ExcelDocTypeUtils

Developer API Guide

Jason Bennett

1/15/2012

http://www.jasonsdevelopercorner.com

Table of Contents

ExcelDocTypeUtils API Guide

The ExcelDocTypeUtils API is a PL/SQL API for the ExcelDocumentType. The API was developed in an effort to make the creation of PL/SQL based Excel reports less cumbersome than when using the base ExcelDocumentType object. The ExcelDocTypeUtils package takes an object oriented approach in that it provides developers with a set of types that map to and group functionalities provided by the base ExcelDocumentType object. The remaining sections of this document will provide an explanation of the API and its functionality.

API Constants …

The ExcelDocTypeUtils API provides the following constants:

Constant Name	Value	Description
HREF_INDICATOR	ExcelHRef	Indicates to the API that the data column referenced in the report select statement is URL. Example: ExcelHRef:::#Sheet1!A1:::Sheet1 (<hrefIndicator>:::<target>:::<label>) Or ExcelHref:::http://www.google.com:::Google The API actually provides a function (*createWorksheetLink*) that can be referenced by the select statement and will generate the entire URL indicator string.
HREF_SEP_CHAR	:::	Separator character used to delimit the distinct parts of the ExcelHRef statement. See example above.
HF_PAGE_NUMBER_SINGLE	PNS	Print Header/Footer format indicator used to specify a single page number on a worksheet header or footer. Used in conjunction with the T_WORKSHEET_HF_DATA (header footer definition record) *hf_type* attribute. Example: v_footer_rec.hf_type := ExcelDocTypeUtils.HF_PAGE_NUMBER_SINGLE; Only visible on a printed report.
HF_PAGE_NUMBER_PAGES	PNP	Print Header/Footer format indicator used to specify a page number and total pages combination (i.e. … 1 of 5)

Constant Name	Value	Description
		on a worksheet header or footer. Used in conjunction with the T_WORKSHEET_HF_DATA (header footer definition record) *hf_type* attribute. Example: v_footer_rec.hf_type := ExcelDocTypeUtils.HF_PAGE_NUMBER_PAGES; Only visible on a printed report.
HF_DATE	SDT	Print Header/Footer format indicator used to specify a date on a worksheet header or footer. Used in conjunction with the T_WORKSHEET_HF_DATA (header footer definition record) *hf_type* attribute. Example: v_footer_rec.hf_type := ExcelDocTypeUtils.HF_DATE; Only visible on a printed report.
HF_DATE_TIME	DTT	Print Header/Footer format indicator used to specify the date and time on a worksheet header or footer. Used in conjunction with the T_WORKSHEET_HF_DATA (header footer definition record) *hf_type* attribute. Example: v_footer_rec.hf_type := ExcelDocTypeUtils.HF_DATE_TIME; Only visible on a printed report.

HF_TEXT	TXT	Print Header/Footer format indicator used to specify that custom text is being displayed on a worksheet header or footer. Used in conjunction with the T_WORKSHEET_HF_DATA (header footer definition record) *hf_type* attribute.
		
		Example:
		
		v_header_rec.hf_type := ExcelDocTypeUtils.HF_TEXT;
		v_header_rec.position := ExcelDocTypeUtils.HF_CENTER;
		v_header_rec.text := 'Employee Report';
		
		Only visible on a printed report.

Constant Name	Value	Description
HF_LEFT	LFT	Print Header/Footer position indicator used to specify where to display a header/footer component. Used in conjunction with the T_WORKSHEET_HF_DATA (header footer definition record) *position* attribute. Example: v_header_rec.hf_type := ExcelDocTypeUtils.HF_TEXT; v_header_rec.position := **ExcelDocTypeUtils.HF_LEFT;** v_header_rec.text := 'Employee Report'; Only visible on a printed report.
HF_RIGHT	RHT	Print Header/Footer position indicator used to specify where to display a header/footer component. Used in conjunction with the T_WORKSHEET_HF_DATA (header footer definition record) *position* attribute. Example: v_header_rec.hf_type := ExcelDocTypeUtils.HF_TEXT; v_header_rec.position := **ExcelDocTypeUtils.HF_RIGHT;** v_header_rec.text := 'Employee Report'; Only visible on a printed report.
HF_CENTER	CTR	Print Header/Footer position indicator used to specify where to display a header/footer component. Used in conjunction with the T_WORKSHEET_HF_DATA (header footer definition record) *position* attribute. Example: v_header_rec.hf_type := ExcelDocTypeUtils.HF_TEXT; v_header_rec.position := **ExcelDocTypeUtils.HF_CENTER;** v_header_rec.text := 'Employee Report'; Only visible on a printed report.
HF_FILEPATH	FPT	Print Header/Footer format indicator used to specify that the file access path be displayed on a worksheet

Constant Name	Value	Description

		header or footer. Used in conjunction with the T_WORKSHEET_HF_DATA (header footer definition record) *hf_type* attribute.
		
		Example:
		
		v_header_rec.hf_type := ExcelDocTypeUtils.HF_FILEPATH;
		v_header_rec.position := ExcelDocTypeUtils.HF_CENTER;
		
		Only visible on a printed report.

WS_ORIENT_PORTRAIT	Portrait	Specifies a worksheet print orientation of 'Portait'. Used in conjunction with the T_WORKSHEET_DATA *worksheet_orientation* attribute. Example: v_worksheet_rec.worksheet_orientation := ExcelDocTypeUtils.WS_ORIENT_PORTRAIT; Only visible on a printed report.
WS_ORIENT_LANDSCAPE	Landscape	Specifies a worksheet print orientation of 'Landscape'. Used in conjunction with the T_WORKSHEET_DATA *worksheet_orientation* attribute. Example: v_worksheet_rec.worksheet_orientation := ExcelDocTypeUtils.WS_ORIENT_LANDSCAPE; Only visible on a printed report.

API Types …

The following section describes a collection schema and package level types (records) and collections of those types (record).

Schema Level Types …

The schema level types and collections are required due to the way they are referenced by dynamic code generated by the package internally.

Type Name	Definition	Description
T_ROW	CREATE OR REPLACE TYPE T_ROW AS TABLE OF VARCHAR2(500);	Collection type used as query result row storage that is passed to the base ExcelDocumentType object. Used by the internal package

Type Name	Definition	Description
		function *buildDataSet* (uses dynamic refcursor execution)
RESULT_TABLE	CREATE OR REPLACE TYPE RESULT_TABLE AS TABLE OF T_ROW;	Collection of T_ROW. This object collection is used to pass the result set (rows) generated by a dynamic (user defined) worksheet query to the procedure *createExcelDocument* where the data is transformed into spreadsheet cells.

Package Level Types …

The package level types and collections make it easy for a user to define specific attributes of the an Excel Worksheet (or multiple worksheets) such as styles, headings, etc … This approach also lends itself to extending this API to a GUI presentation. The follow sections cover each of the package types and provide explanations of their attributes.

T_SHEET_TITLE …

The T_SHEET_TITLE type (record) allows the user to create a title row at the top of a worksheet. It has the follow definition/structure:

Attribute	Data Type/Precision	Description
title	VARCHAR2(1000)	Text of the title to be displayed.
cell_span	NUMBER(12)	How many cells to span over the worksheet for the title. Must be less than or equal to the maximum number of columns in the worksheet.
style	VARCHAR2(200)	Name of style definition being applied to the title. (Style definition is explained in another section of this document).

Example:

```
DECLARE
    v_sheet_title      ExcelDocTypeUtils.T_SHEET_TITLE := NULL;
   v_worksheet_rec     ExcelDocTypeUtils.T_WORKSHEET_DATA := NULL;
BEGIN
      -- Define Sheet Title
  v_sheet_title.title     := 'Employee Salary Report';
  v_sheet_title.cell_span  := '3';
  v_sheet_title.style     := 'SheetTitleStyle';
```

```
  v_worksheet_rec.title    := v_sheet_title;
END;
```

T_CONDITION …

The T_CONDITION type is used to define conditional formats to be applied to worksheet cells. It has the following structure:

Attribute	Data Type/Precision	Description
qualifier	VARCHAR2(200)	Value qualifier such as 'Between' or 'Less' or 'Greater'. These qualifiers are Excel specific. Allowable Values: Between, NotBetween, Equal, NotEqual, Greater, Less, GreaterOrEqual, LessOrEqual
value	VARCHAR2(200)	Qualifying value to trigger the conditional format application. Example: Identify text … (used with NO qualifier) 'NOT(ISERROR(SEARCH("**Ba**",RC)))' Or Numeric range (used with 'Between') '10001,1000000'
format_style	VARCHAR2(500)	Style string to indicate special formatting of cell if it meets conditional requirement. This string is unrelated to the style object defined later. Example: 'color:green'

Example 1:

```
DECLARE
   v_condition_rec     ExcelDocTypeUtils.T_CONDITION   := NULL;
BEGIN
  v_condition_rec.qualifier    := 'Less';
  v_condition_rec.value        :=  50;
```

```
  v_condition_rec.format_style := 'color:red’;

END;
```

Example 2:

```
DECLARE
  v_condition_rec     ExcelDocTypeUtils.T_CONDITION   := NULL;
BEGIN
   -- All Names beginning with 'Ba' will be purple.
    v_condition_rec            := NULL;
    v_condition_rec.value      := 'NOT(ISERROR(SEARCH(&amp;quot;Ba&amp;quot;,RC)))';
    v_condition_rec.format_style := 'color:purple';
END;
```

CONDITIONS_TABLE …

The CONDITIONS_TABLE type is a collection (PL/SQL table) of type T_CONDITION. It is used to combine multiple condition types into a group for comparison over a specific range of cells.

Example:

```
DECLARE
  v_condition_rec       ExcelDocTypeUtils.T_CONDITION     := NULL;
  v_condition_array     ExcelDocTypeUtils.CONDITIONS_TABLE :=
ExcelDocTypeUtils.CONDITIONS_TABLE();
BEGIN
   -- Add conditional formating for Salary Ranges ... color code salary amounts
  -- across three different ranges.
  v_condition_rec.qualifier   := 'Between';
  v_condition_rec.value       := '0,5000';
  v_condition_rec.format_style := 'color:red';
  ExcelDocTypeUtils.addConditionType(v_condition_array,v_condition_rec);
  v_condition_rec.qualifier   := 'Between';
  v_condition_rec.value       := '5001,10000';
  v_condition_rec.format_style := 'color:blue';
  ExcelDocTypeUtils.addConditionType(v_condition_array,v_condition_rec);
  v_condition_rec.qualifier   := 'Between';
  v_condition_rec.value       := '10001,1000000';
  v_condition_rec.format_style := 'color:green';
  ExcelDocTypeUtils.addConditionType(v_condition_array,v_condition_rec);
END;
```

T_CONDITIONAL_FORMATS …

The T_CONDITIONAL_FORMATS type is used to apply the collection T_CONDITION records (CONDITIONS_TABLE) across a range of worksheet cells.

Attribute	Data Type/Precision	Description
range	VARCHAR2(200)	Cell range to which the conditional formats apply. The range uses RC format. Example: 'R2C1:R65000C1'; Translation: From Row 2 Column 1 to Row 65000 Column1
conditions	CONDITIONS_TABLE	Collection of T_CONDITIONS records to be applied across the range value.

Example:

```
DECLARE
  v_condition_rec       ExcelDocTypeUtils.T_CONDITION     := NULL;

  v_condition_array     ExcelDocTypeUtils.CONDITIONS_TABLE :=
ExcelDocTypeUtils.CONDITIONS_TABLE();

 v_conditional_format_rec   ExcelDocTypeUtils.T_CONDITIONAL_FORMATS;

BEGIN
  -- Add conditional formating for Salary Ranges ... color code salary amounts
  -- across three different ranges.
  v_condition_rec.qualifier    := 'Between';
  v_condition_rec.value        := '0,5000';
  v_condition_rec.format_style := 'color:red';
  ExcelDocTypeUtils.addConditionType(v_condition_array,v_condition_rec);

  v_condition_rec.qualifier    := 'Between';
  v_condition_rec.value        := '5001,10000';
  v_condition_rec.format_style := 'color:blue';
  ExcelDocTypeUtils.addConditionType(v_condition_array,v_condition_rec);

  v_condition_rec.qualifier    := 'Between';
  v_condition_rec.value        := '10001,1000000';
  v_condition_rec.format_style := 'color:green';
  ExcelDocTypeUtils.addConditionType(v_condition_array,v_condition_rec);

  -- Format range for Column 3 starting at row 2 and going to row 65000 ...
  v_conditional_format_rec.range      := 'R2C3:R65000C3';
  v_conditional_format_rec.conditions := v_condition_array;
```

```
END;
```

CONDITIONAL_FORMATS_TABLE …

The CONDITIONAL_FORMATS_TABLE type is a collection (PL/SQL table) of T_CONDITIONAL_FORMATS. It's purpose is to store all the conditional formatting data for a single worksheet.

EXAMPLE:

```
DECLARE

  v_worksheet_rec     ExcelDocTypeUtils.T_WORKSHEET_DATA := NULL;
  v_condition_rec     ExcelDocTypeUtils.T_CONDITION      := NULL;

  v_condition_array      ExcelDocTypeUtils.CONDITIONS_TABLE :=
ExcelDocTypeUtils.CONDITIONS_TABLE();

 v_conditional_format_rec   ExcelDocTypeUtils.T_CONDITIONAL_FORMATS;

  v_conditional_format_array ExcelDocTypeUtils.CONDITIONAL_FORMATS_TABLE :=
ExcelDocTypeUtils.CONDITIONAL_FORMATS_TABLE();

BEGIN
  -- Add conditional formating for Salary Ranges ... color code salary amounts
  -- across three different ranges.

  v_condition_rec.qualifier    := 'Between';
  v_condition_rec.value        := '0,5000';
  v_condition_rec.format_style := 'color:red';
  ExcelDocTypeUtils.addConditionType(v_condition_array,v_condition_rec);

  v_condition_rec.qualifier    := 'Between';
  v_condition_rec.value        := '5001,10000';
  v_condition_rec.format_style := 'color:blue';
  ExcelDocTypeUtils.addConditionType(v_condition_array,v_condition_rec);

  v_condition_rec.qualifier    := 'Between';
  v_condition_rec.value        := '10001,1000000';
  v_condition_rec.format_style := 'color:green';
  ExcelDocTypeUtils.addConditionType(v_condition_array,v_condition_rec);

  -- Format range for Column 3 starting at row 2 and going to row 65000 ...
  v_conditional_format_rec.range      := 'R2C3:R65000C3';
  v_conditional_format_rec.conditions := v_condition_array;
```

ExcelDocTypeUtils.addConditionalFormatType(v_conditional_format_array,v_conditional_format_rec);

```
   v_worksheet_rec.worksheet_cond_formats := v_conditional_format_array;
END;
```

T_WORKSHEET_HF_DATA …

The T_WORKSHEET_HF_DATA type holds data related to printed worksheet header and footer formatting.

Attribute	Data Type/Precision	Description
hf_type	VARCHAR2(3)	Defines the header/footer component type. Uses the package constants defined at the beginning of this document: HF_PAGE_NUMBER_SINGLE, HF_PAGE_NUMBER_PAGES, HF_DATE, HF_DATE_TIME, HF_TEXT
text	VARCHAR2(200)	Used only in conjunction with the HF_TEXT type. String containing text to be displayed in the header/footer.
position	VARCHAR2(3)	Specifies where to position the header/footer component. Uses the package constants defined at the beginning of this document: HF_LEFT, HF_RIGHT, HF_CENTER
fontsize	VARCHAR2(3)	Used only in conjunction with the HF_TEXT type. Specifies the font size of the text.

Example:

```
DECLARE
    -- Worksheet Print Headers and Footers.  A Header or Footer can have multiple components
  v_header_rec       ExcelDocTypeUtils.T_WORKSHEET_HF_DATA := NULL;
  v_header_rec_array   ExcelDocTypeUtils.WORKSHEET_HF_TABLE  :=
ExcelDocTypeUtils.WORKSHEET_HF_TABLE();

v_footer_rec       ExcelDocTypeUtils.T_WORKSHEET_HF_DATA := NULL;
  v_footer_rec_array   ExcelDocTypeUtils.WORKSHEET_HF_TABLE  :=
ExcelDocTypeUtils.WORKSHEET_HF_TABLE();
```

```
BEGIN
  -- Create Header Footer Elements
  v_header_rec.hf_type  := ExcelDocTypeUtils.HF_DATE_TIME;
  v_header_rec.position := ExcelDocTypeUtils.HF_RIGHT;

  ExcelDocTypeUtils.addHeaderFooterType(v_header_rec_array,v_header_rec);

  v_header_rec := NULL;
  v_header_rec.hf_type  := ExcelDocTypeUtils.HF_TEXT;
  v_header_rec.position := ExcelDocTypeUtils.HF_CENTER;
  v_header_rec.text     := 'Employee Report';
  v_header_rec.fontsize := '12';

  ExcelDocTypeUtils.addHeaderFooterType(v_header_rec_array,v_header_rec);

  v_footer_rec := NULL;
  v_footer_rec.hf_type  := ExcelDocTypeUtils.HF_PAGE_NUMBER_PAGES;
  v_footer_rec.position := ExcelDocTypeUtils.HF_CENTER;


  ExcelDocTypeUtils.addHeaderFooterType(v_footer_rec_array,v_footer_rec);
END;
```

WORKSHEET_HF_TABLE …

The WORKSHEET_HF_TABLE type is a collection (PL/SQL table) of type T_WORKSHEET_HF_DATA. This collection is passed to the T_WORKSHEET_DATA object for application to a specific worksheet. See the example above for how this is populated using the *addHeaderFooterType* procedure.

T_WORKSHEET_HF_MARGINS …

The T_WORKSHEET_HF_MARGINS type defines the margins for a printed worksheet.

Attribute	Data Type/Precision	Description
Zoom	NUMBER(4)	Contains a value specifying the zoom level of a worksheet window. The ExcelDocumentType default value for this attribute is 100 (percent).
PageBreakZoom	NUMBER(4)	Contains a value specifying the zoom level used during page break preview. The ExcelDocumentType default value for this attribute is 60 (percent).
MarginB	NUMBER(4)	Value for bottom margin. The ExcelDocumentType default value

Attribute	Data Type/Precision	Description
		for this attribute is 1 (inch).
MarginT	NUMBER(4)	Value for top margin. The ExcelDocumentType default value for this attribute is 1 (inch).
MarginL	NUMBER(4)	Value for left margin. The ExcelDocumentType default value for this attribute is 0.75 (inch).
MarginR	NUMBER(4)	Value for right margin. The ExcelDocumentType default value for this attribute is 0.75 (inch).

Example:

```
DECLARE
    v_margin_rec  ExcelDocTypeUtils.T_WORKSHEET_HF_MARGINS := NULL;
BEGIN
   v_margin_rec.Zoom := 100;
   v_margin_rec.PageBreakZoom := 50;
   v_margin_rec.MarginT := 2;
  v_margin_rec.MarginB := 2;
  v_margin_rec.MarginL := 0.5;
 v_margin_rec.MarginR := 0.5;

END;
```

T_STYLE_DEF …

The T_STYLE_DEF type is used to define style definitions that can be applied to almost any part of the Excel worksheet (except the conditional formatting style attribute …).

Attribute	Data Type/Precision	Description
p_style_id	VARCHAR2(50)	The style identifier or name used when applying this style to a worksheet object. This value IS CASE SENSITIVE.
p_font	VARCHAR2(50)	Font type such as 'Times New Roman'
p_ffamily	VARCHAR2(50)	Font family such as 'Roman'. Allowable Values: Decorative, Modern, Roman,

Attribute	Data Type/Precision	Description
		Script, Swiss
p_fsize	VARCHAR2(50)	Font size
p_bold	VARCHAR2(1)	Bold text indicator. Valid values are 'Y' and 'N'.
p_italic	VARCHAR2(1)	Italic text indicator. Valid values are 'Y' and 'N'.
p_underline	VARCHAR2(10)	Underline format type value. Enter values such as 'Single' or 'Double'. Standard Excel underline types.
p_text_color	VARCHAR2(50)	Text color indicator. Can be text value such as 'Red', 'Blue', 'Green' or Hex color value.
p_cell_color	VARCHAR2(50)	Cell background color indicator. Can be text value such as 'Red', 'Blue', 'Green' or Hex color value.
p_cell_pattern	VARCHAR2(50)	Cell pattern indicator. Can be text value such as 'Solid'. Allowable Values: None, Solid, Gray75, Gray50, Gray25, Gray125, Gray0625, HorzStripe, VertStripe, ReverseDiagStripe, DiagStripe, DiagCross, ThickDiagCross, ThinHorzStripe, ThinVertStripe, ThinReverseDiagStripe, ThinDiagStripe, ThinHorzCross, ThinDiagCross
p_align_vertical	VARCHAR2(50)	Text vertical alignment value. Allowable values are: Automatic, Bottom, Center,

Attribute	Data Type/Precision	Description
		Distributed, Justify, Top
p_align_horizontal	VARCHAR2(50)	Text horizontal alignment value. Allowable values are: Automatic, Center, Distributed, Justify, Left, Right
p_align_indent	VARCHAR2(50)	Number that specifies the number of indents.
p_wrap_text	VARCHAR2(1)	Text wrap indicator. Valid values are 'Y' and 'N'.
p_rotate_text_deg	VARCHAR2(3)	Specifies the rotation of text within a cell. Allowable ranges are between -90 and 90.
p_number_format	VARCHAR2(100)	Specifies a number format code in the Excel number format syntax. (See: http://office.microsoft.com/en-us/excel/HP051986791033.aspx)
p_custom_xml	VARCHAR2(4000)	Free text field that allows a developer to include custom style XML code. This assumes some familiarity with the syntax of the spreadsheetML "Style" element and its children. Example: Adding a style entry for cell borders is not included in the API. However, custom XML can added to implement this feature … <Borders> <Border ss:Position="Left" ss:LineStyle="Continuous" ss:Weight="3"/> <Border ss:Position="Right" ss:LineStyle="Continuous" ss:Weight="3"/> <Border ss:Position="Top"

Attribute	Data Type/Precision	Description
		ss:LineStyle="Continuous" ss:Weight="3"/> <Border ss:Position="Bottom" ss:LineStyle="Continuous" ss:Weight="3"/> </Borders>

Example:

```
DECLARE

  v_style_def       ExcelDocTypeUtils.T_STYLE_DEF := NULL;
  v_style_array     ExcelDocTypeUtils.STYLE_LIST  := ExcelDocTypeUtils.STYLE_LIST();

BEGIN

   -- Column Header Style
   v_style_def := NULL;

   v_style_def.p_style_id             := 'ColumnHeaderStyle';
   v_style_def.p_text_color           := 'White';
   v_style_def.p_cell_color           := 'DarkBlue';
   v_style_def.p_cell_pattern         := 'Solid';
   v_style_def.p_font                  := 'Times New Roman';
   v_style_def.p_ffamily               := 'Roman';
   v_style_def.p_fsize                 := '10';
   v_style_def.p_bold                  := 'Y';
   v_style_def.p_underline             := 'Single';
   v_style_def.p_align_vertical        := 'Bottom';
   v_style_def.p_rotate_text_deg       := '-66';
   v_style_def.p_custom_xml            := '<Borders>'||
                            '<Border ss:Position="Left" ss:LineStyle="Continuous" ss:Weight="3"/>'||
                            '<Border ss:Position="Right" ss:LineStyle="Continuous" ss:Weight="3"/>'||
                            '<Border ss:Position="Top" ss:LineStyle="Continuous" ss:Weight="3"/>'||
                            '<Border ss:Position="Bottom" ss:LineStyle="Continuous" ss:Weight="3"/>'||
                          '</Borders>';

    ExcelDocTypeUtils.addStyleType(v_style_array,v_style_def);

END;
```

STYLE_LIST …

The STYLE_LIST type is a collection (PL/SQL table) of T_STYLE_DEF. The collection is passed as a parameter to the *createExcelDocument* procedure. Styles are at the document level.

T_WORKSHEET_DATA …

The T_WORKSHEET_DATA type contains all of the components required to create an Excel Report worksheet. This structure is used by the *createExcelDocument* procedure to construct a worksheet.

The T_WORKSHEET_DATA type makes use of delimited lists for passing such values as column header names, specifying data types for columns, specifying column widths, associated styles with columns, etc. This list approach provides a less complicated way a passing a considerable of data to the report processing engine using a minimal amount of code for the report developer.

Attribute	Data Type/Precision	Description
query	VARCHAR2(4000)	Holds the SQL statement that will populate the worksheet with data.
title	T_SHEET_TITLE	Sheet title that will be displayed at the top of the worksheet.
worksheet_name	VARCHAR2(50)	Name of the worksheet. This is the name/label that is displayed on the worksheet tab at the bottom of the worksheet. If no name is given, then sheet name will default to Sheet(n). n represents the sheet number.
worksheet_header	WORKSHEET_HF_TABLE	Collection of T_WORKSHEET_HF_DATA containing worksheet print header information.
worksheet_footer	WORKSHEET_HF_TABLE	Collection of T_WORKSHEET_HF_DATA containing worksheet print footer information.
worksheet_page_margins	T_WORKSHEET_HF_MARGINS	Contains worksheet printed page margin information.
worksheet_cond_formats	CONDITIONAL_FORMATS_TABLE	Collection containing conditional formatting data for the worksheet.
worksheet_list_delimiter	VARCHAR2(10)	Specifies the delimiter used to delimit items in the record item lists used by this object. The default delimiter is ‘,’, but it can be changed if that character is a valid character in a list item.
worksheet_orientation	VARCHAR2(12)	Indicates WS print orientation. Valid values are 'Portait' and 'Landscape'.
worksheet_show_gridlines	BOOLEAN	Indicates whether or not the gridlines should be displayed in the worksheet. The default option is 'TRUE'.

Attribute	Data Type/Precision	Description
col_header_freeze	BOOLEAN	Indicates whether or not the column header row will be frozen. If the column header row is frozen, then it stays in place while the data rows are scrolling.
col_header_repeat	BOOLEAN	Indicates whether or not to repeat the header row column headers at the top of each printed page of the printed report.
col_firstcol_freeze	BOOLEAN	Indicates whether or not the first column of the worksheet will be frozen. This allows the spreadsheet to scroll left and right while the first column remains frozen.
col_count	NUMBER(3)	Specifies the number of columns that will displayed in the worksheet. This should match the number of columns in the SQL statement that generates data for the worksheet. This parameter is optional.
col_width_list	VARCHAR2(500)	Delimited list specifying a numeric width for each worksheet column. If the **worksheet_list_delimiter** attribute is set, then this list item must use the specified delimiter. Otherwise, use a comma. If we don't want to specify a value for a particular item, the empty list item must still be included. Example: If a worksheet has a col_count of 5 and we want to include a value for every column, the list would look like this: 45,45,2,6,56 (or 45:45:2:6:56 if the delimiter was set to ':'). If we don't want to include a value for the fourth column, the list would look like this: 45,45,2,,56
col_caption	VARCHAR2(2000)	Delimited list specifying a column header caption for each worksheet column. If the

Attribute	Data Type/Precision	Description
		worksheet_list_delimiter attribute is set, then this list item must use the specified delimiter. Otherwise, use a comma. If we don't want to specify a value for a particular item, the empty list item must still be included. Example: If a worksheet has a col_count of 5 and we want to include a value for every column, the list would look like this: blah,blah,blah,blah,blah (or blah:blah:blah:blah:blah if the delimiter was set to ':'). If we don't want to include a value for the fourth column, the list would look like this: blah,blah,blah,,blah
col_header_list	VARCHAR2(2000)	Delimited list specifying a column header for each worksheet column. If the **worksheet_list_delimiter** attribute is set, then this list item must use the specified delimiter. Otherwise, use a comma. If we don't want to specify a value for a particular item, the empty list item must still be included. Example: If a worksheet has a col_count of 5 and we want to include a value for every column, the list would look like this: blah,blah,blah,blah,blah (or blah:blah:blah:blah:blah if the delimiter was set to ':'). If we don't want to include a value for the fourth column, the list would look like this: blah,blah,blah,,blah
col_header_style_list	VARCHAR2(2000)	Delimited list specifying a style for each worksheet column header. The style names must be valid, or Excel will not process the generated

Attribute	Data Type/Precision	Description
		spread sheet. If the **worksheet_list_delimiter** attribute is set, then this list item must use the specified delimiter. Otherwise, use a comma. If we don't want to specify a value for a particular item, the empty list item must still be included. Example: If a worksheet has a col_count of 5 and we want to include a value for every column, the list would look like this: ColHdrStyle, ColHdrStyle , ColHdrStyle, ColHdrStyle, ColHdrStyle (or ColHdrStyle : ColHdrStyle : ColHdrStyle : ColHdrStyle : ColHdrStyle if the delimiter was set to ':'). If we don't want to include a value for the fourth column, the list would look like this: ColHdrStyle, ColHdrStyle, ColHdrStyle,, ColHdrStyle
col_datatype_list	VARCHAR2(4000)	Delimited list specifying a Data Type for each worksheet column. The default is String. The style names must be valid, or Excel will not process the generated spread sheet. If the **worksheet_list_delimiter** attribute is set, then this list item must use the specified delimiter. Otherwise, use a comma. If we don't want to specify a value for a particular item, the empty list item must still be included. **Valid Data Types are:** Number, Bin.Hex, Boolean,

Attribute	Data Type/Precision	Description
		DateTime, Float, Integer, I1, I2, I8, Int, String, UI1, UI2, UI4, UI8 Example: If a worksheet has a col_count of 5 and we want to include a value for every column, the list would look like this: String, String , Number, Number, Number (or String: String:Number:Number:Number if the delimiter was set to ':'). If we don't want to include a value for the fourth column, the list would look like this: String, String , Number, , Number
col_style_list	VARCHAR2(5000)	Delimited list specifying a style for each worksheet column. The style names must be valid, or Excel will not process the generated spread sheet. If the **worksheet_list_delimiter** attribute is set, then this list item must use the specified delimiter. Otherwise, use a comma. If we don't want to specify a value for a particular item, the empty list item must still be included. Example: If a worksheet has a col_count of 5 and we want to include a value for

Attribute	Data Type/Precision	Description
		every column, the list would look like this: ColStyle, ColStyle , ColStyle, ColStyle, ColStyle (or ColStyle : ColStyle : ColStyle : ColStyle : ColStyle if the delimiter was set to ':'). If we don't want to include a value for the fourth column, the list would look like this: ColStyle, ColStyle, ColStyle,, ColStyle
col_formula_list	VARCHAR2(4000)	Delimited list specifying a formula or function type for the worksheet formula row (last row of worksheet if specified). **Valid Formulas/Functions:** Avg, Count, CountNums, Max, Min, Product, StdDev, StdDevP, Sum, Var, VarP If the **worksheet_list_delimiter** attribute is set, then this list item must use the specified delimiter. Otherwise, use a comma. If we don't want to specify a value for a particular item, the empty list item must still be included. Example: If a worksheet has a col_count of 5 and we want to include a value for every column, the list would look like this: Avg,Avg,Sum,Sum,Sum

Attribute	Data Type/Precision	Description
		(or Avg:Avg:Sum:Sum:Sum if the delimiter was set to ':'). If we don't want to include a value for the fourth column, the list would look like this: Avg,Avg,Sum,,Sum
col_formula_style_list	VARCHAR2(4000)	Delimited list specifying a style for each worksheet formula column. In case we want to add specific formatting, etc. The style names must be valid, or Excel will not process the generated spread sheet. If the **worksheet_list_delimiter** attribute is set, then this list item must use the specified delimiter. Otherwise, use a comma. If we don't want to specify a value for a particular item, the empty list item must still be included. Example: If a worksheet has a col_count of 5 and we want to include a value for every column, the list would look like this: ColStyle, ColStyle , ColStyle, ColStyle, ColStyle (or ColStyle : ColStyle : ColStyle : ColStyle : ColStyle if the delimiter was set to ':'). If we don't want to include a value for the fourth column, the list would look like this: ColStyle, ColStyle, ColStyle,, ColStyle

Example:

DECLARE

```
    v_sql_salary       VARCHAR2(200) := 'SELECT  '||
                        'ExcelDocTypeUtils.createWorksheetLink(''Hiredate'',last_name),'||
                       'first_name,'||
                       'salary FROM hr.employees ORDER BY last_name,first_name';

  v_worksheet_rec    ExcelDocTypeUtils.T_WORKSHEET_DATA := NULL;
  v_worksheet_array   ExcelDocTypeUtils.WORKSHEET_TABLE  :=
ExcelDocTypeUtils.WORKSHEET_TABLE();

  v_sheet_title      ExcelDocTypeUtils.T_SHEET_TITLE := NULL;

  -- Objects for Defining Document Styles (Optional)

  v_style_def        ExcelDocTypeUtils.T_STYLE_DEF := NULL;
  v_style_array      ExcelDocTypeUtils.STYLE_LIST  := ExcelDocTypeUtils.STYLE_LIST();

  -- Object for Defining Conditional Formating (Optional)

  v_condition_rec        ExcelDocTypeUtils.T_CONDITION      := NULL;
  v_condition_array      ExcelDocTypeUtils.CONDITIONS_TABLE :=
ExcelDocTypeUtils.CONDITIONS_TABLE();

  -- Conditions are applied to a range of cells ... there can be more than grouping of format conditions
per worksheet.
  v_conditional_format_rec   ExcelDocTypeUtils.T_CONDITIONAL_FORMATS;
  v_conditional_format_array ExcelDocTypeUtils.CONDITIONAL_FORMATS_TABLE :=
ExcelDocTypeUtils.CONDITIONAL_FORMATS_TABLE();

  -- Worksheet Print Headers and Footers.  A Header or Footer can have multiple components
  v_header_rec        ExcelDocTypeUtils.T_WORKSHEET_HF_DATA := NULL;
  v_header_rec_array   ExcelDocTypeUtils.WORKSHEET_HF_TABLE  :=
ExcelDocTypeUtils.WORKSHEET_HF_TABLE();

  v_footer_rec        ExcelDocTypeUtils.T_WORKSHEET_HF_DATA := NULL;
  v_footer_rec_array   ExcelDocTypeUtils.WORKSHEET_HF_TABLE  :=
ExcelDocTypeUtils.WORKSHEET_HF_TABLE();

BEGIN

 -- Define Styles (Optional)
   v_style_def.p_style_id     := 'LastnameStyle';
   v_style_def.p_text_color   := 'Red';
```

```
ExcelDocTypeUtils.addStyleType(v_style_array,v_style_def);

v_style_def := NULL;
v_style_def.p_style_id        := 'SheetTitleStyle';
v_style_def.p_align_horizontal  := 'Center';
v_style_def.p_bold            := 'Y';
v_style_def.p_text_color      := 'Green';

ExcelDocTypeUtils.addStyleType(v_style_array,v_style_def);

v_style_def := NULL;
v_style_def.p_style_id    := 'FirstnameStyle';
v_style_def.p_text_color  := 'Blue';

ExcelDocTypeUtils.addStyleType(v_style_array,v_style_def);


-- Heading Row Style
v_style_def := NULL;
v_style_def.p_style_id        := 'HeadingRowStyle';
v_style_def.p_text_color      := 'Black';
v_style_def.p_font            := 'Times New Roman';
v_style_def.p_ffamily         := 'Roman';
v_style_def.p_fsize           := '10';
v_style_def.p_bold            := 'Y';
v_style_def.p_underline       := 'Single';
v_style_def.p_align_vertical  := 'Bottom';
v_style_def.p_rotate_text_deg  := '45';

ExcelDocTypeUtils.addStyleType(v_style_array,v_style_def);

-- Style that includes custom borders around numbers
v_style_def := NULL;
v_style_def.p_style_id       := 'NumberStyle';
v_style_def.p_number_format   := '$###,###,###.00';
v_style_def.p_align_horizontal := 'Right';
v_style_def.p_custom_xml       := '<Borders>'||
                                  '<Border ss:Position="Left" ss:LineStyle="Continuous" ss:Weight="3"/>'||
                                  '<Border ss:Position="Right" ss:LineStyle="Continuous" ss:Weight="3"/>'||
                                  '<Border ss:Position="Top" ss:LineStyle="Continuous" ss:Weight="3"/>'||
                                  '<Border ss:Position="Bottom" ss:LineStyle="Continuous" ss:Weight="3"/>'||
```

```
                        '</Borders>';

 ExcelDocTypeUtils.addStyleType(v_style_array,v_style_def);

 -- Define Sheet Title
 v_sheet_title.title     := 'Employee Salary Report';

 -- Must Less be than or Equal to the max number of columns on the worksheet.
 v_sheet_title.cell_span  := '3';
 v_sheet_title.style     := 'SheetTitleStyle';

 v_worksheet_rec.title    := v_sheet_title;

 -- Add conditional formating for Salary Ranges ... color code salary amounts
 -- across three different ranges.

 v_condition_rec.qualifier    := 'Between';
 v_condition_rec.value       := '0,5000';
 v_condition_rec.format_style := 'color:red';

 ExcelDocTypeUtils.addConditionType(v_condition_array,v_condition_rec);

 v_condition_rec.qualifier    := 'Between';
 v_condition_rec.value       := '5001,10000';
 v_condition_rec.format_style := 'color:blue';

 ExcelDocTypeUtils.addConditionType(v_condition_array,v_condition_rec);


 v_condition_rec.qualifier    := 'Between';
 v_condition_rec.value       := '10001,1000000';
 v_condition_rec.format_style := 'color:green';

 ExcelDocTypeUtils.addConditionType(v_condition_array,v_condition_rec);

 -- Format range for Column 3 starting at row 2 and going to row 65000 ...
 v_conditional_format_rec.range      := 'R2C3:R65000C3';
 v_conditional_format_rec.conditions := v_condition_array;


ExcelDocTypeUtils.addConditionalFormatType(v_conditional_format_array,v_conditional_format_rec);
```

```
  -- All Names beginning with 'Ba' will be purple.

  v_condition_array          := ExcelDocTypeUtils.CONDITIONS_TABLE();

  v_condition_rec            := NULL;
  v_condition_rec.value      := 'NOT(ISERROR(SEARCH(&quot;Ba&quot;,RC)))';
  v_condition_rec.format_style := 'color:purple';

  ExcelDocTypeUtils.addConditionType(v_condition_array,v_condition_rec);

  -- Format range for Column 1 starting at row 2 and going to row 65000 ...
  v_conditional_format_rec          := NULL;
  v_conditional_format_rec.range    := 'R2C1:R65000C1';
  v_conditional_format_rec.conditions := v_condition_array;


ExcelDocTypeUtils.addConditionalFormatType(v_conditional_format_array,v_conditional_format_rec);

  v_worksheet_rec.worksheet_cond_formats := v_conditional_format_array;


  -- Create Header Footer Elements
  v_header_rec.hf_type  := ExcelDocTypeUtils.HF_DATE_TIME;
  v_header_rec.position := ExcelDocTypeUtils.HF_RIGHT;

  ExcelDocTypeUtils.addHeaderFooterType(v_header_rec_array,v_header_rec);

  v_header_rec := NULL;
  v_header_rec.hf_type  := ExcelDocTypeUtils.HF_TEXT;
  v_header_rec.position := ExcelDocTypeUtils.HF_CENTER;
  v_header_rec.text     := 'Employee Report';
  v_header_rec.fontsize := '12';

  ExcelDocTypeUtils.addHeaderFooterType(v_header_rec_array,v_header_rec);

  v_footer_rec := NULL;
  v_footer_rec.hf_type  := ExcelDocTypeUtils.HF_PAGE_NUMBER_PAGES;
  v_footer_rec.position := ExcelDocTypeUtils.HF_CENTER;


  ExcelDocTypeUtils.addHeaderFooterType(v_footer_rec_array,v_footer_rec);
```

```
  -- Salary

  v_worksheet_rec.worksheet_header := v_header_rec_array;
  v_worksheet_rec.worksheet_footer := v_footer_rec_array;

  -- !!! SETTING THE LIST ITEM DELIMITER TO A ':' INSTEAD OF THE DEFAULT ',' !!!
  v_worksheet_rec.worksheet_list_delimiter := ':';

  v_worksheet_rec.query             := v_sql_salary;
  v_worksheet_rec.worksheet_name    := 'Salaries';
  v_worksheet_rec.col_count         := 3;
  v_worksheet_rec.col_width_list    := '25:20:15';
  v_worksheet_rec.col_header_freeze := TRUE;
  v_worksheet_rec.col_header_list   := 'Lastname:Firstname:Salary';
  v_worksheet_rec.col_header_style_list := 'HeadingRowStyle:HeadingRowStyle:HeadingRowStyle';
  v_worksheet_rec.col_datatype_list := 'String:String:Number';
  v_worksheet_rec.col_style_list    := 'LastnameStyle:FirstnameStyle:NumberStyle';

  -- Add a SUM formula to the last column. Valid simply formulas include SUM, AVERAGE, COUNT, MIN,
MAX
  v_worksheet_rec.col_formula_list  := '::SUM';

  ExcelDocTypeUtils.addWorksheetType(v_worksheet_array,v_worksheet_rec);

END;
```

WORKSHEET_TABLE …

The worksheet table type is a collection (PL/SQL table) of type T_WORKSHEET_DATA. The collection is passed to the ExcelDocTypeUtils procedure *createExcelDocument*, where all of the worksheet entries are processed into a complete Excel Workbook.

API Procedures and Functions …

This section details the public PL/SQL functions and procedures provided with the ExcelDocTypeUtils package.

Procedure/Function	Specification	Description
addStyleType	PROCEDURE addStyleType(p_style_array IN OUT NOCOPY STYLE_LIST, p_style_rec T_STYLE_DEF)	Convenience procedure that adds a T_STYLE_DEF type to a STYLE_LIST collection. The STYLE_LIST collection is passed as an IN/OUT

Procedure/Function	Specification	Description
		parameter.
addWorksheetType	PROCEDURE addWorksheetType(p_worksheet_data IN OUT NOCOPY WORKSHEET_TABLE, p_worksheet_rec T_WORKSHEET_DATA)	Convenience procedure that adds a T_WORKSHEET_DATA type to a WORKSHEET_TABLE collection. The WORKSHEET_TABLE collection is passed as an IN/OUT parameter.
addConditionType	PROCEDURE addConditionType(p_condition_data IN OUT NOCOPY CONDITIONS_TABLE, p_condition_rec T_CONDITION)	Convenience procedure that adds a T_CONDITION type to a CONDITIONS_TABLE collection. The CONDITIONS_TABLE collection is passed as an IN/OUT parameter.
addConditionalFormatType	PROCEDURE addConditionalFormatType(p_cond_format_data IN OUT NOCOPY CONDITIONAL_FORMATS_TABLE, p_cond_format_rec T_CONDITIONAL_FORMATS)	Convenience procedure that adds a T_CONDITIONAL_FORMATS type to a CONDITIONAL_FORMATS_TABLE collection. The CONDITIONAL_FORMATS_TABLE collection is passed as an IN/OUT parameter.
addHeaderFooterType	PROCEDURE addHeaderFooterType(p_hf_data IN OUT NOCOPY WORKSHEET_HF_TABLE, p_hf_rec T_WORKSHEET_HF_DATA)	Convenience procedure that adds a T_WORKSHEET_HF_DATA type to a WORKSHEET_HF_TABLE collection. The WORKSHEET_HF_TABLE collection is passed as an IN/OUT parameter.
createWorksheetLink	FUNCTION createWorksheetLink(p_worksheet_name VARCHAR2 := NULL, p_link_text VARCHAR2 := NULL) RETURN VARCHAR2	This function creates an ExcelDocTypeUtils Worksheet hyperlink reference. The generated link will navigate the user to the first cell in the first row of the indicated worksheet. **Example:** ExcelHRef:::#Sheet1!A1:::Sheet1 The formatted text above will be translated in intra-worksheet hyperlink.
getColHeaderString	FUNCTION getColHeaderString(p_query VARCHAR2 := NULL, p_list_delimiter VARCHAR2 := ',') RETURN VARCHAR2;	Given a SELECT statement, this function will generate a delimited Column Header list based on column details or column alias in the query's SELECT clause. The p_list_delimiter parameter needs to match the list delimiter defined for the current worksheet.

Procedure/Function	Specification	Description
createExternalLink	FUNCTION createExternalLink(p_url VARCHAR2 := NULL, p_link_text VARCHAR2 := NULL) RETURN VARCHAR2	This function creates an ExcelDocTypeUtils External hyperlink reference. The generated link will open a web browser window. **Example:** ExcelHref:::http://www.google.com:::Google
createExcelDocument	FUNCTION createExcelDocument(p_worksheet_data WORKSHEET_TABLE, p_style_data STYLE_LIST := STYLE_LIST()) RETURN ExcelDocumentType	This is the primary function in the API. The createExcelDocument function takes a WORKSHEET_TABLE collection and a STYLE_LIST collection as parameters. The function processes all of the T_WORKSHEET_DATA objects/types in the WORKSHEET_TABLE collection to produce a fully populated ExcelDocumentType object.

Putting It All Together …

The following code sample (included with the API code set) give a complete example of creating a multi-worksheet document using the vast majority of the features described in this document.

```
/**
*  This example covers a few features:
*  - Multiple worksheets with multiple queries
*  - Creating Styles and applying them to columns
*  - Worksheet Title (spanning multiple cells)
*  - Conditional Formatting for a range of cells in a worksheet
*  - Sending finished report to a web browser (call it thru a PL/SQL DAD ...)
*  - Hyperlinked cells
*  - Worksheet Level Print Header and Footer
*  - Adding a formula to a column
*/
SET SCAN OFF;

CREATE OR REPLACE PROCEDURE employeeReport AS

  -- Notice the special hyperlink function in col1 of salary select statement links the column to the Hiredate worksheet
```

```
  v_sql_salary       VARCHAR2(200) := 'SELECT ExcelDocTypeUtils.createWorksheetLink(''Hiredate'',last_name),first_name,salary FROM
hr.employees ORDER BY last_name,first_name';
  v_sql_contact      VARCHAR2(200) := 'SELECT last_name,first_name,phone_number,email  FROM hr.employees ORDER BY
last_name,first_name';
  v_sql_hiredate     VARCHAR2(200) := 'SELECT last_name,first_name,to_char(hire_date,''MM/DD/YYYY'') hire_date FROM hr.employees
ORDER BY last_name,first_name';

  excelReport        ExcelDocumentType := ExcelDocumentType();
  documentArray      ExcelDocumentLine := ExcelDocumentLine();

  v_worksheet_rec    ExcelDocTypeUtils.T_WORKSHEET_DATA := NULL;
  v_worksheet_array  ExcelDocTypeUtils.WORKSHEET_TABLE  := ExcelDocTypeUtils.WORKSHEET_TABLE();

  v_sheet_title      ExcelDocTypeUtils.T_SHEET_TITLE := NULL;

  -- Objects for Defining Document Styles (Optional)

  v_style_def        ExcelDocTypeUtils.T_STYLE_DEF := NULL;
  v_style_array      ExcelDocTypeUtils.STYLE_LIST  := ExcelDocTypeUtils.STYLE_LIST();

  -- Object for Defining Conditional Formating (Optional)

  v_condition_rec        ExcelDocTypeUtils.T_CONDITION     := NULL;
  v_condition_array      ExcelDocTypeUtils.CONDITIONS_TABLE := ExcelDocTypeUtils.CONDITIONS_TABLE();

  -- Conditions are applied to a range of cells ... there can be more than grouping of format conditions per worksheet.
  v_conditional_format_rec   ExcelDocTypeUtils.T_CONDITIONAL_FORMATS;
  v_conditional_format_array ExcelDocTypeUtils.CONDITIONAL_FORMATS_TABLE := ExcelDocTypeUtils.CONDITIONAL_FORMATS_TABLE();

  -- Worksheet Print Headers and Footers.  A Header or Footer can have multiple components
  v_header_rec        ExcelDocTypeUtils.T_WORKSHEET_HF_DATA := NULL;
  v_header_rec_array  ExcelDocTypeUtils.WORKSHEET_HF_TABLE  := ExcelDocTypeUtils.WORKSHEET_HF_TABLE();

  v_footer_rec        ExcelDocTypeUtils.T_WORKSHEET_HF_DATA := NULL;
  v_footer_rec_array  ExcelDocTypeUtils.WORKSHEET_HF_TABLE  := ExcelDocTypeUtils.WORKSHEET_HF_TABLE();

BEGIN

 -- Define Styles (Optional)
   v_style_def.p_style_id    := 'LastnameStyle';
   v_style_def.p_text_color  := 'Red';

   ExcelDocTypeUtils.addStyleType(v_style_array,v_style_def);

   v_style_def := NULL;
   v_style_def.p_style_id        := 'SheetTitleStyle';
   v_style_def.p_align_horizontal  := 'Center';
   v_style_def.p_bold            := 'Y';
   v_style_def.p_text_color      := 'Green';

   ExcelDocTypeUtils.addStyleType(v_style_array,v_style_def);

   v_style_def := NULL;
   v_style_def.p_style_id    := 'FirstnameStyle';
   v_style_def.p_text_color  := 'Blue';

   ExcelDocTypeUtils.addStyleType(v_style_array,v_style_def);
```

```
-- Heading Row Style
v_style_def := NULL;
v_style_def.p_style_id        := 'HeadingRowStyle';
v_style_def.p_text_color      := 'Black';
v_style_def.p_font            := 'Times New Roman';
v_style_def.p_ffamily         := 'Roman';
v_style_def.p_fsize           := '10';
v_style_def.p_bold            := 'Y';
v_style_def.p_underline       := 'Single';
v_style_def.p_align_vertical  := 'Bottom';
v_style_def.p_rotate_text_deg  := '45';

ExcelDocTypeUtils.addStyleType(v_style_array,v_style_def);

-- Style that includes custom borders around numbers
v_style_def := NULL;
v_style_def.p_style_id       := 'NumberStyle';
v_style_def.p_number_format   := '$###,###,###.00';
v_style_def.p_align_horizontal := 'Right';
v_style_def.p_custom_xml       := '<Borders>'||
                                  '<Border ss:Position="Left" ss:LineStyle="Continuous" ss:Weight="3"/>'||
                                  '<Border ss:Position="Right" ss:LineStyle="Continuous" ss:Weight="3"/>'||
                                  '<Border ss:Position="Top" ss:LineStyle="Continuous" ss:Weight="3"/>'||
                                  '<Border ss:Position="Bottom" ss:LineStyle="Continuous" ss:Weight="3"/>'||
                                '</Borders>';

ExcelDocTypeUtils.addStyleType(v_style_array,v_style_def);

-- Define Sheet Title
v_sheet_title.title     := 'Employee Salary Report';

-- Must Less than or Equal to the max number of columns on the worksheet.
v_sheet_title.cell_span  := '3';
v_sheet_title.style     := 'SheetTitleStyle';

v_worksheet_rec.title    := v_sheet_title;

-- Add conditional formating for Salary Ranges ... color code salary amounts
-- across three different ranges.

v_condition_rec.qualifier    := 'Between';
v_condition_rec.value       := '0,5000';
v_condition_rec.format_style := 'color:red';

ExcelDocTypeUtils.addConditionType(v_condition_array,v_condition_rec);

v_condition_rec.qualifier    := 'Between';
v_condition_rec.value       := '5001,10000';
v_condition_rec.format_style := 'color:blue';

ExcelDocTypeUtils.addConditionType(v_condition_array,v_condition_rec);


v_condition_rec.qualifier    := 'Between';
v_condition_rec.value       := '10001,1000000';
```

```
v_condition_rec.format_style := 'color:green';

ExcelDocTypeUtils.addConditionType(v_condition_array,v_condition_rec);

-- Format range for Column 3 starting at row 2 and going to row 65000 ...
v_conditional_format_rec.range      := 'R2C3:R65000C3';
v_conditional_format_rec.conditions := v_condition_array;

ExcelDocTypeUtils.addConditionalFormatType(v_conditional_format_array,v_conditional_format_rec);

-- All Names beginning with 'Ba' will be purple.

v_condition_array          := ExcelDocTypeUtils.CONDITIONS_TABLE();

v_condition_rec            := NULL;
v_condition_rec.value      := 'NOT(ISERROR(SEARCH(&amp;quot;Ba&amp;quot;,RC)))';
v_condition_rec.format_style := 'color:purple';

ExcelDocTypeUtils.addConditionType(v_condition_array,v_condition_rec);

-- Format range for Column 1 starting at row 2 and going to row 65000 ...
v_conditional_format_rec          := NULL;
v_conditional_format_rec.range      := 'R2C1:R65000C1';
v_conditional_format_rec.conditions := v_condition_array;

ExcelDocTypeUtils.addConditionalFormatType(v_conditional_format_array,v_conditional_format_rec);

v_worksheet_rec.worksheet_cond_formats := v_conditional_format_array;


-- Create Header Footer Elements
v_header_rec.hf_type  := ExcelDocTypeUtils.HF_DATE_TIME;
v_header_rec.position := ExcelDocTypeUtils.HF_RIGHT;

ExcelDocTypeUtils.addHeaderFooterType(v_header_rec_array,v_header_rec);

v_header_rec := NULL;
v_header_rec.hf_type  := ExcelDocTypeUtils.HF_TEXT;
v_header_rec.position := ExcelDocTypeUtils.HF_CENTER;
v_header_rec.text     := 'Employee Report';
v_header_rec.fontsize := '12';

ExcelDocTypeUtils.addHeaderFooterType(v_header_rec_array,v_header_rec);

v_footer_rec := NULL;
v_footer_rec.hf_type  := ExcelDocTypeUtils.HF_PAGE_NUMBER_PAGES;
v_footer_rec.position := ExcelDocTypeUtils.HF_CENTER;


ExcelDocTypeUtils.addHeaderFooterType(v_footer_rec_array,v_footer_rec);


-- Salary

v_worksheet_rec.worksheet_header := v_header_rec_array;
v_worksheet_rec.worksheet_footer := v_footer_rec_array;
```

```
  -- !!! SETTING THE LIST ITEM DELIMITER TO A ':' INSTEAD OF THE DEFAULT ',' !!!
  v_worksheet_rec.worksheet_list_delimiter := ':';

  v_worksheet_rec.query              := v_sql_salary;
  v_worksheet_rec.worksheet_name     := 'Salaries';
  v_worksheet_rec.col_count          := 3;
  v_worksheet_rec.col_width_list     := '25:20:15';
  v_worksheet_rec.col_header_freeze  := TRUE;
  v_worksheet_rec.col_header_list    := 'Lastname:Firstname:Salary';
  v_worksheet_rec.col_header_style_list := 'HeadingRowStyle:HeadingRowStyle:HeadingRowStyle';
  v_worksheet_rec.col_datatype_list  := 'String:String:Number';
  v_worksheet_rec.col_style_list     := 'LastnameStyle:FirstnameStyle:NumberStyle';

  -- Add a SUM formula to the last column. Valid simply formulas include SUM, AVERAGE, COUNT, MIN, MAX
  v_worksheet_rec.col_formula_list  := '::SUM';

  ExcelDocTypeUtils.addWorksheetType(v_worksheet_array,v_worksheet_rec);

  v_worksheet_rec := NULL;

  -- Contact
  v_worksheet_rec.worksheet_header := v_header_rec_array;
  v_worksheet_rec.worksheet_footer := v_footer_rec_array;


  -- !!! THE LISTS HERE USE THE DEFAULT ITEM DELIMITER ',' !!!
  v_worksheet_rec.query         := v_sql_contact;
  v_worksheet_rec.worksheet_name  := 'Contact_Info';
  v_worksheet_rec.col_count     := 4;
  v_worksheet_rec.col_width_list  := '25,25,25,25';
  v_worksheet_rec.col_header_list := 'Lastname,Firstname,Phone,Email';
  v_worksheet_rec.col_style_list   := 'LastnameStyle,FirstnameStyle,,';

  ExcelDocTypeUtils.addWorksheetType(v_worksheet_array,v_worksheet_rec);
  v_worksheet_rec := NULL;

  -- Hiredate

  v_worksheet_rec.worksheet_header := v_header_rec_array;
  v_worksheet_rec.worksheet_footer := v_footer_rec_array;

  v_worksheet_rec.query         := v_sql_hiredate;
  v_worksheet_rec.worksheet_name  := 'Hiredate';
  v_worksheet_rec.col_count     := 3;
  v_worksheet_rec.col_width_list  := '25,20,20';
  v_worksheet_rec.col_header_list := 'Lastname,Firstname,Hiredate';
  v_worksheet_rec.col_style_list   := 'LastnameStyle,FirstnameStyle,,';

  ExcelDocTypeUtils.addWorksheetType(v_worksheet_array,v_worksheet_rec);

  excelReport := ExcelDocTypeUtils.createExcelDocument(v_worksheet_array,v_style_array);

  excelReport.displayDocument;

END;
/
```