Returns
owa_pattern.pattern data type	owa_pattern.pattern - data type	Not applicable.
owa_pattern.amatch function	<pre> owa_pattern.amatch(line in varchar2 from_loc in integer pat in varchar2 flags in varchar2 DEFAULT NULL) return integer; owa_pattern.amatch(line in varchar2 from_loc in integer pat in out pattern flags in varchar2 DEFAULT NULL) return integer; owa_pattern.amatch(line in varchar2 from_loc in integer pat in varchar2 backrefs out owa_text.vc_arr flags in varchar2 DEFAULT NULL) return integer; </pre>	The index of the character after the end of the match, counting from the beginning of <i>line</i> . If there was no match, the function returns 0.

Name	Syntax	Generates/Returns
owa_pattern.amatch function <i>(continued)</i>	<pre> owa_pattern.amatch(line in varchar2 from_loc in integer pat in out pattern backrefs out owa_text.vc_arr flags in varchar2 DEFAULT NULL) return integer; </pre>	The index of the character after the end of the match, counting from the beginning of <i>line</i> . If there was no match, the function returns 0.
owa_pattern.change function and procedure	<pre> /* function */ owa_pattern.change(line in out varchar2 from_str in varchar2 to_str in varchar2 flags in varchar2 DEFAULT NULL) return integer; /* procedure */ owa_pattern.change(line in out varchar2 from_str in varchar2 to_str in varchar2 flags in varchar2 DEFAULT NULL); /* function */ owa_pattern.change(mline in out owa_text.multi_line from_str in varchar2 to_str in varchar2 flags in varchar2 DEFAULT NULL) return integer; </pre>	The number of substitutions made.

Name	Syntax	Generates/Returns
owa_pattern.change function and procedure <i>(continued)</i>	<pre> /* procedure */ owa_pattern.change(mline in out owa_text.multi_line from_str in varchar2 to_str in varchar2 flags in varchar2 DEFAULT NULL); </pre>	The number of substitutions made.
owa_pattern.getpat procedure	owa_pattern.getpat(arg in VARCHAR2, pat in out pattern);	None.
owa_pattern.match function	<pre> owa_pattern.match(line in varchar2 pat in varchar2 flags in varchar2 DEFAULT NULL) return boolean; owa_pattern.match(line in varchar2 pat in out pattern flags in varchar2 DEFAULT NULL) return boolean; owa_pattern.match(line in varchar2 pat in varchar2 backrefs out owa_text.vc_arr flags in varchar2 DEFAULT NULL) return boolean; </pre>	TRUE if a match was found, FALSE otherwise.

Name	Syntax	Generates/Returns
owa_pattern.match function <i>(continued)</i>	<pre> owa_pattern.match(line in varchar2 pat in out pattern backrefs out owa_text.vc_arr flags in varchar2 DEFAULT NULL) return boolean; owa_pattern.match(mline in owa_text.multi_line pat in varchar2 rlist out owa_text.row_list flags in varchar2 DEFAULT NULL) return boolean; owa_pattern.match(mline in owa_text.multi_line pat in out pattern rlist out owa_text.row_list flags in varchar2 DEFAULT NULL) return boolean; </pre>	TRUE if a match was found, FALSE otherwise.

The owa_sec Package

Name	Syntax	Generates/Returns
owa_sec.get_client_hostname function	<code>owa_sec.get_client_hostname return varchar2;</code>	The hostname.
owa_sec.get_client_ip function	<code>owa_sec.get_client_ip return owa_util.ip_address;</code>	The IP address. The owa_util.ip_address data type is a PL/SQL table where the first four elements contain the four numbers of the IP address.
owa_sec.get_password function	<code>owa_sec.get_password return varchar2;</code>	The password.
owa_sec.get_user_id function	<code>owa_sec.get_user_id return varchar2;</code>	The username.
owa_sec.set_authorization procedure	<code>owa_sec.set_authorization(scheme in integer);</code>	Not applicable.
owa_sec.set_protection_realm procedure	<code>owa_sec.set_protection_realm(realm in varchar2);</code>	Not applicable.

The owa_text Package

Name	Syntax	Generates/Returns
owa_text.multi_line data type	<pre>type multi_line is record(rows vc_arr, num_rows integer, partial_row boolean);</pre>	Not applicable.
owa_text.row_list data type	<pre>type row_list is record(rows int_arr, num_rows integer); int_arr is defined as: type int_arr is table of integer index by binary_integer;</pre>	Not applicable.
owa_text.vc_arr data type	<pre>type vc_arr is table of varchar2(32767) index by binary_integer;</pre>	Not applicable.
owa_text.add2multi procedure	<pre>owa_text.add2multi(stream in varchar2 mline in out multi_line continue in boolean DEFAULT TRUE);</pre>	None.
owa_text.new_row_list function and procedure	<pre>/* function */ owa_text.new_row_list return row_list; /* procedure */ owa_text.new_row_list(rlist out row_list);</pre>	The function version returns the new row_list data type.
owa_text.print_multi procedure	<pre>owa_text.print_multi(mline in multi_line);</pre>	The contents of the multi_line.

Name	Syntax	Generates/Returns
owa_text.print_row_list procedure	<code>owa_text.print_row_list(rlist in row_list);</code>	The contents of the row_list.
owa_text.stream2multi procedure	<code>owa_text.stream2multi(stream in varchar2 mline out multi_line);</code>	None.

The owa_util Package

Name	Syntax	Generates/Returns
owa_util.dateType data type	type dateType is table of varchar2(10) index by binary_integer;	Not applicable.
owa_util.ident_arr data type	type ident_arr is table of varchar2(30) index by binary_integer;	Not applicable.
owa_util.ip_address data type	type ip_address is table of integer index by binary_integer;	Not applicable.
owa_util.bind_variables function	<pre>owa_util.bind_variables(theQuery in varchar2 DEFAULT NULL bv1Name in varchar2 DEFAULT NULL bv1Value in varchar2 DEFAULT NULL bv2Name in varchar2 DEFAULT NULL bv2Value in varchar2 DEFAULT NULL bv3Name in varchar2 DEFAULT NULL bv3Value in varchar2 DEFAULT NULL ... bv25Name in varchar2 DEFAULT NULL bv25Value in varchar2 DEFAULT NULL) return integer;</pre>	An integer identifying the opened cursor.
owa_util.calendarprint procedure	<pre>owa_util.calendarprint(p_query in varchar2 p_mf_only in varchar2 DEFAULT 'N'); owa_util.calendarprint(p_cursor in integer p_mf_only in varchar2 DEFAULT 'N');</pre>	A calendar in the form of an HTML table with a visible border.

Name	Syntax	Generates/Returns
owa_util.cellsprint procedure	<pre> owa_util.cellsprint(p_colCnt in integer p_resultTbl in vc_arr p_format_numbers in varchar2 DEFAULT NULL); owa_util.cellsprint(p_theQuery in varchar2 p_max_rows in number DEFAULT 100 p_format_numbers in varchar2 DEFAULT NULL); owa_util.cellsprint(p_theCursor in integer p_max_rows in number DEFAULT 100 p_format_numbers in varchar2 DEFAULT NULL); owa_util.cellsprint(p_theQuery in varchar2 p_max_rows in number DEFAULT 100 p_format_numbers in varchar2 DEFAULT NULL p_skip_rec in number DEFAULT 0 p_more_data out boolean); owa_util.cellsprint(p_theCursor in integer p_max_rows in number DEFAULT 100 p_format_numbers in varchar2 DEFAULT NULL p_skip_rec in number DEFAULT 0 p_more_data out boolean); owa_util.cellsprint(p_theQuery in varchar2 p_max_rows in number DEFAULT 100 p_format_numbers in varchar2 DEFAULT NULL p_recnt out number); </pre>	<pre> <tr><td>QueryResultItem</td><td>QueryResultItem </td></tr> <tr><td>QueryResultItem</td><td>QueryResultItem </td></tr> </pre>

Name	Syntax	Generates/Returns
owa_util.cellsprint procedure (continued)	<pre>owa_util.cellsprint(p_theCursor in integer p_max_rows in number DEFAULT 100 p_format_numbers in varchar2 DEFAULT NULL p_recnt out number);</pre> <pre>owa_util.cellsprint(p_theQuery in varchar2 p_max_rows in number DEFAULT 100 p_format_numbers in varchar2 DEFAULT NULL p_skip_rec in number DEFAULT 0 p_more_data out boolean p_recnt out number);</pre> <pre>owa_util.cellsprint(p_theCursor in integer p_max_rows in number DEFAULT 100 p_format_numbers in varchar2 DEFAULT NULL p_skip_rec in number DEFAULT 0 p_more_data out boolean p_recnt out number);</pre>	<pre><tr><td>QueryResultItem</td><td>QueryResultItem </td></tr> <tr><td>QueryResultItem</td><td>QueryResultItem </td></tr></pre>
owa_util.choose_date procedure	<pre>owa_util.choose_date(p_name in varchar2, p_date in date DEFAULT SYSDATE);</pre>	Three HTML form elements that allow the user to select the day, the month and the year.
owa_util.get_cgi_env function	<pre>owa_util.get_cgi_env(param_name in varchar2) return varchar2;</pre>	The value of the specified CGI environment variable. If the variable is not defined, the function returns NULL.
owa_util.get_owa_service_path function	<pre>owa_util.get_owa_service_path return varchar2;</pre>	A virtual path of the PL/SQL cartridge that is handling the request.

Name	Syntax	Generates/Returns
owa_util.get _procedure function	owa_util.get_procedure return varchar2;	The name of a procedure, including the package name if the procedure is defined in a package.
owa_util.http _header_close procedure	owa_util.http_header_close;	A newline character, which closes the HTTP header.
owa_util.listprint procedure	<pre>owa_util.listprint(p_theQuery in varchar2 p_cname in varchar2 p_nsize in number p_multiple in boolean DEFAULT FALSE); owa_util.listprint(p_theCursor in integer p_cname in varchar2 p_nsize in number p_multiple in boolean DEFAULT FALSE);</pre>	An HTML selection list form element from the output of a SQL query.
owa_util.mime _header procedure	<pre>owa_util.mime_header(ccontent_type in varchar2 DEFAULT 'text/html', bclose_header in boolean DEFAULT TRUE, ccharset in varchar2 DEFAULT NULL);</pre>	Content-type: <content_type>; charset=<charset>
owa_util.print_cgi _env procedure	owa_util.print_cgi_env;	A list in the following format: cgi_env_var_name = value\n
owa_util.redirect _url procedure	<pre>owa_util.redirect_url(curl in varchar2 bclose_header in boolean DEFAULT TRUE);</pre>	Location: <curl>\n\n

Name	Syntax	Generates/Returns
owa_util.showpage procedure	owa_util.showpage;	<p>The output of htp procedure is displayed in SQL*Plus, SQL*DBA, or Oracle Server Manager. For example:</p> <pre>SQL> set serveroutput on SQL> spool gretzky.html SQL> execute hockey.pass("Gretzky") SQL> execute owa_util.showpage SQL> exit</pre> <p>This would generate an HTML page that could be accessed from web browsers.</p>
owa_util.showsource procedure	owa_util.showsource(cname in varchar2);	The source code of the specified function, procedure, or package.
owa_util.signature procedure	owa_util.signature; owa_util.signature(cname in varchar2);	<p>Without a parameter, the procedure generates a line that looks like the following:</p> <pre>This page was produced by the PL/SQL Agent on August 14, 1999 09:30.</pre> <p>With a parameter, the procedure generates a signature line in the HTML document that might look like the following:</p> <pre>This page was produced by the PL/SQL Agent on 8/14/99 09:30</pre> <p>View PL/SQL Source</p>
owa_util.status_line procedure	<pre>owa_util.status_line(nstatus in integer, creason in varchar2 DEFAULT NULL bclose_header in boolean DEFAULT TRUE);</pre>	Status: <nstatus> <creason>\n\n

Name	Syntax	Generates/Returns
owa_util.tablePrint function	<pre>owa_util.tablePrint(ctable in varchar2 cattributes in varchar2 DEFAULT NULL ntable_type in integer DEFAULT HTML_TABLE ccolums in varchar2 DEFAULT '*' cclauses in varchar2 DEFAULT NULL ccol_aliases in varchar2 DEFAULT NULL nrow_min in number DEFAULT 0 nrow_max in number DEFAULT NULL) return boolean;</pre>	A preformatted or HTML table. Returns TRUE if there are more rows beyond the nrow_max requested, FALSE otherwise.
owa_util.todate function	owa_util.todate(p_dateArray in dateType) return date;	A standard DATE.
owa_util.who _called_me procedure	<pre>owa_util.who_called_me(owner out varchar2 name out varchar2 lineno out number caller_t out varchar2);</pre>	Not applicable.

