

Saving Report Output in Excel XLSX Format

With Excel® 2007, Microsoft® introduced enhanced spreadsheet functionality in a new workbook file format. WebFOCUS Release 7.7.02 introduced the capability to retrieve data from any WebFOCUS supported data source and generate a native XLSX format (Excel 2007, Excel 2010, and Excel 2013) workbook for data analysis and distribution.

As of WebFOCUS Release 7.7.03 Gen 716 and Release 7.7.04 Gen 753, the WebFOCUS XLSX/EXL07 format supports the following Microsoft Office software products:

- ❑ Microsoft Office 2013/2010/2007 and Microsoft Office 2000/2003 with the Microsoft Office Compatibility Pack.
- ❑ Open Office Support (FORMAT EXL07/XLSX). Core Excel functionality generated by the EXL07/XLSX format is supported for Open Office as of WebFOCUS 8. For details on Open Office, see <http://www.openoffice.org/>.
- ❑ MAC Office 2008 and 2011. FORMAT EXL07/XLSX is certified with WebFOCUS 8.

WebFOCUS generates XLSX workbooks based on the Microsoft XLSX standard. These workbooks are accessible through all browsers and mobile applications that support native Microsoft XLSX files.

Note: This Technical Memo applies to Excel 2007, Excel 2010, and Excel 2013, unless otherwise indicated.

Overview of EXL07/XLSX Format

FORMAT EXL07 and FORMAT XLSX are synonyms and can be used interchangeably. The FILE SAVED message will always display "XLSX FILE SAVED", regardless of the syntax specified.

The WebFOCUS procedure generates a new workbook containing a single worksheet with the report output containing your defined report elements (headings and subtotals), as well as StyleSheet syntax (such as conditional styling and drill downs):

- ❑ XLSX format accurately displays formatted numeric, character, and date formats.

- ❑ XLSX FORMULA enables you to convert summed information (such as column totals, row totals, and calculated values) into Excel formulas that will automatically update as you edit the Excel worksheet.
- ❑ ReportCaster supports distribution of XLSX workbooks and XLSX FORMULA workbooks.
- ❑ Within each generated worksheet, the columns in the report are automatically sized to fit the largest value in the column (SQUEEZE=ON). WebFOCUS calculates the width of each data column based on the font and size requirement of all cells in that column using font metrics developed for other styled formats, including PDF and DHTML. Calculations are based on the data and title elements of the report. Heading and footing elements are not used in the sizing calculation and will be sized based on the data column requirements.
- ❑ By default, there is a standard height for the data and Title rows. Heading, Footing, Subhead, and Subfoot rows are taller than the data rows to support wrapping and for a clearer distinction between headings and data.
- ❑ Using the TITLETEXT StyleSheet attribute described in the *Creating Reports With WebFOCUS Language* manual, tab names within the workbook can be customized to provide better descriptions of the worksheet content.
- ❑ Unlike the HTML-based (EXL2K) format, which removes all blanks, XLSX, by default, retains leading, internal, and trailing blanks in cells within the worksheet. For more information on how to affect these blanks, see [Preserving Leading and Internal Blanks in Report Output](#) on page 32.
- ❑ An XLSX worksheet can contain 1,048,576 rows by 16,384 columns. WebFOCUS will generate worksheets larger than these defined limits, but Excel is not able to open the workbook. For more information on how to support overflow in worksheets, see [Overcoming the Excel 2007/2010 Row Limit Using Overflow Worksheets](#) on page 58.
- ❑ Because of the new format of the zipped XLSX files, native HTML symbols, such as a caret (<), cannot be supported. To display as data in a cell in a workbook, use the SET HTMLENCODE command. For more information on the SET HTMLENCODE command, see the *Developing Reporting Applications* manual.

Building the .xlsx Workbook File

Microsoft changed the format and structure of the Excel workbook in Excel 2007. The new .xlsx file is a binary compilation of a group of xml files. Generating this new file format using WebFOCUS is a two-step process that consists of generating the xml files containing the report output and zipping the xml documents into the binary .xlsx format. The Reporting Server performs the xml generation process. The zipping process can be completed either by the client (WebFOCUS Servlet) or the server (JSCOM3):

- ❑ **WebFOCUS Servlet.** The WebFOCUS Client within the application server performs the zipping process. This can be done within the local client or through a remotely accessed client. The servlet method is the default approach defined for each WebFOCUS Client, with the client pointing to itself, by default.
- ❑ **JSCOM3.** The Java layer of the Reporting Server performs the zipping operation. This option should be used when the WebFOCUS Servlet is configured on a secured web or application server. This is because JSCOM3 does not require URL access to a remote WebFOCUS Client.

Syntax: How to Select the Method for Zipping the .xlsx File

You designate the method and location where the zipping will occur by setting EXCELSERVURL to a URL (for the WebFOCUS Servlet) or to a blank (for JSCOM3). You can set this value for a specific procedure or for the entire environment:

- ❑ **For a procedure.** Issue the SET EXCELSERVURL command within the procedure.
- ❑ **For the entire environment.** Edit the IBIF_excelservurl variable in the WebFOCUS Administration Console by selecting:

Configuration/Client Settings/General/IBIF_excelservurl

For more information on accessing the WebFOCUS Administration Console and setting the IBIF_excelservurl variable, see the *WebFOCUS Security and Administration* manual.

The value you assign to EXCELSERVURL determines whether the WebFOCUS Servlet or JSCOM3 performs the zipping operation:

- ❑ **Specifying the Servlet.** To specify that the WebFOCUS Servlet should be used, set the EXCELSERVURL parameter or the IBIF_variable to the URL of a WebFOCUS Release 7.7.02 or higher client configuration. For example,

In a procedure:

```
SET EXCELSERVURL = http://servername:8080/ibi_apps
```

In the WebFOCUS Administration Console:

```
IBIF_excelservurl = http://servername:8080/ibi_apps
```

- ❑ **Specifying JSCOM3.** To specify that JSCOM3 should be used within the current Reporting Server, set EXCELSERVURL to a blank or an empty string.

In a procedure:

```
SET EXCELSERVURL = ''
```

In the WebFOCUS Administration Console:

```
IBIF_excelservurl = ''
```

By default, each WebFOCUS Client contains the following URL definition that points to itself:

```
&URL_PROTOCOL://&servername:&server_port&IBIF_webapp
```

Syntax: How to Generate an Excel XLSX Workbook

You can specify that a report should be saved to an XLSX workbook, displayed in the browser, or displayed in the Excel application.

```
ON TABLE {PCHOLD|HOLD} AS name FORMAT XLSX
```

where:

PCHOLD

Displays the generated workbook in either the browser or the Excel application, based on your desktop settings. For information, see [Viewing Excel Workbooks in the Browser vs. the Excel Application](#) on page 6.

HOLD

Saves a workbook with an .xlsx extension to the designated location.

name

Specifies a file name for the generated workbook.

Note: To assign a file name to the generated workbook, set the *Save Report* option to *YES* for the .xlsx file extension in the WebFOCUS Client Redirection Settings. When opened in the Excel application, the generated workbook will retain the designated AS name. For more information, see the *WebFOCUS Security and Administration* manual.

Opening XLSX Report Output

To open XLSX workbooks, Excel 2013, 2010, or 2007 must be installed on the desktop.

Reference: Opening XLSX Report Output in Excel 2000/2003

Excel 2000 and Excel 2003 can be updated to read Excel XLSX workbooks using the Microsoft Office Compatibility Pack available from the Microsoft download site (<http://www.microsoft.com/downloads/en/default.aspx>). When the file extension of the file being opened is .xlsx (XLSX workbook), the Microsoft Office Compatibility Pack performs the necessary conversion to allow Excel 2000/2003 to read and open it.

In addition to the Microsoft Office Compatibility Pack, it is important to enable the WebFOCUS Client Redirection Settings Save As option so that Excel 2000/2003 will be able to open the XLSX report output without users first having to save it to their machine with the .xlsx file extension. The WebFOCUS Client processing Redirection Settings Save As option configures how the WebFOCUS Client sends each report output file type to the user machine. This option can be set as follows:

- ❑ **Save As Option disabled (NO).** The WebFOCUS Client Redirection Setting Save As is disabled by default. When the Save As option is disabled, the WebFOCUS Client sends report output to the user machine in memory with the application association specified for the report format in the WebFOCUS Client Redirection Settings configuration file (mime.wfs).

A user machine that does not have Excel 2007/2010 installed will not recognize the application association for Excel 2007/2010 and Excel will display a message.

The Excel 2000/2003 user can select Save and provide a file name with the .xlsx extension to save the report output to their machine. The user can then open the .xlsx file directly from Excel 2000/2003.

- ❑ **Save As Option enabled (YES).** When the WebFOCUS Redirection Save As option is enabled, the WebFOCUS Client sends the report output to the user as a file with the extension specified in the WebFOCUS Client Redirection Settings configuration file (mime.wfs).

Upon receiving the file, Windows will display the File Download prompt asking the user to Open or Save the file with the identified application type. The File Download prompt displays the Name with the .xlsx file extension for the report output that is recognized as an Excel XLSX file type.

Note: The download prompt will display for all users, including users who have Excel 2007/2010 installed on their machines.

If an Excel 2000/2003 user chooses to open the file, the Microsoft Office Compatibility Pack will recognize the .xlsx file extension and perform the necessary conversion to allow Excel 2000/2003 to read the Excel XLSX workbook.

If an Excel 2007/2010 user chooses to open the file, Excel will recognize the .xlsx file extension and read the Excel XLSX workbook.

For additional information on WebFOCUS Client Redirection Settings, see the *WebFOCUS Security and Administration Guide*.

Reference: Viewing Excel Workbooks in the Browser vs. the Excel Application

Your Operating System and desktop settings determine whether Excel output sent to the client is displayed in an Internet Browser window or within the Excel application. When Excel output has been defined within the Windows environment to *Browse in same window*, the workbook generated by a WebFOCUS request is opened within an Internet Explorer® browser window. When the *Browse in same window* option is unchecked for the .xls file type, the browser window created by WebFOCUS is blank because the report output is displayed in the stand-alone Excel application window.

- ❑ In Windows XP and earlier, file type specific settings are managed on the desktop within Windows Explorer by selecting *Tools/Folder Options*, clicking the *File Types* tab, selecting the extension (.xls or .xlsx), clicking the *Advanced* button, and checking the *Browse in same window* box.
- ❑ In Windows 7, Microsoft removed the desktop settings that support opening worksheets in the browser. This means that to change this behavior, you can no longer simply navigate to the Folder Options dialog box, but that you must change a registry setting. This change is documented in the Microsoft Knowledge Base Article ID 927009 at the following web site:

<http://support.microsoft.com/kb/927009>

Note: This works the same for both EXL2K and XLSX formats. The only difference is the selection of file type based on the version of Excel output you will be generating.

Formatting Values Within Cells in XLSX Report Output

WebFOCUS formats defined in Master Files or within a FOCEXEC will be represented in the resulting cells in an Excel XLSX worksheet. Where possible, the WebFOCUS formats are translated to custom Excel formats and applied to values passed as raw data. Each data value passed to a cell in Excel is defined with a value and a format mask pair. The data format is associated with the cell rather than embedded in the value. This technique provides enhanced support for editing worksheets generated by WebFOCUS. New values entered into existing cells will retain the cell formats and continue to display in the style defined for the column within the report.

The following types of data can be passed to Excel:

- ❑ **Numeric.** Where corresponding Excel format masks can be defined, numeric values are passed as raw values with associated format masks. In instances where an equivalent format mask cannot be defined, the numeric value is passed as a text string.
- ❑ **Alphanumeric.** Alphanumeric formats are passed to Excel as text strings, with General format defined. By default, General format presents all text fields as left-justified. Alignment and other styling attributes can be applied to these cells to override the default.
- ❑ **Date formats.** Data that contain sufficient elements to define a valid Excel date format are passed as raw date values with the WebFOCUS formats translated to Excel date format masks. In WebFOCUS formats that do not contain sufficient information to create valid Excel date values, the dates are converted to text strings.
- ❑ **Date-Time formats.** Date-time values are passed as raw date-time values with WebFOCUS formats translated to Excel date-time format masks using Custom formats.
- ❑ **Text.** Text values are passed as strings with General Format defined (as with alphanumeric data).

Note: This behavior is a change from EXL2K format, where cells containing dates and more complex numeric formats were passed as formatted text.

Displaying Formatted Numeric Values in XLSX Report Output

Each numeric WebFOCUS format is translated to a custom numeric Excel format. The numeric value is displayed in the Excel formula bar for the selected cell. Within the actual cell, the value with the format mask applied displays.

The WebFOCUS formats for the following numeric data types are translated into Excel XLSX format masks supporting full editing within the resulting workbook:

- ❑ Data types: E, F, D, I, P
- ❑ Comma edit option (C)
- ❑ Zero suppression (S)

Formatting Values Within Cells in XLSX Report Output

- ❑ Leading zero (L)
- ❑ Floating currency symbol (M)
- ❑ Comma suppression (c)
- ❑ Right-side minus sign (-)
- ❑ Credit negative (CR)
- ❑ Bracket negative (B)
- ❑ Fixed extended currency symbol (!d, !e, !l, !y)
- ❑ Floating extended currency symbol (!D, !E, !L, !Y)
- ❑ Percent (%)

Example: Passing Numeric Formats to XLSX Report Output

In the following example, the DOLLARS field is assigned different numeric formats to demonstrate different available options. The column titles have been edited to display the WebFOCUS format options that have been applied:

```
TABLE FILE GGSales
SUM DOLLARS/D12.2 AS 'D12.2'
    DOLLARS/D12C AS 'D12C'
    DOLLARS/D12CM AS 'D12CM'
BY REGION
BY CATEGORY
ON TABLE PCHOLD FORMAT XLSX
ON TABLE SET BYDISPLAY ON
END
```

In the resulting worksheet, notice that cell C2 containing the DOLLAR value for *Midwest Coffee* presents the value with the WebFOCUS format D12.2, which presents the comma (,) and two decimal places. On the formula bar, the actual value is presented without any formatting. Examine each of the DOLLAR values in each row to see that the value as displayed in the formula bar remains the same, and only the display values presented in each cell change.

Also notice that with SET BYDISPLAY ON, the BY field values are repeated for every row on the worksheet. This creates fully qualified data rows that can be used with various data sorting, filtering, and table features in Excel without losing valuable information. This setting is recommended as a best practice for all worksheets.

A1		fx Region				
	A	B	C	D	E	F
1	Region	Category	D12.2	D12C	D12CM	
2	Midwest	Coffee	4,178,513.00	4,178,513	\$4,178,513	
3	Midwest	Food	4,338,271.00	4,338,271	\$4,338,271	
4	Midwest	Gifts	2,883,881.00	2,883,881	\$2,883,881	
5	Northeast	Coffee	4,164,017.00	4,164,017	\$4,164,017	
6	Northeast	Food	4,379,994.00	4,379,994	\$4,379,994	
7	Northeast	Gifts	2,848,289.00	2,848,289	\$2,848,289	
8	Southeast	Coffee	4,415,408.00	4,415,408	\$4,415,408	
9	Southeast	Food	4,308,731.00	4,308,731	\$4,308,731	
10	Southeast	Gifts	2,986,240.00	2,986,240	\$2,986,240	
11	West	Coffee	4,473,517.00	4,473,517	\$4,473,517	
12	West	Food	4,202,337.00	4,202,337	\$4,202,337	
13	West	Gifts	2,977,092.00	2,977,092	\$2,977,092	
14						

The following example uses Fixed Dollar (N) format, as well as multiple combined format options. Each WebFOCUS format option is translated to the appropriate Excel XLSX format mask and applied to the cell value:

```
TABLE FILE GGSALES
SUM BUDDOLLARS/D12N
   DOLLARS/D12M
COMPUTE OVERBUDGET/D12BMc = BUDDOLLARS-DOLLARS; AS 'Over Budget'
BY REGION
BY CATEGORY
ON TABLE PCHOLD FORMAT XLSX
ON TABLE SET BYDISPLAY ON
END
```

Notice the fixed numeric format defined for the BUDDOLLARS column (Column C) presents the local currency symbol in a fixed position within each cell, regardless of the size of the data value. On the formula bar, the values in the *Over Budget* calculated field is passed as a negative value where appropriate. In the actual cells, the bracketed styling is applied to the negative values as part of the custom Excel XLSX format mask.

E2		fx -92481				
	A	B	C	D	E	F
1	Region	Category	Budget Dollars	Dollar Sales	Over Budget	
2	Midwest	Coffee	\$ 4,086,032	\$4,178,513	(\$92481)	
3	Midwest	Food	\$ 4,220,721	\$4,338,271	(\$117550)	
4	Midwest	Gifts	\$ 2,887,620	\$2,883,881	\$3739	
5	Northeast	Coffee	\$ 4,252,462	\$4,164,017	\$88445	
6	Northeast	Food	\$ 4,453,907	\$4,379,994	\$73913	
7	Northeast	Gifts	\$ 2,870,552	\$2,848,289	\$22263	
8	Southeast	Coffee	\$ 4,431,429	\$4,415,408	\$16021	
9	Southeast	Food	\$ 4,409,288	\$4,308,731	\$100557	
10	Southeast	Gifts	\$ 2,967,254	\$2,986,240	(\$18986)	
11	West	Coffee	\$ 4,523,963	\$4,473,517	\$50446	
12	West	Food	\$ 4,183,244	\$4,202,337	(\$19093)	
13	West	Gifts	\$ 2,934,306	\$2,977,092	(\$42786)	
14						
15						

Using Numeric Formats in Report Headings and Footings

By default, headings and footings are passed to Excel as a single character string. Spot markers are not supported for positioning within each line. Numeric fields and dates passed in headings and footings are passed as text strings within the overall heading or footing contents.

To display numeric fields and dates within headings and footings as numeric or date values, use HEADALIGN=BODY in the StyleSheet to define each of the items in the heading as an individual cell. Each cell containing numeric or date values will then be passed as the appropriate value with the associated format mask. For information about the HEADALIGN attribute, see the *Creating Reports With WebFOCUS Language* manual.

Using Numeric Format Punctuation in Headings and Footings

For data columns, all currency formats are translated using the Excel XLSX format masks that use the punctuation rules defined by the regional settings of the desktop.

In languages that use Continental Decimal Notation, the currency definitions designate that a comma (,) is used as the decimal separator, and a period (.) is used as the thousands separator, so D12.2CM may present the value as \$ 9.999,99 rather than the English (United States) value \$ 9,999.99. In headings and footings, you can designate that punctuation should be converted to Continental Decimal notation by issuing the SET CDN=ON command. With this setting in effect, the data embedded within heading and footing text strings will be formatted using the converted punctuation. Specify HEADALIGN=BODY to delineate items as individual cells and to retain the numeric formatting within the field, which will follow the same rules as the report data within the data columns.

Passing Dates to XLSX Report Output

Most translated and smart dates can be sent to Excel as standard date values with format masks, enabling Excel to use them in functions, formulas, and sort sequences.

Excel 2007 only supports mixed-case date text strings so all month and day names are displayed in mixed-case, regardless of how the case has been specified in the WebFOCUS date format. For example, the WebFOCUS date format WRYMTRD presents the date text information in uppercase in all non-Excel formats. Excel transforms this value to mixed-case automatically.

In HTML, the date format displays as:

WRYMTRD
FRIDAY, 10 JANUARY 1

In XLSX, the date format displays as:

WRYMTRD
Friday, 10 January 1

Example: Translating WebFOCUS Dates to Excel XLSX Dates

The following request against the GGSALES data source creates the date January 1, 2010 and converts it to four date formats with translated text:

```
DEFINE FILE GGSALES
NEWDATE/MDYY = '01/01/2010';
WRMtrDY/WRMtrDY = NEWDATE;
wDMTY/wDMTY = NEWDATE;
wrDMTRY/wrDMTRY = NEWDATE;
wrYMtrD/wrYMtrD = NEWDATE;
END
TABLE FILE GGSALES
SUM DATE NOPRINT
NEWDATE WRMtrDY wDMTY wrDMTRY wrYMtrD
ON TABLE PCHOLD FORMAT XLSX
END
```

The following table shows how the dates should appear.

WebFOCUS Format	WebFOCUS Display	XLSX Display	XLSX Value
WRMtrDY	FRIDAY, January 1 10	Friday, January 1 10	1/1/2010
wDMTY	Fri, 1 JAN 10	Fri, 1 Jan 10	1/1/2010
wrDMTY	Friday, 1 JANUARY 10	Friday, 1 January 10	1/1/2010
wrYMtrD	FRIDAY, 10 JANUARY 1	Friday, 10 January 1	1/1/2010

In Excel 2007/2010, all of the cells have a date value with format masks, and all month and day names are in mixed-case, regardless of how the case has been specified in the WebFOCUS format. The output is:

	A	B	C	D	E
1	NEWDATE	WRMtrDY	wDMTY	wrDMTRY	wrYMtrD
2	01/01/2010	Friday, January 1 10	Fri, 1 Jan 10	Friday, 1 January 10	Friday, 10 January 1

Passing Dates Without a Day Component

Date formats that do not specify the day value explicitly are defined as the date value of the first day of the month. Therefore, the value placed in the cell may be different from the day component value in the source data field and may produce unexpected results when used for sorting or date calculations in an Excel formula.

The following table shows how WebFOCUS date formats are represented in XLSX. The table shows how the value is preserved in the cell and how the display is generated using the format mask that corresponds to the WebFOCUS date format.

`DATEFLD/MDYY = '01/02/2010'`

WebFOCUS Format	XLSX Display	XLSX Value
DMYY	02/01/2010	1/2/2010
MY	01/10	1/1/2010
MTY	Jan, 10	1/1/2010
MTDY	Jan 2, 10	1/2/2010

Example: Passing WebFOCUS Dates With and Without a Day Component to XLSX Report Output

The following request against the GGSALES data source creates the date January 2, 2010 and passes it to Excel with formats MDYY, DMYY, MY, and MTDY:

```
DEFINE FILE GGSALES
NEWDATE/MDYY = '01/02/2010';
END
TABLE FILE GGSALES
SUM DATE NOPRINT
NEWDATE AS 'MDYY' NEWDATE/DMYY AS 'DMYY' NEWDATE/MY AS 'MY'
      NEWDATE/MTY AS 'MTY' NEWDATE/MTDY AS 'MTDY'
ON TABLE PCHOLD FORMAT XLSX
END
```

Columns D and E have actual date values with format masks, displayed by Excel 2007/2010 in mixed-case. Since the MTY format does not have a day component, the date value stored is the first of January 2010 (1/1/2010), not the second of January 2010 (1/2/2010):

	A	B	C	D	E
1	MDYY	DMYY	MY	MTY	MTDY
2	01/02/2010	02/01/2010	01/10	Jan, 10	Jan 2, 10

Passing Date Components for Use in Excel Formulas

Dates formatted as individual components (for example, D, Y, M, W) are passed to Excel as numeric values that can be used as parameters to Excel date functions. The values are passed as General format that are recognized by Excel as numbers.

Example: Passing Numeric Date Components to XLSX Report Output

The following request against the GGSALES data source creates the date January 1, 2010 and extracts numeric date components, passing them to Excel 2007/2010:

```
DEFINE FILE GGSALES
NEWDATE/MDYY = '01/01/2010';
D/D = NEWDATE;
Y/Y = NEWDATE;
W/W = NEWDATE;
w/w = NEWDATE;
M/M = NEWDATE;
YY/YY = NEWDATE;
END
TABLE FILE GGSALES
SUM DATE NOPRINT
NEWDATE D Y W w M YY
ON TABLE PCHOLD FORMAT XLSX
END
```

The output is:

	A	B	C	D	E	F	G
1	NEWDATE	D	Y	W	w	M	YY
2	01/01/2010	01	10	5	5	01	2010

Passing Quarter Formats

Date formats that contain a Quarter component are always passed to Excel as text strings since Excel does not support Quarter formats.

Example: Passing Dates With a Quarter Component to XLSX Report Output

The following request against the GGSALES data source creates the date January 1, 2010 and converts it to date formats that contain a Quarter component:

```
DEFINE FILE GGSALES
NEWDATE/MDYY = '01/01/2010';
Q/Q = NEWDATE;
QY/QY = NEWDATE;
YBQ/YBQ = NEWDATE;
END
TABLE FILE GGSALES
SUM DATE NOPRINT
NEWDATE Q QY YBQ
ON TABLE PCHOLD FORMAT XLSX
END
```

In XLSX, the cells containing dates with Quarter components have General format. To see this, open the Format Cells dialog box.

The output is:

	A	B	C	D
1	NEWDATE Q	QY	YBQ	
2	01/01/2010	Q1	Q1 10	10 Q1

Passing Date Components Defined as Translated Text

Date formats that do not contain sufficient information to present a valid date result in Excel are not translated to a value, including formats that do not contain year and/or month information. These dates will be sent to Excel as text. In the absence of complete information, the year defaults to the current year, so the value sent would be incorrect if this type of format was passed as a date value. The following formats will not be sent as values:

- MT, MTR, Mt, Mtr
- W, w, WR, wr

When date formats are passed to XLSX with format masks, all month and day names are in mixed-case, regardless of how the case has been specified in the WebFOCUS format. However, since the values in this example are always sent as text, the casing defined in the WebFOCUS format is applied in the resulting cell.

Example: Passing Date Components Defined as Translated Text to XLSX Report Output

The following request against the GGSALES data source creates the date January 1, 2010 and converts it to date formats that are defined as either month name or day name:

```
DEFINE FILE GGSALES
NEWDATE/MDYY = '01/01/2010';
MT/MT = NEWDATE;
MTR/MTR = NEWDATE;
Mtr/Mtr = NEWDATE;
WR/WR = NEWDATE;
wr/wr = NEWDATE;
END
TABLE FILE GGSALES
SUM DATE NOPRINT
NEWDATE MT MTR Mtr WR wr
ON TABLE PCHOLD FORMAT XLSX
END
```

In Excel 2007 or 2010, the cells containing the days have General format. To see this, open the Format Cells dialog box.

The output is:

	A	B	C	D	E	F
1	NEWDATE	MT	MTR	Mtr	WR	wr
2	01/01/2010	Jan	January	January	FRIDAY	Friday

Reference: Usage Notes for Date Values in XLSX Report Output

- ❑ The following date formats are not supported in XLSX. They will translate into Excel General format and possibly produce unpredictable results:
 - ❑ JUL, YYJUL, and I2MT.
 - ❑ Dates stored as a packed or alphanumeric field with date display options.

Passing Date-Time to XLSX

Most WebFOCUS date-time formats can be sent to XLSX as standard date/time values with format masks, enabling Excel to use them in functions, formulas, and sort sequences.

As with the Date formats, Excel only supports mixed-case to date-time fields, so if the date-time format contains text and is supported by Excel, the text will be in mixed-case, regardless of the casing defined within the WebFOCUS format.

Example: Passing Date-Time to XLSX

The following request shows an example against the GGSALES data source.

```
DEFINE FILE GGSALES
DT1/HYYMDm WITH REGION = DT(20100506 16:17:01.993876);
DPT1/HDMTYym = DT1;
ALPHA_DATE1/A30 = HCNVRT(DT1, '(HYYMDm)', 30, 'A30');
END
TABLE FILE GGSALES
PRINT
ALPHA_DATE1
DT1 AS 'HYYMDm'
DPT1 AS 'HDMTYym'
DT1/HdMTYYBS AS 'HdMTYYBS'
DT1/HdMTYYBs AS 'HdMTYYBs'
ON TABLE SET SPACES 1
IF RECORDLIMIT EQ 1
ON TABLE PCHOLD FORMAT XLSX
END
```


The output is:

A	B	C	D	E
ALPHA_DATE1	HYYMDm	HDMTYYm	HdMTYYBS	HdMTYYBs
2010/05/06 16:17:01.993876	2010/05/06 16:17:01.993	06 May 2010 16:17:01.993	6 May 2010 16:17:01	6 May 2010 16:17:01.993

Note: Minutes by themselves are not supported in Excel and will be sent as an integer to XLSX with a Custom format.

Also, Excel time formats only support to the milliseconds. WebFOCUS formats that display microseconds will send the value to Excel, but the value will be rounded to milliseconds within the worksheet if the cell is edited.

The following table shows how the date-time values appear.

WebFOCUS Format	XLSX Displays	XLSX Value
HYYMDm	2010/05/06 16:17:01.993	5/6/2010 4:17:02 PM
HDMTYYm	06 May 2010 16:17:01.993	5/6/2010 4:17:02 PM
HdMTYYBS	6 May 2010 16:17:01	5/6/2010 4:17:01 PM
HdMTYYBs	6 May 2010 16:17:01.993	5/6/2010 4:17:02 PM

Generating Native Excel Formulas in XLSX Report Output

When you display or save a tabular report request using XLSX FORMULA, the resulting worksheet contains an Excel formula that computes and displays the results of any type of summed information, such as column totals, row totals, subtotals, and calculated values, rather than static numbers. A formula for a calculated value is generated by translating the internal form of the WebFOCUS expression into an Excel formula. Worksheets saved using the XLSX FORMULA format are interactive, allowing for "what if" scenarios that immediately reflect any additions or modifications made to the data.

Understanding Formula Versus Value

The XLSX FORMULA format will generate formulas rather than values for the following WebFOCUS TABLE commands: ROW-TOTAL, COLUMN-TOTAL, SUB-TOTAL, SUBTOTAL, and SUMMARIZE, as well as for calculations performed by functions.

- A DEFINE field will always generate a constant value and not a formula.
- COMPUTE will generate the formula, except when the COMPUTE is equal to a single variable. In that case, the constant is placed and not the formula.

- ❑ If your report contains a calculated value (generated by the COMPUTE or RECOMPUTE command), all of the fields referenced by the calculated value must be displayed in the report in order for a cell reference to be included in the formula. If the referenced column is not displayed in the workbook, the data value will be placed in the formula, rather than a cell reference. Additionally, if the value cannot be reliably calculated based on the information passed to Excel, the value, rather than an expression, will be used. For example, using the LAST function in WebFOCUS cannot be translated correctly into Excel. In this instance, the LAST value is used in the expression, rather than a cell reference.

XLSX FORMULA is not supported with financial reports created with the Financial Report Painter or the underlying Financial Modeling Language (FML).

For more information, see [Translation Support for FORMAT XLSX FORMULA](#) on page 18.

Reference: Translation Support for FORMAT XLSX FORMULA

This topic describes translation support for FORMAT XLSX FORMULA. Use of unsupported WebFOCUS features may produce unreliable results.

- ❑ All standard operators are supported. These include arithmetic operators, relational operators, string operators, IF/THEN/ELSE, and logical operators. However, column notation is not supported.

The IS-PRESENT, IS-MISSING, IS-FROM, FROM, NOT-FROM, IS-MORE-THAN, IS-LESS-THAN, CONTAINS, and OMITS operators are not supported.

The logical operators AND and OR are not supported in conditional (IF-THEN-ELSE) or logical expressions.

- ❑ The following functions are supported:

ABS, ARGLEN, ATODBL, BYTVAL, CHARGET, CTRAN, DMOD, DOWK, DOWK, DOWKL, EXP, FMOD, HEXBYT, HHMMSS, IMOD, LCWORD, LOCASE, LOG, MAX, MIN, OVLAY, POSIT, RDUNIF, SQRT, SUBSTR, TODAY, and UPCASE. The EDIT function is supported for converting formats (one argument variant). It is not supported for editing strings.

The functions CTRLFD, LJUST, and RJUST are not recommended for justifying data in Excel columns. With the use of Excel proportional fonts, the StyleSheet JUSTIFY attribute is more appropriate.

Be cautious when using functions that use decimal values as an argument (BYTVAL, CTRAN, HEXBYT). Based on whether the operating environment is EBCDIC or ASCII, the results may be different.

- ❑ XLSX FORMULA is not supported with the following WebFOCUS commands and phrases:
 - ❑ DEFINE
 - ❑ OVER

- ❑ FOR
- ❑ NOPRINT
- ❑ Multiple display (PRINT, LIST, SUM, and COUNT) commands
- ❑ SEQUENCE StyleSheet attribute
- ❑ RECAP
- ❑ SET HIDENULLACRS
- ❑ SET SUBTOTALS = ABOVE
- ❑ LAST
- ❑ The BYDISPLAY ON setting is recommended to allow the sort field value to be available on all rows for recalculations.
- ❑ If an expression requires more than 1024 characters, WebFOCUS will place the value into the cell, and not the formula.
- ❑ Conditional styling is based on the values in the original report. If the worksheet values are changed and the formulas are recomputed, the styling will not reflect the updated information.

Syntax: **How to Save Reports as FORMAT XLSX FORMULA**

Add the following syntax to your request to take advantage of Excel formulas in your workbook:

```
ON TABLE {PCHOLD|HOLD} FORMAT XLSX FORMULA
```

where:

PCHOLD

Displays the output in an XLSX workbook.

HOLD

Saves the output for reuse in an Excel worksheet. For details, see *Saving and Reusing Your Report Output* in the *Creating Reports With WebFOCUS Language* manual.

Example: Generating Native Excel Formulas for Column Totals

The following example illustrates how a column total in a report request is translated to an Excel formula when you use the XLSX FORMULA format. Notice that the formatting of the column total (TYPE=GRANDTOTAL) is retained in the Excel workbook. When you select the total in the report, the equation =SUM(B4:B10) displays in the formula bar, representing the column total as a sum of cell ranges.

```
TABLE FILE SHORT
HEADING
"Projected Return By Region"
" "
SUM PROJECTED_RETURN AS 'RETURN'
BY REGION AS 'REGION'
ON TABLE COLUMN-TOTAL
ON TABLE PCHOLD FORMAT XLSX FORMULA
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, FONT='ARIAL', SIZE=9, TITLETEXT='By Region', $
TYPE=TITLE, BACKCOLOR=RGB(102 102 102), COLOR=RGB(255 255 255), $
TYPE=HEADING, SIZE=12, STYLE=BOLD, JUSTIFY=CENTER, $
TYPE=GRANDTOTAL, BACKCOLOR=RGB(210 210 210), STYLE=BOLD, $
END
```

The output is:

The screenshot shows an Excel spreadsheet with a table titled "Projected Return By Region". The table has two columns: "REGION" and "RETURN". The data rows are: CENTRAL AMERICA (9.520), EASTERN EUROPE (16.150), FAR EAST (9.100), MIDDLE EAST (7.980), NORTH AMERICA (12.460), SOUTH AMERICA (8.400), and WESTERN EUROPE (7.980). A "TOTAL" row is highlighted in grey, showing a value of 71.590. The formula bar at the top shows the formula =SUM(B4:B10). The spreadsheet title is "By Region".

	A	B
	Projected Return By Region	
1		
2		
3	REGION	RETURN
4	CENTRAL AMERICA	9.520
5	EASTERN EUROPE	16.150
6	FAR EAST	9.100
7	MIDDLE EAST	7.980
8	NORTH AMERICA	12.460
9	SOUTH AMERICA	8.400
10	WESTERN EUROPE	7.980
11	TOTAL	71.590
12		
13		
14		

WebFOCUS can translate any total (subtotal, row total, or column total) to an Excel formula. For related information, see [Translation Support for FORMAT XLSX FORMULA](#) on page 18.

Example: Generating Native Excel Formulas for Row Totals

The following request calculates totals for returns and balances across continents. The row totals are represented as sums of cell ranges.

```
TABLE FILE SHORT
HEADING
"Projected Return Across Continent"
" "
SUM PROJECTED_RETURN AS 'Return' AND BALANCE AS 'Balance'
ACROSS CONTINENT AS 'CONTINENT'
BY REGION AS 'REGION'
ON CONTINENT ROW-TOTAL AS 'TOTAL'
ON TABLE COLUMN-TOTAL AS 'TOTAL'
ON TABLE PCHOLD FORMAT XLSX FORMULA
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, FONT='ARIAL', SIZE=9, TITLETEXT='Across Continent',$
TYPE=TITLE, BACKCOLOR=RGB(102 102 102), COLOR=RGB(255 255 255),$
TYPE=HEADING, SIZE=12, STYLE=BOLD, JUSTIFY=CENTER,$
TYPE=ACROSSTITLE, STYLE=BOLD,$
TYPE=GRANDTOTAL, BACKCOLOR=RGB(210 210 210), STYLE=BOLD,$
END
```

The following output highlights the formula that calculates the row total in cell I12=C12+E12+G12.

Projected Return Across Continent								
	CONTINENT		ASIA		EUROPE		TOTAL	
REGION	Return	Balance	Return	Balance	Return	Balance	Return	Balance
CENTRAL AMERICA	9.520	40,200,667	9.520	40,200,667
EASTERN EUROPE	16.150	121,977,597	16.150	121,977,597
FAR EAST	.	.	9.100	202,660,842	.	.	9.100	202,660,842
MIDDLE EAST	.	.	7.980	44,659,771	.	.	7.980	44,659,771
NORTH AMERICA	12.460	367,610,683	12.460	367,610,683
SOUTH AMERICA	8.400	87,661,381	8.400	87,661,381
WESTERN EUROPE	7.980	210,550,374	7.980	210,550,374
TOTAL	30.380	495,472,731	17.080	247,320,613	24.130	332,527,971	71.590	1,075,321,315

Example: Generating Native Excel Formulas for Calculated Values

The following request totals the columns for retail cost and dealer cost, and calculates the value of a field called PROFIT by subtracting the DOLLARS from the BUDDOLLARS.

Generating Native Excel Formulas in XLSX Report Output

The formula for the calculated values is generated by translating the internal form of the WebFOCUS expression (PROFIT/D12.2MC = BUDDOLLARS - DOLLARS;) into an Excel formula. In this example, the formulas appear in cells B8, C8, and D8.

All fields referenced in the calculation should be displayed in the report for a valid formula to be created using cell references. Otherwise, it may be created using values not in the report. If the fields used in the calculation are not present in the report and there is a subsequent RECOMPUTE, the formula created for the RECOMPUTE will not be correct.

```
TABLE FILE GGSALES
ON TABLE SET PAGE-NUM OFF
SUM BUDDOLLARS/I8MC AND DOLLARS/I8MC
COMPUTE PROFIT/D12.2MC = BUDDOLLARS - DOLLARS;
BY REGION
HEADING
"Profit By Region"
" "
ON TABLE COLUMN-TOTAL
ON TABLE PCHOLD FORMAT XLSX FORMULA
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, FONT='ARIAL', SIZE=9, TITLETEXT='By Region', $
TYPE=TITLE, BACKCOLOR=RGB(102 102 102), COLOR=RGB(255 255 255), $
TYPE=HEADING, SIZE=12, STYLE=BOLD, JUSTIFY=CENTER, $
TYPE=GRANDTOTAL, BACKCOLOR=RGB(210 210 210), STYLE=BOLD, $
END
```

The following output highlights the formula that calculates for the column total of PROFIT: D8=SUM(D4:D7).

fx =SUM(D4:D7)				
	A	B	C	D
1	Profit By Region			
2				
3	Region	Budget Dollars	Dollar Sales	PROFIT
4	Midwest	\$11,194,373	\$11,400,665	-\$206,292.00
5	Northeast	\$11,576,921	\$11,392,300	\$184,621.00
6	Southeast	\$11,807,971	\$11,710,379	\$97,592.00
7	West	\$11,641,513	\$11,652,946	-\$11,433.00
8	TOTAL	\$46,220,778	\$46,156,290	\$64,488.00
9				

Example: Generating a Native Excel Formula for a Function

The following example illustrates how functions are translated to Excel reports. The function IMOD divides ACCTNUMBER by 1000 and returns the remainder to LAST3_ACCT. The Excel formula corresponds to =TRUNC((MOD(\$C3,(1000)))). TRUNC is used when the answer returned from an equation is being placed into an Integer field, to be sure there are no decimals.

```
TABLE FILE EMPLOYEE
PRINT ACCTNUMBER AS 'Account Number'
COMPUTE LAST3_ACCT/I3L = IMOD(ACCTNUMBER, 1000, LAST3_ACCT);
BY LAST_NAME AS 'Last Name'
BY FIRST_NAME AS 'First Name'
WHERE (ACCTNUMBER NE 00000000) AND (DEPARTMENT EQ 'MIS');
ON TABLE PCHOLD FORMAT XLSX FORMULA
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, FONT='ARIAL', SIZE=9,$
TYPE=TITLE, BACKCOLOR=RGB(102 102 102), COLOR=RGB(255 255 255), STYLE=BOLD,$
END
```

The output is:

fx =TRUNC((MOD(\$C3,1000)))				
	A	B	C	D
1	Last Name	First Name	Account Number	LAST3_ACCT
2	BLACKWOOD	ROSEMARIE	122850108	108
3	CROSS	BARBARA	163800144	144
4	GREENSPAN	MARY	150150302	302
5	JONES	DIANE	040950036	036
6	MCCOY	JOHN	109200096	096
7	SMITH	MARY	027300024	024
8				

Reference: Generating a Formula With Recomputed Values

If your report contains a calculated value (generated by the COMPUTE or RECOMPUTE command), all of the fields referenced by the calculated value must be displayed in the report in order for cell references to be included in the formula. If a referenced column is not displayed in the workbook, the data value will be placed in the formula, rather than a cell reference. In the case of RECOMPUTE, the value used may be an incorrect value from the last detail record of the sort break.

Example: Generating a Formula With Recomputed Values

The following request computes the difference (DIFF) by subtracting budgeted dollars from dollar sales. The budgeted dollars field used in the expression is not included in the SUM command. The value of DIFF is recomputed on the region level.

```
TABLE FILE GGSales
HEADING
"Profit By Region"
" "
SUM DOLLARS/I8CM
COMPUTE DIFF/I8CM=DOLLARS - BUDDOLLARS;
BY REGION
BY CATEGORY
ON REGION RECOMPUTE
ON TABLE PCHOLD FORMAT XLSX FORMULA
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, FONT='ARIAL', SIZE=9, TITLETEXT='By Region', $
TYPE=TITLE, BACKCOLOR=RGB(102 102 102), COLOR=RGB(255 255 255), $
TYPE=HEADING, SIZE=12, STYLE=BOLD, JUSTIFY=CENTER, $
TYPE=SUBTOTAL, BACKCOLOR=RGB(210 210 210), $
TYPE=GRANDTOTAL, BACKCOLOR=RGB(166 166 166), STYLE=BOLD, $
END
```


The output shows that the formula is subtracting a data value that is not displayed on the worksheet. It is actually the BUDDOLLARS value from the current hardcoded value, since there is no cell reference.

fx =TRUNC(\$C7 -2887620)				
	A	B	C	D
1	Profit By Region			
2				
3	Region	Category	Dollar Sales	DIFF
4	Midwest	Coffee	\$4,178,513	\$92,481
5		Food	\$4,338,271	\$117,550
6		Gifts	\$2,883,881	-\$3,739
7	*TOTAL Midwest		\$11,400,665	\$8,513,045
8	Northeast	Coffee	\$4,164,017	-\$88,445
9		Food	\$4,379,994	-\$73,913
10		Gifts	\$2,848,289	-\$22,263
11	*TOTAL Northeast		\$11,392,300	\$8,521,748
12	Southeast	Coffee	\$4,415,408	-\$16,021
13		Food	\$4,308,731	-\$100,557
14		Gifts	\$2,986,240	\$18,986
15	*TOTAL Southeast		\$11,710,379	\$8,743,125
16	West	Coffee	\$4,473,517	-\$50,446
17		Food	\$4,202,337	\$19,093
18		Gifts	\$2,977,092	\$42,786
19	*TOTAL West		\$11,652,946	\$8,718,640
20	TOTAL		\$46,156,290	-\$64,488
21				

If you add the BUDDOLLARS column to the request, the formula can be recomputed correctly.

SUM DOLLARS/I8MC BUDDOLLARS/I8MC

Generating Native Excel Formulas in XLSX Report Output

The formula generated with the new SUM command contains cell references for both fields used in the calculation.

	A	B	C	D	E
1	Profit By Region				
2					
3	Region	Category	Dollar Sales	Budget Dollars	DIFF
4	Midwest	Coffee	\$4,178,513	\$4,086,032	\$92,481
5		Food	\$4,338,271	\$4,220,721	\$117,550
6		Gifts	\$2,883,881	\$2,887,620	-\$3,739
7	*TOTAL Midwest		\$11,400,665	\$11,194,373	\$206,292
8	Northeast	Coffee	\$4,164,017	\$4,252,462	-\$88,445
9		Food	\$4,379,994	\$4,453,907	-\$73,913
10		Gifts	\$2,848,289	\$2,870,552	-\$22,263
11	*TOTAL Northeast		\$11,392,300	\$11,576,921	-\$184,621
12	Southeast	Coffee	\$4,415,408	\$4,431,429	-\$16,021
13		Food	\$4,308,731	\$4,409,288	-\$100,557
14		Gifts	\$2,986,240	\$2,967,254	\$18,986
15	*TOTAL Southeast		\$11,710,379	\$11,807,971	-\$97,592
16	West	Coffee	\$4,473,517	\$4,523,963	-\$50,446
17		Food	\$4,202,337	\$4,183,244	\$19,093
18		Gifts	\$2,977,092	\$2,934,306	\$42,786
19	*TOTAL West		\$11,652,946	\$11,641,513	\$11,433
20	TOTAL		\$46,156,290	\$46,220,778	-\$64,488
21					

Using XLSX Formula With Prefix Operators

XLSX FORMULA output supports prefix operators that are used on summary lines generated by WebFOCUS commands, such as SUBTOTAL and RECOMPUTE. Where a corresponding formula exists in Excel, these prefix operators are translated into the equivalent Excel summarization formula. The results of prefix operators used directly against retrieved data continue to be passed to Excel as values, not formulas.

The following table identifies the prefix operators supported by XLSX FORMULA when used on summary lines, and the Excel formula equivalent placed in the generated worksheet.

Prefix Operator	Excel Formula Equivalent
SUM.	=SUM()
AVE.	=AVERAGE()
CNT.	=COUNT()

Prefix Operator	Excel Formula Equivalent
MIN.	=MIN()
MAX.	=MAX()

The following prefix operators are not translated to formulas when used on summary lines in XLSX FORMULA.

- ❑ ASQ.
- ❑ FST.
- ❑ LST.

Note:

- ❑ When using a prefix operator on a field specified directly against retrieved data, there is no space between the prefix operator and the field on which it operates.

For example, in the following aggregating display command, the AVE. prefix operator operates on the DOLLARS field.

```
SUM AVE.DOLLARS
```

- ❑ When using a prefix operator on a summary line, you must leave a space between the prefix operator and the aggregated field on which it operates.

In the following summary command, the MAX. prefix operator operates on the DOLLARS field at the REGION sort break. Note the required blank space between the prefix operator and the field name.

```
ON REGION RECOMPUTE MAX. DOLLARS
```

Example: Using a Summary Prefix Operator With FORMAT XLSX FORMULA

In the following request against the GGSales data source, the RECOMPUTE command for the REGION sort field calculates the maximum of the aggregated DOLLARS field and the minimum of the aggregated BUDDOLLARS field.

```
TABLE FILE GGSales
SUM UNITS DOLLARS/I8MC BUDDOLLARS/I8MC
AND COMPUTE DIFF/I8MC= DOLLARS-BUDDOLLARS;
BY REGION
BY CATEGORY
WHERE CATEGORY EQ 'Food' OR 'Coffee'
WHERE REGION EQ 'West' OR 'Midwest'
ON REGION RECOMPUTE MAX. DOLLARS MIN. BUDDOLLARS DIFF
ON TABLE PCHOLD FORMAT XLSX FORMULA
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, FONT='ARIAL', SIZE=9,$
TYPE=TITLE, BACKCOLOR=RGB(102 102 102), COLOR=RGB(255 255 255),$
TYPE=SUBTOTAL, BACKCOLOR=RGB(210 210 210),$
TYPE=GRANDTOTAL, BACKCOLOR=RGB(166 166 166), STYLE=BOLD,$
END
```

In the output, shown in the following image, the cell that represents the recomputed DOLLARS for the Midwest region has been generated as the following formula.

=MIN(E2:E3)

fx		=MIN(E2:E3)				
	A	B	C	D	E	F
1	Region	Category	Unit Sales	Dollar Sales	Budget Dollars	DIFF/I10
2	Midwest	Coffee	332777	\$4,178,513	\$4,086,032	\$92,481
3		Food	341414	\$4,338,271	\$4,220,721	\$117,550
4	*TOTAL Midwest			\$4,338,271	\$4,086,032	\$252,239
5	West	Coffee	356763	\$4,473,517	\$4,523,963	-\$50,446
6		Food	340234	\$4,202,337	\$4,183,244	\$19,093
7	*TOTAL West			\$4,473,517	\$4,183,244	\$290,273
8	TOTAL			\$4,473,517	\$4,086,032	\$387,485
9						

Example: Using a Prefix Operator on a Display Command With FORMAT XLSX FORMULA

In the following request against the GGSALES data source, the CNT., AVE., and PCT. Prefix operators are used in the SUM display command.

```
TABLE FILE GGSALES
SUM UNITS
CNT.UNITS
AVE.UNITS
PCT.UNITS
BY REGION
BY ST
ON TABLE PCHOLD FORMAT XLSX FORMULA
END
```

The output shows that the prefix operators were not passed to Excel as formulas. They were passed as data values.

	A	B	C	D	E	F
1	Region	State	Unit Sales	Unit Sales COUNT	AVE Unit Sales	PCT Unit Sales
2	Midwest	IL	307581	360	854	8
3		MO	297727	359	829	8
4		TX	299737	360	832	8
5	Northeast	CT	302440	360	840	8
6		MA	301909	360	838	8
7		NY	312326	360	867	8
8	Southeast	FL	310302	360	861	8
9		GA	330283	360	917	8
10		TN	294647	360	818	7
11	West	CA	610570	719	849	16
12		WA	321469	359	895	8
13						

Controlling Column Width and Wrapping in XLSX Report Output

- ❑ Column width and data wrapping can be controlled in an Excel worksheet when using FORMAT XLSX.
- ❑ To size the column without wrapping and define the exact size width, use SQUEEZE=ON. If a data value is wider than the specified width of the column, a portion of the data will be hidden from view, but fully visible in the formula bar. You can adjust the column width in Excel after the worksheet has been generated.
- ❑ The default behavior is for all data to wrap within the defined column width. You can also specify the exact width of a column using WRAP=ON.

Syntax: **How to Set Column Width in XLSX Report Output**

`TYPE=REPORT, [COLUMN=column,] SQUEEZE=value, $`

where:

column

Identifies a particular column. If COLUMN is not included in the declaration, default SQUEEZE behavior is applied to the entire report.

value

Is one of the following:

ON

Automatically sizes the columns based on the largest data value in the column. This is the default behavior.

OFF

Sizes the columns based on the maximum size defined for the field in the Master File or Define.

n

Represents a specific numeric value for which the column width can be set. The value represents the measure specified with the UNITS parameter (the default is inches). This is the most commonly used SQUEEZE setting in an XLSX report. This turns off data wrapping.

Note:

- ❑ SQUEEZE can be applied to the entire report by using the ON TABLE SET SQUEEZE ON command.
- ❑ SQUEEZE is not supported for columns created with the OVER phrase or with TABLEF.

Syntax: **How to Wrap Data in XLSX Report Output**

`TYPE=REPORT, [COLUMN=column,] WRAP=value, $`

where:

column

Designates a particular column to apply wrapping behavior to. If COLUMN is not included in the declaration, wrapping will be applied to the entire report.

value

Is one of the following:

ON

Turns on data wrapping. ON is the default value. With this setting, the column width is determined by the client (Excel). Data wraps if it exceeds the width of the column and the row height expands to meet the new height of the wrapped data.

OFF

Turns off data wrapping. Data will not wrap in any cell in the column.

n

Represents a specific numeric value that the column width can be set to. The value represents the measure specified with the UNITS parameter (the default is inches).

This setting implies ON. However, the column width is set to the specified width unless the data is wider than the column width, in which case, wrapping will occur as for ON.

Example: Controlling Column Width and Wrapping in XLSX Report Output

The following example illustrates how to turn on and turn off data wrapping in a column and how to set the column width for a particular column. The UNITS in this example are set to inches (the default).

```
DEFINE FILE GGSALES
PROFIT/D14.3 = BUDDOLLARS-DOLLARS;
DESCRIPTION/A80 = 'Subtract Total Sales Quota from Reported Sales to
calculate profit.';
END

TABLE FILE GGSALES
SUM
DESCRIPTION AS 'DEFAULT'
DESCRIPTION AS 'WRAP = 2'
DESCRIPTION AS 'WRAP = OFF'
DESCRIPTION AS 'SQUEEZE = 1.5'
PROFIT
BY REGION NOPRINT
ON TABLE PCHOLD FORMAT XLSX
ON TABLE SET STYLE *
TYPE=REPORT, COLUMN=DESCRIPTION(2), WRAP=2,$
TYPE=REPORT, COLUMN=DESCRIPTION(3), WRAP=OFF,$
TYPE=REPORT, COLUMN=DESCRIPTION(4), SQUEEZE=1.5,$
END
```

Preserving Leading and Internal Blanks in Report Output

where:

1. The column titled "DEFAULT" illustrates the default column width and wrapping behavior.
2. The column titled "WRAP=2" sets the column width to 2 inches with data wrapping on.
3. The column titled "WRAP=OFF" turns off data wrapping for that column.
4. The column titled "SQUEEZE=1.5" sets the column width to 1.5 inches with data wrapping off.

Since the output spans two pages, the output is shown below in two separate images.

The following output displays the different behavior for the "DEFAULT" and "WRAP=2" columns.

A	B
DEFAULT	WRAP=2
Subtract Total Sales Quota from Reported Sales to calculate profit.	Subtract Total Sales Quota from Reported Sales to calculate profit.
Subtract Total Sales Quota from Reported Sales to calculate profit.	Subtract Total Sales Quota from Reported Sales to calculate profit.
Subtract Total Sales Quota from Reported Sales to calculate profit.	Subtract Total Sales Quota from Reported Sales to calculate profit.
Subtract Total Sales Quota from Reported Sales to calculate profit.	Subtract Total Sales Quota from Reported Sales to calculate profit.
Subtract Total Sales Quota from Reported Sales to calculate profit.	Subtract Total Sales Quota from Reported Sales to calculate profit.

The following output displays the output for the "WRAP=OFF" and "SQUEEZE=1.5" columns.

fx Subtract Total Sales Quota from Reported Sales to calculate profit.		
C	D	E
WRAP=OFF	SQUEEZE=1.5	PROFIT
Subtract Total Sales Quota from Reported Sales to calculate profit.	Subtract Total Sales Quota	-206,292.000
Subtract Total Sales Quota from Reported Sales to calculate profit.	Subtract Total Sales Quota	184,621.000
Subtract Total Sales Quota from Reported Sales to calculate profit.	Subtract Total Sales Quota	97,592.000
Subtract Total Sales Quota from Reported Sales to calculate profit.	Subtract Total Sales Quota	-11,433.000

Preserving Leading and Internal Blanks in Report Output

The SHOWBLANKS command allows you to preserve leading blanks in data cells in XLSX reports.

HEADALIGN=BODY can be used to support trailing blanks in a heading, subheading, footing, and subfooting.

Note: Since XLSX is not HTML-based like EXL2K, setting SHOWBLANKS OFF will not affect internal blanks. All internal blanks will be displayed. This means that by default, in XLSX, all leading blanks in data fields are removed. Because the spacing is different, wrapped fields may display differently in the spreadsheet.

SET SHOWBLANKS Command	XLSX (not HTML-based)	EXL2K (HTML-based)
SET SHOWBLANKS = ON	Leading and additional embedded blanks are preserved.	Leading and additional embedded blanks are preserved.
SET SHOWBLANKS = OFF	Leading blanks are removed, but embedded blanks are respected.	Leading and additional embedded blanks are removed.
HEADALIGN=BODY	If there are leading blanks and HEADALIGN=BODY is set, both leading and trailing blanks will be displayed.	If HEADALIGN=BODY is set, EXL2K will never display the trailing blanks.

Syntax: How to Preserve Leading and Internal Blanks in XLSX Reports

In a FOCEXEC or in a profile, use the following syntax:

```
SET SHOWBLANKS = {OFF|ON}
```

In a request, use the following syntax

```
ON TABLE SET SHOWBLANKS {OFF|ON}
```

where:

OFF

Removes leading blanks and preserves internal blanks in XLSX report output. OFF is the default value.

ON

Preserves leading and internal blanks in XLSX report output. Also preserves trailing blanks in heading, subheading, footing, subfooting lines that use the default heading or footing alignment.

Example: Preserving Leading and Internal Blanks in XLSX Report Output

The following request creates a virtual field that adds leading blanks to the value ACTION, and both leading and internal blanks to the values TRAIN/EX and SCI/FI in the CATEGORY field.

```

SET SHOWBLANKS = OFF
DEFINE FILE MOVIES
NEWCAT/A30 = IF CATEGORY EQ 'ACTION' THEN ' ACTION'
ELSE IF CATEGORY EQ 'SCI/FI' THEN ' SCIENCE FICTION'
ELSE IF CATEGORY EQ 'TRAIN/EX' THEN ' TRAINING EXERCISE'
ELSE IF CATEGORY EQ 'COMEDY' THEN 'COMEDY '
ELSE 'GENERAL';
END
TABLE FILE MOVIES
HEADING
"Example of Excel produced using SET SHOWBLANKS=OFF and XLSX"
ON TABLE SUBHEAD
"<NEWCAT<NEWCAT"
SUM CATEGORY LISTPR/D12.2 COPIES
BY NEWCAT
ON TABLE PCHOLD FORMAT XLSX
ON TABLE SET STYLE *
GRID=OFF,$
TYPE=REPORT, FONT=COURIER NEW,$
TYPE=TABLEHEADING, HEADALIGN=BODY,$
ENDSTYLE
END
    
```

The following reports show the differences in Excel generated using SET SHOWBLANKS = OFF and SET SHOWBLANKS = ON.

SET SHOWBLANKS = OFF with HEADALIGN=BODY

	A	B	C	D
1	TRAINING EXERCISE	TRAINING EXERCISE		
2	Example of Excel produced using SET SHOWBLANKS=OFF and XLSX			
3	NEWCAT	CATEGORY	LISTPR	COPIES
4	ACTION	ACTION	94.82	14
5	SCIENCE FICTION	SCI/FI	114.84	7
6	TRAINING EXERCISE	TRAIN/EX	119.87	10
7	COMEDY	COMEDY	154.80	19
8	GENERAL	MYSTERY	1,216.82	67
9				

SET SHOWBLANKS = OFF without HEADALIGN=BODY

	A	B	C	D
1	TRAINING EXERCISE	TRAINING EXERCISE		
2	Example of Excel produced using SET SHOWBLANKS=OFF without HEADALIGN=BODY			
3	NEWCAT	CATEGORY	LISTPR	COPIES
4	ACTION	ACTION	94.82	14
5	SCIENCE FICTION	SCI/FI	114.84	7
6	TRAINING EXERCISE	TRAIN/EX	119.87	10
7	COMEDY	COMEDY	154.80	19
8	GENERAL	MYSTERY	1,216.82	67
9				

SET SHOWBLANKS = ON with HEADALIGN=BODY

	A	B	C	D
1	TRAINING EXERCISE	TRAINING EXERCISE		
2	Example of Excel produced using SET SHOWBLANKS=ON and XLSX			
3	NEWCAT	CATEGORY	LISTPR	COPIES
4	ACTION	ACTION	94.82	14
5	SCIENCE FICTION	SCI/FI	114.84	7
6	TRAINING EXERCISE	TRAIN/EX	119.87	10
7	COMEDY	COMEDY	154.80	19
8	GENERAL	MYSTERY	1,216.82	67
9				

Support for Drill Downs With XLSX Report Output

Drill downs are supported within the data elements in a report in XLSX format in the same manner as they are supported in EXL2K format. Hyperlink connections can be defined in the StyleSheet declaration of any data column to provide access to any external web source or to execute a FOCEXEC. Drill downs to FOCEXECs can contain data-driven parameters and can generate any of the supported output formats, including XLSX, PDF, HTML, DHTML, and PPT.

Drill downs within text embedded in headings, subheadings, subfootings, and footings will be implemented for XLSX format in a future release.

The JAVASCRIPT and IMAGE drill-down options are not supported with FORMAT XLSX.

Redirection and Excel Drill-Down Reports

The WebFOCUS Client can use redirection when passing the report output to the client application. When redirection is enabled, the WebFOCUS Client saves report output in a temporary directory when a request is executed. Then, an HTTP call is made from the browser to retrieve the temporary stored output for display. When redirection is disabled, the report output is sent directly to the browser without any buffering.

Redirection is disabled by default for the .xlsx file extension because this enables drill downs to run successfully whether the user machine is configured to launch Excel in the browser or as an application outside of the browser.

When redirection is enabled, drill downs within Excel reports will work differently depending on whether the workbook is opened in the browser (only applies to Windows XP) or in the Excel application. For information about launching Excel in the browser or as an application, see [Viewing Excel Workbooks in the Browser vs. the Excel Application](#) on page 6.

- ❑ **For workbooks opened outside the browser in the Excel application:** The redirection context is not retained by the Excel application, so drill-down reports will not have the information required to access the redirected files. The initial workbook will open within Excel, but the target drill-down workbook will not open and you will receive a message stating *You are not allowed to access this viewer file*.
- ❑ **For workbooks opened in the browser (only applies to Windows XP):** Drill downs will work with redirection enabled because the browser session has access to the HTTP header and/or cookies that need to be sent with the HTTP request to the WebFOCUS Client in order to obtain the redirected target workbook file.

Note: For Windows 7, Excel applications no longer display in a browser window.

For additional information about redirection options, see WebFOCUS Administration Console Client Settings described in the *WebFOCUS Security and Administration* manual.

Adding an Image to a Report

WebFOCUS supports the placement of images within each area or node of the report on the worksheet. An image, such as a logo, gives corporate identity to a report, or provides visual appeal. Data specific images can be placed in headers, footers, and data columns to provide additional clarity and style.

The image must reside on the WebFOCUS Reporting Server in a directory named on EDAPATH or APPPATH. If the file is not on the search path, supply the full path name.

Inserting Images Into Excel XLSX Reports

Images can be placed in any available WebFOCUS reporting node or element of a worksheet. Supported image formats include .gif, .png, and .jpg.

Usage Considerations

- ❑ All images will be placed in the top-left corner of the first cell of the defined area, based on the top and left gap. Defined explicit positioning and justification have not been implemented yet.
- ❑ Standard page setting keywords can be used in conjunction with XLSXPAGESETS to control the page layout in standard reports (not compound).
- ❑ Images placed within a report cell in a row or column is anchored to the top-left corner of the cell. The cell is automatically sized to the height and width to fit the largest image (SQUEEZE=ON).
- ❑ Additional lines may need to be added within a heading, footing, subhead, or subfoot to accommodate the placement of the image.

Syntax: How to Insert Images Into WebFOCUS Report Elements in XLSX Reports

```
TYPE={REPORT|heading|data}, IMAGE={url|file|(column)} [,BY=byfield]
[,SIZE=(w h)] ,§
```

where:

REPORT

Embeds an image in the body of a report. The image appears in the background of the report. REPORT is the default value.

heading

Embeds an image in a heading or footing. Valid values are TABHEADING, TABFOOTING, FOOTING, HEADING, SUBHEAD, and SUBFOOT. Provide sufficient blank space in the heading or footing so that the image does not overlap the heading or footing text. You may also want to place heading or footing text to the right of the image using spot markers.

data

Defines a cell within a data column to place the image. Must be used with COLUMNS= attributes to identify the specific report column where the image should be anchored.

url

Is the URL of the image file.

file

Is the name of the image file. It must reside on the WebFOCUS Reporting Server in a directory named on EDAPATH or APPPATH. If the file is not on the search path, supply the full path name. When specifying a GIF file, you can omit the file extension.

Adding an Image to a Report

column

Is an alphanumeric field in the data source that contains the name of an image file. Enclose the column in parentheses (). The field containing the file name or image must be a display field or BY field referenced in the request. Note that the value of the field is interpreted exactly as if it were typed as the URL of the image in the StyleSheet. If you omit the suffix, .GIF is supplied, by default. You can use the SET BASEURL command for supplying the base URL of the images. This way, the value of the field does not have to include the complete URL. This syntax is useful, for example, if you want to embed an image in a SUBHEAD, and you want a different image for each value of the BY field on which the SUBHEAD occurs.

byfield

Is the sort field that generates the subhead or subfoot.

SIZE

Is the size of the image. By default, an image is added at its original size.

w

Is the width of the image, expressed in the unit of measurement specified by the UNITS parameter. Enclose the *w* and *h* values in parentheses. Do not include a comma between them.

h

Is the height of the image, expressed in the unit of measurement specified by the UNITS parameter.

Example: Adding a GIF Image to a Single Table Request

```
DEFINE FILE GGSALES
SHOWCAT/A100=CATEGORY || '.GIF';
END
TABLE FILE GGSALES
SUM DOLLARS/D12CM UNITS/D12C
BY LOWEST CATEGORY NOPRINT
BY SHOWCAT NOPRINT
BY PRODUCT
ACROSS REGION
WHERE CATEGORY NE 'Gifts'
```

```

ON CATEGORY SUBHEAD
" "
"Image in SUBHEAD for Category <CATEGORY "
" "
ON TABLE SUBHEAD
" "
" "
" Report Heading "
" "
ON CATEGORY SUBFOOT
"ON CATEGORY SUBFOOT"





ON TABLE SUBFOOT
"Report Footing"
" "

ON TABLE SET PAGE-NUM NOLEAD
ON TABLE NOTOTAL
ON TABLE SET ACROSSTITLE SIDE
ON TABLE PCHOLD FORMAT XLSX
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, FONT='ARIAL', SIZE=9, TITLETEXT='Food and Coffee',$
TYPE=REPORT, COLUMN=PRODUCT, SQUEEZE=1,$
TYPE=TITLE, BACKCOLOR=RGB(90 90 90), COLOR=RGB(255 255 255), STYLE=BOLD,$
TYPE=ACROSSTITLE, STYLE=BOLD, BACKCOLOR=RGB(90 90 90),
COLOR=RGB(255 255 255),$
TYPE=ACROSSVALUE, BACKCOLOR=RGB(218 225 232), STYLE=BOLD, JUSTIFY=CENTER,$
TYPE=HEADING, STYLE=BOLD, COLOR=RGB(0 35 95), SIZE=12, JUSTIFY=Center,$
TYPE=FOOTING, BACKCOLOR=RGB(90 90 90), SIZE=12, COLOR=RGB(255 255 255),
STYLE=BOLD, JUSTIFY=CENTER,$
TYPE=SUBHEAD, SIZE=12, STYLE=BOLD, BACKCOLOR=RGB(218 225 232),
JUSTIFY=CENTER,$
TYPE=SUBHEAD, IMAGE=(SHOWCAT), SIZE=(.6 .6),$
TYPE=SUBFOOT, SIZE=10, STYLE=BOLD, JUSTIFY=CENTER,$
TYPE=TABHEADING, SIZE=12, STYLE=BOLD, JUSTIFY=CENTER,$
TYPE=TABHEADING, IMAGE=gglogo.gif,$
TYPE=TABFOOTING, SIZE=12, STYLE=BOLD, JUSTIFY=RIGHT,$
TYPE=TABFOOTING, IMAGE=logo.gif, SIZE=(1.67 .6),$
END

```

Adding an Image to a Report

In the following request, since the referenced images are not part of the existing GGSales table, the image files (.gif) are being built in the DEFINE and then referenced in the TABLE request. You can NOPRINT fields if you do not want them to display as columns, but the fields must be referenced in the table to include them in the internal matrix. This will allow the images to be placed in the headings, footings, or data cells. The specific location is defined using StyleSheet definitions for attaching the image based on field value.

	A	B	C	D	E	F	G	H	I									
1																		
2																		
3										Report Heading								
4																		
5	Region	Midwest		Northeast		Southeast		West										
6	Product	Dollar Sales	Unit Sales	Dollar Sales	Unit Sales	Dollar Sales	Unit Sales	Dollar Sales	Unit Sales									
7																		
8										Image in SUBHEAD for Category Coffee								
9																		
10	Capuccino	.	.	\$542,095	44,785	\$944,000	73,264	\$895,495	71,168									
11	Espresso	\$1,294,947	101,154	\$850,107	68,127	\$853,572	68,030	\$907,617	71,675									
12	Latte	\$2,883,566	231,623	\$2,771,815	222,866	\$2,617,836	209,654	\$2,670,405	213,920									
13	ON CATEGORY SUBFOOT																	
14																		
15										Image in SUBHEAD for Category Food								
16																		
17										Biscotti	\$1,091,727	86,105	\$1,802,005	145,242	\$1,505,717	119,594	\$863,868	70,436
18	Croissant	\$1,751,124	139,182	\$1,670,818	137,394	\$1,902,359	156,456	\$2,425,601	197,022									
19	Scone	\$1,495,420	116,127	\$907,171	70,732	\$900,655	73,779	\$912,868	72,776									
20	ON CATEGORY SUBFOOT																	
21								Report Footing										
22																		
23																		
24																		

Example: Adding a GIF Image to a Compound Request

Note: Compound Layout syntax cannot contain hidden carriage return or line feed characters. For purposes of presenting the example in this Technical Memo, line feed characters have been added so that the sample code wraps to fit within the printed page. To run this example in your environment, copy the code into a text editor and delete any line feed characters within the Compound Layout syntax by going to the end of each line and pressing *Delete*. In some instances, you may need to add a space to maintain the structure of the string.

```
APP PATH IBISAMP
SET HTMLARCHIVE=ON
*-HOLD_SOURCE
COMPOUND LAYOUT PCHOLD FORMAT XLSX
UNITS=IN,$
SECTION=section1, LAYOUT=ON, METADATA='prop_with_names, Margins_Left=0.5,
Margins_Top=0.5, Margins_Right=0.5, Margins_Bottom=0.5,
thumbnailscale=4', MERGE=OFF, ORIENTATION=LANDSCAPE, PAGESIZE=Legal,
SHOW_GLOBALFILTER=OFF,$
PAGELAYOUT=1, NAME='Page layout 1', text='Page layout 1', TOC-LEVEL=1,
BOTTOMMARGIN=0.5, TOPMARGIN=0.5, METADATA='BOTTOMMARGIN=0.5, TOPMARGIN=0.5,
LEFTMARGIN=0,RIGHTMARGIN=0,', $
COMPONENT='report1', TEXT='report1', TOC-LEVEL=2, POSITION=(0.650 0.917),
DIMENSION=(7.250 3.000), BYTOC=0, ARREPORTSIZE=DIMENSION,
METADATA='left: 0.65in; top: 0.917in; width: 7.25in; height: 3in;
position: absolute; z-index: 1;', $
COMPONENT='chart1', TEXT='chart1', TOC-LEVEL=2, POSITION=(0.735 4.332),
DIMENSION=(7.167 2.917), COMPONENT-TYPE=GRAPH, ARREPORTSIZE=DIMENSION,
METADATA='left: 0.735in; top: 4.332in; width: 7.167in; height: 2.917in;
position: absolute; z-index: 2;', $
END

SET COMPONENT='report1'
-*component_type report
DEFINE FILE GGSales
SHOWCAT/A100=CATEGORY || '.GIF';
SHOWDATEQ/Q=DATE;
SHOWDATEY/YY=DATE;
SHOWDATEQY/YYQ=DATE;
END
```

Adding an Image to a Report







```
TABLE FILE GGSALES
SUM DOLLARS/D12CM AS 'Dollars'
BY REGION AS ''
BY LOWEST CATEGORY
BY SHOWCAT AS 'Data Image'
ACROSS SHOWDATEY AS ''
ACROSS SHOWDATEQ AS ''
WHERE REGION NE 'Midwest' OR 'West'
ON TABLE SET HIDENULLACRS ON
HEADING
" "
"Image in Page Heading "
ON REGION SUBHEAD
" <+0> SUBHEAD: <REGION"
FOOTING
" "
"Image in Page Footing"
ON TABLE SET PAGE-NUM NOLEAD
ON TABLE NOTOTAL
ON TABLE PCHOLD FORMAT XLSX
ON TABLE SET STYLE *
TYPE=REPORT, SIZE=10, BACKCOLOR=NONE, GRID=OFF, FONT='ARIAL',
XLSXPAGESETS=ON, TOPMARGIN=1, BOTTOMMARGIN=1, ORIENTATION=LANDSCAPE,
PAGESIZE=LEGAL, TITLETEXT='With Images',$
TYPE=REPORT, COLUMN=REGION, SQUEEZE=1.5, JUSTIFY=CENTER,$
TYPE=DATA, BACKCOLOR=NONE,$
TYPE=DATA, COLUMN=SHOWCAT, IMAGE=(SHOWCAT), SIZE=(.5 .5),$
TYPE=TITLE, BACKCOLOR=RGB(218 225 232), BORDER=LIGHT,
STYLE=-UNDERLINE+BOLD,$
TYPE=HEADING, IMAGE=GGLOGO.GIF, SIZE=(.65 .65),$
TYPE=HEADING, SIZE=12, STYLE=BOLD, JUSTIFY=CENTER,$
TYPE=SUBHEAD, SIZE=10, STYLE=BOLD, BORDER-TOP=LIGHT,$
TYPE=SUBHEAD, BY=1, JUSTIFY=CENTER, BORDER-TOP=LIGHT,$
TYPE=SUBFOOT, STYLE=BOLD,$
TYPE=FOOTING, SIZE=12, STYLE=+BOLD, JUSTIFY=CENTER,$
TYPE=FOOTING, IMAGE=logo.gif, SIZE=(1.67 .6),$
TYPE=ACROSS, JUSTIFY=CENTER, BORDER=LIGHT,$
TYPE=ACROSSTITLE, STYLE=-UNDERLINE+BOLD,$
TYPE=ACROSSVALUE, BACKCOLOR=RGB(218 225 232), STYLE=-UNDERLINE+BOLD,$
END
```

```
SET COMPONENT='chart1'  
ENGINE INT CACHE SET ON  
-DEFAULTH &WF_STYLE_UNITS='PIXELS';  
-DEFAULTH &WF_STYLE_HEIGHT='1005.0';  
-DEFAULTH &WF_STYLE_WIDTH='1070.0';  
-DEFAULTH &WF_TITLE='WebFOCUS Report';  
GRAPH FILE GGSALES  
HEADING  
"Sales Graph"  
SUM  
GGSALES.SALES01.DOLLARS  
BY SHOWDATEY AS Year  
BY GGSALES.SALES01.REGION  
ON GRAPH PCHOLD FORMAT XLSX  
ON GRAPH SET VZERO OFF  
ON GRAPH SET HTMLENCODE ON  
ON GRAPH SET GRAPHDEFAULT OFF  
ON GRAPH SET GRWIDTH 1  
ON GRAPH SET UNITS &WF_STYLE_UNITS  
ON GRAPH SET HAXIS 1000  
ON GRAPH SET VAXIS 1000  
ON GRAPH SET GRMERGE ADVANCED  
ON GRAPH SET GRLEGEND 0  
ON GRAPH SET GRXAXIS 2  
ON GRAPH SET LOOKGRAPH HBAR  
ON GRAPH SET STYLE *
```

Adding an Image to a Report

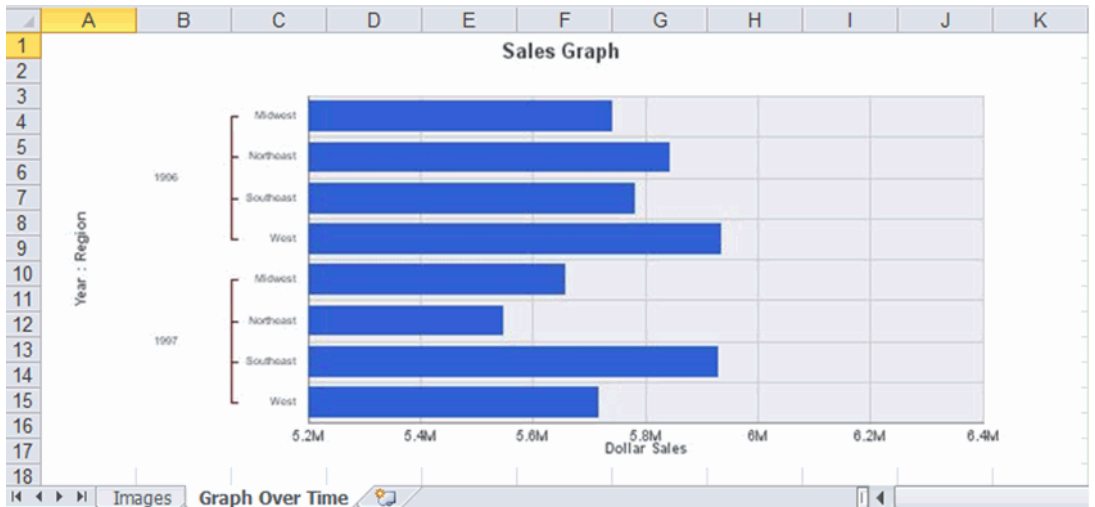
```
*GRAPH_SCRIPT
setPieDepth(0);
setPieTilt(0);
setDepthRadius(0);
setCurveFitEquationDisplay(false);
setPlace(true);
setPieFeelerTextDisplay(1);
setUseSeriesShapes(true);
setMarkerSizeDefault(50);
setScaleMustIncludeZero(getX1Axis(), false);
setScaleMustIncludeZero(getY1Axis(), false);
setScaleMustIncludeZero(getY2Axis(), false);
setMarkerSizeDefault(60);
*END
INCLUDE=endeflt.sty,$
TYPE=REPORT, TITLETEXT='Graph Over Time',$
*GRAPH_SCRIPT
setReportParsingErrors(false);
setSelectionEnableMove(false);
*END
ENDSTYLE
END
COMPOUND END
```

The output is shown in the following images.

Image in Page Heading										
			1996				1997			
			Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
Category	Data Image	SUBHEAD: Northeast								
Northeast	Coffee		\$560,057	\$489,050	\$528,299	\$559,488	\$503,629	\$565,503	\$501,884	\$456,107
	Food		\$529,237	\$535,075	\$635,881	\$546,154	\$492,300	\$523,784	\$566,931	\$550,632
	Gifts		\$369,301	\$327,843	\$385,484	\$378,030	\$317,487	\$345,227	\$367,737	\$357,180
SUBHEAD: Southeast										
Southeast	Coffee		\$517,360	\$528,692	\$577,773	\$478,972	\$620,073	\$515,993	\$598,251	\$578,294
	Food		\$539,876	\$540,733	\$523,482	\$596,651	\$556,701	\$528,073	\$499,708	\$523,507
	Gifts		\$372,652	\$375,837	\$364,880	\$364,858	\$374,260	\$329,274	\$378,170	\$426,309

Information Builders The Standard for Enterprise Business Intelligence

Image in Page Footing








Example: Adding a GIF Image to a BYTOC Compound Request

The following syntax is a portion of the code from the previous example to show the COMPOUND BYTOC syntax. By adding the ON TABLE SET COMPOUND BYTOC command to the compound report above, you can turn the report into a Compound Table of Contents report. The BYTOC syntax can be added to a stand-alone request or to a component of a compound document.

```
TABLE FILE GGSALES
SUM DOLLARS/D12CM AS 'Dollars'
SHOWREG
NOPRINT
BY REGION AS ''
BY LOWEST CATEGORY
BY SHOWCAT AS 'Data Image'
WHERE REGION NE 'Midwest' OR 'West'
ACROSS SHOWDATEY AS '' ACROSS SHOWDATEQ AS ''
ON TABLE SET HIDENULLACRS ON HEADING
"Image in Page Heading" ON REGION SUBHEAD
"<+0> Image in SUBHEAD:<REGION" FOOTING
" "
"Image in Page Footing"
ON TABLE SET PAGE-NUM NOLEAD ON TABLE NOTOTAL
ON TABLE PCHOLD FORMAT XLSX ON TABLE SET COMPOUND BYTOC
ON TABLE SET STYLE *
```

The output is:

	A	B	C	D	E	F	G	H	I	J	K
1											
2	Image in Page Heading										
3											
4											
5											
6			1996				1997				
6		Category	Data Image	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
7	SUBHEAD: Northeast										
8	Northeast	Coffee		\$560,057	\$489,050	\$528,299	\$559,488	\$503,629	\$565,503	\$501,884	\$456,107
9		Food		\$529,237	\$535,075	\$635,881	\$546,154	\$492,300	\$523,784	\$566,931	\$550,632
10		Gifts		\$369,301	\$327,843	\$385,484	\$378,030	\$317,487	\$345,227	\$367,737	\$357,180
11											
12											
13			Image in Page Footing								
14											
15											

Inserting Images Into XLSX Workbook Headers and Footers

WebFOCUS supports the insertion of images into Excel headers and footers and the definition of key page settings to support the placement of these images in relationship to the overall worksheet and the Excel generated page breaks. This new access to the Excel page functionality is designed to enhance overall usability of the worksheets for users who will be printing these reports. Page settings including orientation, page size, and page margins will directly affect the layout of each Excel page based on values defined within the FOCEXEC. Images can be included on headers and footers on every printed page, on the first page of the report only, or only on all subsequent pages. The WebFOCUS headings and footings continue to display within the worksheet. With this new feature, WebFOCUS can insert logos to be printed once at the top of a report and watermark images that need to be displayed on every printed page.

Syntax: How to Define Excel Page Settings

```
[TYPE=REPORT, ] XLSXPAGESETS={ON|OFF} [ ,PAGESIZE={pagesize|LETTER} ]
[ ,ORIENTATION={PORTRAIT|LANDSCAPE} ] [ ,TOPMARGIN=n] [ ,BOTMOMMARGIN=m], $
```

where:

`XLSXPAGESETS={ON|OFF}`

ON causes the page settings defined in the WebFOCUS request to be applied to the Excel worksheet page settings. OFF retains the default page settings defined in the standard Excel workbook. OFF is the default value.

n

Defines the top margin for the worksheet in the units identified by the UNITS parameter (inches, by default). The default value is .25.

m

Defines the bottom margin for the worksheet in the units identified by the UNITS parameter (inches, by default). The default value is .25.

pagesize

Is one of the PAGESIZE values supported in a WebFOCUS StyleSheet, as described in the *Creating Reports With WebFOCUS Language* manual. LETTER is the default page size.

`PORTRAIT|LANDSCAPE`

PORTRAIT displays the report across the narrower dimension of a vertical page, producing a page that is longer than it is wide. PORTRAIT is the default value.

LANDSCAPE displays the report across the wider dimension of a horizontal page, producing a page that is wider than it is long.

Syntax: **How to Insert Images Into Excel Headers and Footers**

```
TYPE={PAGEHEADER|PAGEFOOTER}, OBJECT=IMAGE,  
IMAGE=imagename, JUSTIFY={LEFT|CENTER|RIGHT}  
[, DISPLAYON={FIRST|NOT-FIRST}] [, SIZE=(w h)], $
```

where:

PAGEHEADER

Places the image in the worksheet header.

PAGEFOOTER

Places the image in the worksheet footer.

imagename

Is the name of a valid image file to be placed in the header or footer. The image must be located in the defined application path on the Reporting Server. The image types supported are GIF and JPEG.

JUSTIFY={LEFT|CENTER|RIGHT}

Identifies the area in the header or footer to contain the image and the justification or placement within that defined area.

DISPLAYON

Defines whether the image should be placed on the first page only or on all pages except the first. Omit this attribute to place the image on all pages.

Valid values are:

FIRST places the image only on the first page.

NOT-FIRST places the image on every page, except the first page.

SIZE=(*w h*)

Is the size of the image. By default, an image is added at its original size.

w is the width of the image, expressed in the unit of measurement specified by the UNITS parameter.

h is the height of the image, expressed in the unit of measurement specified by the UNITS parameter.

Example: Inserting Images in Excel Headers and Footers and Defining Page Settings

The following request against the GGSALES data source places the image ibi_logo.gif on the left header area of the first page and the right header area of every subsequent page of the resulting worksheet. It places the image webfocus1.gif in the center area of the footer on every page.

```
TABLE FILE GGSALES
SUM DOLLARS UNITS BUDDOLLARS BUDUNITS
BY REGION
BY ST
BY CATEGORY
BY PRODUCT
ON TABLE SET BYDISPLAY ON
ON TABLE PCHOLD FORMAT XLSX
ON TABLE SET STYLE *
FONT=ARIAL,SIZE=12,
XLSXPAGESETS=ON, TOPMARGIN=1, BOTTOMMARGIN=1, ORIENTATION=LANDSCAPE,
PAGE SIZE=LETTER,$
TYPE=TITLE, COLOR=WHITE, BACKCOLOR=GREY,$
TYPE=PAGEHEADER, OBJECT=IMAGE, JUSTIFY=LEFT, IMAGE=IBI_LOGO.GIF,
DISPLAYON=FIRST,$
TYPE=PAGEHEADER, OBJECT=IMAGE, JUSTIFY=RIGHT, IMAGE=IBI_LOGO.GIF,
DISPLAYON=NOT-FIRST,$
TYPE=PAGEFOOTER, OBJECT=IMAGE, JUSTIFY=CENTER, IMAGE=WEBFOCUS1.GIF,$
END
```

Inserting Images Into XLSX Workbook Headers and Footers

The first page of output has the image `ibilogo.gif` in the left area of the header and the image `webfocus1.gif` in the center area of the footer.



Region	State	Category	Product	Dollar Sales	Unit Sales	Budget Dollars	Budget Units
Mid west	IL	Coffee	Espresso	420439	32237	401477	32416
Mid west	IL	Coffee	Latte	978340	77344	964787	79015
Mid west	IL	Food	Biscotti	378412	29413	385369	30001
Mid west	IL	Food	Croissant	549366	43300	528255	43271
Mid west	IL	Food	Scone	595089	45355	567231	45091
Mid west	IL	Gifts	Coffee Grinder	233292	19339	241711	19224
Mid west	IL	Gifts	Coffee Pot	204828	15785	208255	16035
Mid west	IL	Gifts	Mug	376754	30157	388612	30881
Mid west	IL	Gifts	Thermos	187901	14651	181159	14137
Mid west	MO	Coffee	Espresso	419143	32596	416875	32787
Mid west	MO	Coffee	Latte	966981	77347	921336	77141
Mid west	MO	Food	Biscotti	368077	29188	360403	28764
Mid west	MO	Food	Croissant	613871	48941	592609	49327
Mid west	MO	Food	Scone	481953	37602	478691	36573
Mid west	MO	Gifts	Coffee Grinder	181570	14614	171501	14779
Mid west	MO	Gifts	Coffee Pot	190153	14807	191451	14970
Mid west	MO	Gifts	Mug	343852	27040	324488	26837
Mid west	MO	Gifts	Thermos	195686	15592	189484	15903
Mid west	TX	Coffee	Espresso	455365	36321	439880	36666
Mid west	TX	Coffee	Latte	938245	76932	941677	77501
Mid west	TX	Food	Biscotti	345238	27504	321857	27074
Mid west	TX	Food	Croissant	587887	46941	587869	47050
Mid west	TX	Food	Scone	418398	33170	398437	32112
Mid west	TX	Gifts	Coffee Grinder	204292	16440	200241	16625
Mid west	TX	Gifts	Coffee Pot	204897	16564	214301	16774
Mid west	TX	Gifts	Mug	366337	29521	383050	29374
Mid west	TX	Gifts	Thermos	194319	16344	193367	16779
Northeast	CT	Coffee	Capuccino	158995	12386	141574	11098
Northeast	CT	Coffee	Espresso	279373	22482	299854	23676
Northeast	CT	Coffee	Latte	926052	74623	953855	74427
Northeast	CT	Food	Biscotti	589355	46214	587501	46404



The second page of output has the image `ibilogo.gif` in the right area of the header and the image `webfocus1.gif` in the center area of the footer.



Northeast	CT	Food	Croissant	551489	45847	580168	46335
Northeast	CT	Food	Scone	283874	22378	269221	21038
Northeast	CT	Gifts	Coffee Grinder	169908	13691	159620	13117
Northeast	CT	Gifts	Coffee Pot	208209	15523	197051	15190
Northeast	CT	Gifts	Mug	392967	31728	424333	32415
Northeast	CT	Gifts	Thermos	221827	17568	219025	17667
Northeast	MA	Coffee	Capuccino	174344	15358	192747	15672
Northeast	MA	Coffee	Espresso	248356	19698	254310	19888
Northeast	MA	Coffee	Latte	917737	74572	941438	73874
Northeast	MA	Food	Biscotti	570391	47064	616766	48246
Northeast	MA	Food	Croissant	497234	41029	519322	41351
Northeast	MA	Food	Scone	332486	25363	312004	23774
Northeast	MA	Gifts	Coffee Grinder	177940	14382	187686	15384
Northeast	MA	Gifts	Coffee Pot	184119	15349	184071	15171
Northeast	MA	Gifts	Mug	401944	32360	401617	31324
Northeast	MA	Gifts	Thermos	203435	16734	208436	16921
Northeast	NY	Coffee	Capuccino	208756	17041	227170	17662
Northeast	NY	Coffee	Espresso	322378	25947	318738	26212
Northeast	NY	Coffee	Latte	928026	73671	922776	73411
Northeast	NY	Food	Biscotti	642259	51964	644415	50502
Northeast	NY	Food	Croissant	622095	50518	640032	50178
Northeast	NY	Food	Scone	290811	22991	284478	23603
Northeast	NY	Gifts	Coffee Grinder	161352	12904	164336	12796
Northeast	NY	Gifts	Coffee Pot	198452	15313	192227	15043
Northeast	NY	Gifts	Mug	349300	27409	344364	26801
Northeast	NY	Gifts	Thermos	178836	14568	187786	15179
Southeast	FL	Coffee	Capuccino	317027	24143	285194	23092
Southeast	FL	Coffee	Espresso	256539	19730	236531	18690
Southeast	FL	Coffee	Latte	889887	71123	886465	72975
Southeast	FL	Food	Biscotti	511597	40606	516984	41242
Southeast	FL	Food	Croissant	602076	50175	644884	51437
Southeast	FL	Food	Scone	311836	24543	299547	24576



Reference: Usage Notes for Inserting Images Into XLSX Worksheet Headers and Footers

- ❑ The Excel headers and footers are not automatically sized based on contents of the areas. Define page margins within the page settings (XLSPAGESETS) to account for the space required to display the images within each page of the report.
- ❑ The image sizing based on the specified height and width is not proportional. Sizing may cause image distortion.
- ❑ BLOB image fields are not supported in this release.
- ❑ Compound reports are not supported in this release.

Reference: Displaying Watermarks on XLSX Report Output

Watermark images can be placed into the Excel headers to display on every printed page of the generated worksheet.

Excel places images on the page starting in the header from left to right and then the footer from left to right. Large images placed in the header may overlap images before them in the presentation order. For page layouts with a logo in the left area and watermark centered on the page, watermark image background must be transparent so it does not overlay the logo image.

In Excel, images are placed first on the page. All other contents of the worksheet are then placed on top of the images. Text in cells and styling, such as background color and drawing objects, are placed on top of the images. Excel supports transparency in drawing objects and images, but not in cell background color. BACKCOLOR will cover over images placed on the page.

Example: Placing a Watermark in an XLSX Header

The following request against the GGSales data source uses the image internaluseonly.gif as a watermark to display in the background of every page of the worksheet. Although the image is placed in the center area of the header, it is large enough to span the entire worksheet page. It has a transparent background, so it does not cover the logo images placed at the left in the header and the center in the footer.

```
TABLE FILE GGSales
SUM DOLLARS UNITS BUDDOLLARS BUDUNITS
BY REGION
BY ST
BY CATEGORY
BY PRODUCT
ON TABLE SET BYDISPLAY ON
ON TABLE PCHOLD FORMAT XLSX
ON TABLE SET STYLE *
XLSXPAGESETS=ON,
TOPMARGIN=1,BOTTOMMARGIN=1,LEFTMARGIN=1, RIGHTMARGIN=1,
ORIENTATION=LANDSCAPE,PAGESIZE=LETTER,$
TYPE=PAGEHEADER, OBJECT=IMAGE, JUSTIFY=LEFT, IMAGE=IBI_LOGO.GIF,
DISPLAYON=FIRST,$
TYPE=PAGEHEADER, OBJECT=IMAGE, JUSTIFY=CENTER, IMAGE=WFINTERNALUSEONLY.GIF,$
TYPE=PAGEFOOTER, OBJECT=IMAGE, JUSTIFY=RIGHT, IMAGE=WEBFOCUS1.GIF,$
END
```

The first page of the generated worksheet shows the watermark image beneath the data. This image is displayed on every page of the worksheet.

Region	State	Category	Product	Dollar Sales	Unit Sales	Budget Dollars	Budget Units
Midwest	IL	Coffee	Espresso	420439	32237	401477	32416
Midwest	IL	Coffee	Latte	978340	77344	964787	79015
Midwest	IL	Food	Biscotti	378412	29413	385369	30001
Midwest	IL	Food	Croissant	549366	43300	528255	43271
Midwest	IL	Food	Scone	595069	45355	567231	45091
Midwest	IL	Gifts	Coffee Grinder	233292	19339	241711	19224
Midwest	IL	Gifts	Coffee Pot	204828	15785	208255	16035
Midwest	IL	Gifts	Mug	376754	30157	388629	30881
Midwest	IL	Gifts	Thermos	187901	14651	18115	14137
Midwest	MO	Coffee	Espresso	419143	32596	410075	32787
Midwest	MO	Coffee	Latte	966981	77347	921336	77141
Midwest	MO	Food	Biscotti	368077	29184	360403	28764
Midwest	MO	Food	Croissant	613871	48911	592609	49327
Midwest	MO	Food	Scone	481953	38562	478691	36573
Midwest	MO	Gifts	Coffee Grinder	181572	14614	171501	14779
Midwest	MO	Gifts	Coffee Pot	17115	14807	191451	14970
Midwest	MO	Gifts	Mug	193852	27040	324488	26837
Midwest	MO	Gifts	Thermos	196686	15592	189484	15903
Midwest	TX	Coffee	Espresso	455365	36321	439880	36666
Midwest	TX	Coffee	Latte	938245	76932	941677	77501
Midwest	TX	Food	Biscotti	345238	27504	321857	27074
Midwest	TX	Food	Croissant	587887	46941	587869	47050
Midwest	TX	Food	Scone	418398	33170	398437	32112
Midwest	TX	Gifts	Coffee Grinder	204292	16440	200241	16625
Midwest	TX	Gifts	Coffee Pot	204897	16564	214301	16774
Midwest	TX	Gifts	Mug	366337	29521	383050	29374
Midwest	TX	Gifts	Thermos	194319	16344	193367	16779
Northeast	CT	Coffee	Capuccino	158995	12386	141574	11098
Northeast	CT	Coffee	Espresso	279373	22482	299854	23676
Northeast	CT	Coffee	Latte	926052	74623	953855	74427
Northeast	CT	Food	Biscotti	589355	46214	587501	46404
Northeast	CT	Food	Croissant	551489	45847	580168	46335
Northeast	CT	Food	Scone	283874	22378	269221	21038
Northeast	CT	Gifts	Coffee Grinder	169908	13691	159620	13117
Northeast	CT	Gifts	Coffee Pot	208209	15523	197051	15190
Northeast	CT	Gifts	Mug	392967	31728	424333	32415

Creating Excel XLSX Worksheets Using Templates

XLSX report output can be generated based on Excel templates. This feature allows for the integration of WebFOCUS reports into workbooks containing multiple worksheets. Any native Excel template can be used to generate a new workbook containing a WebFOCUS report.

The following Excel file types can be used as template files to generate XLSX workbooks.

Template File Type	Output Workbook Generated
Template (.xltx)	Workbook (.xlsx)
Macro-Enabled Template (.xlsm)	Macro-Enabled workbook (.xlsm)

Template File Type	Output Workbook Generated
Workbook (.xlsx)	Workbook (.xlsx)
Macro-Enabled workbook (.xlsm)	Macro-Enabled workbook (.xlsm)

WebFOCUS XLSX TEMPLATE format provides support for basic Excel templates (.xltx) files. These templates cannot contain macros or other content that Microsoft considers active, as well as templates with active content (XLTM/XLSM). Additionally, macro-enabled templates (.xlsm) allow for the inclusion of active content (macros and VB script) into templates.

A WebFOCUS EXL07 template procedure generates a native Excel workbook with the standard Excel extension, based on the defined template file. The WebFOCUS request will replace an existing worksheet within the template workbook, and any formulas or references defined in other worksheets to cells within the replaced worksheet will automatically update when the workbook is opened.

Since the template feature replaces existing worksheets, the designated worksheet must exist in the template workbook. Any content on the replaced worksheet within the template will not be retained. Content contained on any other worksheets will be retained and updated.

Named ranges can be defined within the procedure using the INRANGES attribute to designate cell groupings that can be referenced by other worksheets.

An Excel 2007/2010 template can be generated by saving any workbook with the .xltx extension. The template file should be stored within your application path (EDAPATH or APP PATH) rather than the default Excel template directory so that it can be accessed by the Reporting Server when the procedure is executed.

The EXL07 TEMPLATE feature is supported for basic EXL07 format reports. The following features are not supported with EXL07 TEMPLATE in this release: PIVOT, BYTOC, FORMULA, EXL97, EXCEL, and compound Excel reports.

In most cases, existing Excel 2003/2000 templates created as .mht files can easily be converted to Excel 2013/2010/2007 templates by opening the .mht file in Excel 2013/2010/2007 and resaving the file as either an Excel template (.xltx) or a macro enabled (.xlsm) file. Native Excel formulas and functionality should be retained within these templates. Use .xlsm to retain active content, including macros. This new XLTX template can be used with XLSX procedures.

Note: The XLSX template support introduced in WebFOCUS Release 7.7.02 has been expanded to support macro-enabled templates as of WebFOCUS Release 7.7.03 HF5 and higher with WebFOCUS Reporting Server Release 7.7.04. This requires a manual update to the mime.wfs file.

Syntax: How to Create an XLSX Report Using Any Supported Template File Type

To support the expanded template files types, the template file name attribute has been enhanced to allow for the inclusion of the file extension. If no extension is specified within the template name, the file extension will default to .XLTX.

```
ON TABLE PCHOLD FORMAT XLSX TEMPLATE template_name SHEETNUMBER n
```

where:

template_name

Is the name of the Excel template file (workbook), including the file name and extension, residing on the WebFOCUS Reporting Server application directory search path. For example, IPOLICY.XLTX, PRINTSHEETS.XLTM, or DASHBOARD.XLSM. If the extension is not provided, it defaults to .XLTX.

n

Is the number of the existing Excel worksheet being replaced in the template file (workbook).

Reference: Usage Notes for XLSX Templates

The workbook template used by the WebFOCUS procedure must contain valid worksheets.

- ❑ The sheet that is updated must exist in the workbook, as WebFOCUS is replacing the worksheet rather than inserting a worksheet. If the sheet designated does not exist, the procedure will return an error.
- ❑ In any template file, at least one of the sheets in the workbook must contain a cell with a valid value (blank or any other value). To replace a worksheet in a template that contains only empty worksheets, replace one of the cells in any of the sheets with a space and save. This will instantiate the worksheets so they are accessible to WebFOCUS for updating.

Creating Excel Table of Contents Reports

As of WebFOCUS Release 7.7.03, the Excel Table of Contents feature is supported.

Excel Table of Contents (BYTOC) enables you to generate a separate worksheet within an instance of the report for each value of the first BY field in the WebFOCUS report.

Syntax: **How to Use the Excel Table of Contents Feature**

There are three different ways that BYTOC can be invoked:

```
ON TABLE {HOLD|PCHOLD} FORMAT XLSX BYTOC
SET COMPOUND=BYTOC
ON TABLE SET COMPOUND BYTOC
```

Since a BYTOC report generates separate worksheets according to the value of the first BY field in the report, the report must contain at least one BY field. The primary BY field may be a NOPRINT field.

The BYTOC feature is not supported with the XLSX TEMPLATE format.

Example: **Creating a Simple BYTOC Report**

The following request against the GGSALES data source creates separate tabs based on the REGION sort field.

```
TABLE FILE GGSALES
SUM UNITS/D12C DOLLARS/D12CM
BY REGION NOPRINT
BY CATEGORY
BY PRODUCT
HEADING
"<REGION Region Sales"
ON TABLE PCHOLD FORMAT XLSX
ON TABLE SET BYDISPLAY ON
ON TABLE SET COMPOUND BYTOC
ON TABLE SET STYLE *
TYPE=REPORT, FONT=ARIAL, SIZE=9,$
TYPE=HEADING, SIZE=12,$
TYPE=TITLE, BACKCOLOR=GREY, COLOR=WHITE,$
ENDSTYLE
END
```


The output is:

	A	B	C	D
1	Midwest Region Sales			
2	Category	Product	Unit Sales	Dollar Sales
3	Coffee	Espresso	101,154	\$1,294,947
4	Coffee	Latte	231,623	\$2,883,566
5	Food	Biscotti	86,105	\$1,091,727
6	Food	Croissant	139,182	\$1,751,124
7	Food	Scone	116,127	\$1,495,420
8	Gifts	Coffee Grinder	50,393	\$619,154
9	Gifts	Coffee Pot	47,156	\$599,878
10	Gifts	Mug	86,718	\$1,086,943
11	Gifts	Thermos	46,587	\$577,906
12				

Midwest Northeast Southeast West

Reference: How to Name Worksheets

- ❑ The worksheet tab names are the BY field values that correspond to the data on the current worksheet. If the user specifies the TITLETEXT keyword in the StyleSheet, it will be ignored.
- ❑ Excel limits the length of worksheet titles to 31 characters. The following special characters cannot be used: ':', '?', '*', and '/'.
- ❑ If you want to use date fields as the bursting BY field, you can include the - character instead of the / character. The - character is valid in an Excel tab title. However, if you do use the / character, WebFOCUS will substitute it with the - character.

Naming XLSX Worksheets With Case Sensitive Data

Excel requires each sheet name to be unique. Excel is case insensitive meaning it evaluates two values as being the same when the values contain the same characters but have different casing. For example, Excel evaluates the values WEST and West to be the same value. WebFOCUS XLSX format identifies duplicate names and adds a unique number to the name to allow Excel to maintain both sheets.

By default, WebFOCUS sort processing is case-sensitive, so the same field value with different casing is considered to be two different values when used as a sort (BY) field. In an Excel BYTOC report, WebFOCUS will generate sheets with sheet names for each value of the primary sort (BY) key based on case sensitivity. To account for this, XLSX has been enhanced to add counters where duplicate tab names are found in the data to ensure the names are unique.

For example, if the report had EAST and East as the values for the Region, each worksheet would be displayed as EAST(1) and East(2), as shown in the following image.

	A	B	C	D
1	East Region Sales			
2	Category	Product	Unit Sales	Dollar Sales
3	Coffee	Capuccino	73,264	\$944,000
4	Coffee	Espresso	68,030	\$853,572
5	Coffee	Latte	209,654	\$2,617,836
6	Food	Biscotti	119,594	\$1,505,717
7	Food	Croissant	156,456	\$1,902,359
8	Food	Scone	73,779	\$900,655
9	Gifts	Coffee Grinder	47,083	\$605,777
10	Gifts	Coffee Pot	49,922	\$645,303
11	Gifts	Mug	88,474	\$1,102,703
12	Gifts	Thermos	48,976	\$632,457
13				
14				

Overcoming the Excel 2007/2010 Row Limit Using Overflow Worksheets

The maximum number of rows supported by Excel 2007/2010 on a worksheet is 1,048,576 (1MB). When you create an XLSX output file from a WebFOCUS report, the number of rows generated can be greater than this maximum.

To avoid creating an incomplete output file, you can have extra rows flow onto a new worksheet, called an *overflow worksheet*. The name of each overflow worksheet will be the name of the original worksheet appended with an increment number.

In addition, when the overflow worksheet feature is enabled, you can set a target value for the maximum number of rows to be included on a worksheet. By default, the row limit will be set to the default value for the LINES parameter (57).

Note: By default, when generating XLSX output, the WebFOCUS page heading and page footing commands generate only worksheet headings and worksheet footings.

Syntax: How to Enable Overflow Worksheets

Add the ROWOVERFLOW attribute to your WebFOCUS StyleSheet

```
TYPE=REPORT, ROWOVERFLOW={ON|OFF|PBON}, [ROWLIMIT={n|MAX}], ]$
```

where:

ON

Enables overflow worksheets.

OFF

Disables overflow worksheets. OFF is the default value.

PBON

Inserts WebFOCUS page breaks that display the page heading, footing, and column titles at the appropriate places within the worksheet rows. This option does not cause a new worksheet to start when a WebFOCUS page break occurs.

ROWLIMIT=*n*

Sets a target value for the number of rows to be included on a worksheet to *n* rows. The default value is the LINES value (by default, 57).

ROWLIMIT=MAX

Sets a target value for the number of rows to be included on a worksheet to 1,048,000 rows for XLSX output.

Reference: Usage Notes for XLSX Overflow Worksheets

- ❑ The report heading is placed once at the start of the first sheet. The report footing is placed once at the bottom of the last overflow sheet.
- ❑ Unless the PBON setting is used, worksheet headings and column titles are repeated at the top of the original sheet and each subsequent overflow sheet. worksheet footings are placed at the bottom of the original sheet and each subsequent overflow sheet. The data values are displayed on the top data row of each overflow sheet as they would be on a standard new page.
- ❑ Report total lines are displayed at the bottom of the last overflow sheet directly above the final page and table footings.
- ❑ Subheadings, subfootings, and subtotal lines display within the data flow as normal. No special consideration is made to retain groupings within a given sheet.
- ❑ If ROWOVERFLOW=PBON, the page headings and footings and column titles display within the worksheet when a WebFOCUS command causes a page break.
- ❑ For XLSX output, if the ROWOVERFLOW attribute is specified in the StyleSheet and ROWLIMIT is greater than 1MB, the following message is presented and no output file is generated:

(FOC3338) The row limit for EXCEL XLSX worksheets is 1048576.

- ❑ Output types that contain formula references (EXL2K PIVOT and EXL2K FORMULA) are not supported, as formula references are not automatically updated to reflect placement on new overflow worksheets.
- ❑ The overflow worksheet feature applies to rows only, not columns. A new worksheet will not automatically be created if a report generates more than the Excel 2007/2010 limit or 16,384 columns.
- ❑ ROWOVERFLOW is supported for BYTOC reports for XLSX.
- ❑ As named ranges in Excel cannot run across multiple worksheets, the IN-RANGES phrase that defines named ranges in the resulting workbook is not supported with the ROWOVERFLOW feature. When they exist together in the same request, ROWOVERFLOW takes precedence and the IN-RANGES phrase is ignored.

Example: Creating Overflow Worksheets

The following request creates XLSX report output with overflow worksheets. The ROWOVERFLOW=ON attribute in the StyleSheet activates the overflow feature. Without this attribute, one worksheet would have been generated instead of three.

```
TABLE FILE GGSales
-* ****Report Heading****
ON TABLE SUBHEAD
"SALES BY REGION, CATEGORY, AND PRODUCT"
" "
-* ****Worksheet Heading****
HEADING
"SALES REPORT WORKSHEET <TABPAGENO"
" "
-* ****Worksheet Footing****
FOOTING
" "
"END OF WORKSHEET <TABPAGENO"
PRINT DOLLARS UNITS BUDDOLLARS BUDUNITS
BY REGION
BY CATEGORY
BY PRODUCT
BY DATE
```

```

-* ****Subfoot****
ON REGION SUBFOOT
" "
" End of Region <REGION>
" "
-* ****Subhead****
ON REGION SUBFOOT
" "
" End of Region <REGION>
" "
-* ****Report Footing****
ON TABLE SUBFOOT
" "
"END OF REPORT"
ON TABLE PCHOLD FORMAT XLSX
ON TABLE SET STYLE *
TYPE=REPORT, TITLETEXT=EXLOVER, ROWOVERFLOW=ON, ROWLIMIT=2000,$
ENDSTYLE
END

```

The report heading displays on the first worksheet only, the page heading and column titles display on each worksheet, and the subhead and subfoot display whenever the associated sort field changes value. The following image shows the top of the first worksheet, displaying the report heading, page heading, column titles, and first subhead.

SALES BY REGION, CATEGORY, AND PRODUCT										
	A	B	C	D	E	F	G	H	I	J
1	SALES BY REGION, CATEGORY, AND PRODUCT									
2										
3	SALES REPORT WORKSHEET 1									
4										
5	Region	Category	Product	Date	Dollar Sales	Unit Sales	Budget Dollars	Budget Units		
6										
7	Category Coffee for Region Midwest									
8										
9	Midwest	Coffee	Espresso	1996/01/01	19752	1646	24141	1857		
10					13416	1118	16068	1236		
11					13170	878	10000	1000		
12					10164	924	12552	1046		
13					6048	432	6565	505		
14					4693	361	8310	554		
15				1996/02/01	16968	1212	13622	973		
16					13420	1220	20415	1361		
17					13095	873	6910	691		
18					8340	834	8460	705		
19					6454	461	8540	610		
20					4355	335	5400	360		
21				1996/03/01	18466	1319	15120	1512		
22					18135	1395	17069	1313		
23					12690	846	7060	706		
24					11450	1145	14490	1035		
25					10802	982	9555	735		
26					10033	771	9180	765		

Overcoming the Excel 2007/2010 Row Limit Using Overflow Worksheets

Note that the TITLETEXT attribute in the StyleSheet specified the name EXLOVER, so the three worksheets were generated with the names EXLOVER1, EXLOVER2, and EXLOVER3. If there had been no TITLETEXT attribute, the sheets would have been named SHEET1, SHEET2, and SHEET3.

The worksheet footing displays at the bottom of each worksheet and the report footing displays at the bottom of the last worksheet. The following image shows the bottom of the last worksheet, displaying the last subfoot, the page footing, and the report footing.

	A	B	C	D	E	F	G	H	I	J
363					9194	657	7722	772		
364					8981	816	7983	665		
365				1997/08/01	14276	1020	17087	1139		
366					8679	868	7366	737		
367					2179	168	1911	127		
368				1997/09/01	4380	365	5869	451		
369					2152	215	3732	287		
370					1459	112	0	0		
371				1997/10/01	9798	754	8070	673		
372					6402	640	9880	659		
373					6302	450	6570	597		
374				1997/11/01	8448	845	7855	714		
375					6154	440	4966	451		
376					4552	414	4844	484		
377				1997/12/01	15092	1161	16549	1182		
378					11197	746	13464	898		
379					10346	690	10536	810		
380										
381	End of Region West									
382										
383										
384	END OF WORKSHEET 3									
385										
386	END OF REPORT									

Example: Creating Overflow Worksheets With WebFOCUS Page Breaks

The following request creates XLSX report output with overflow worksheets. The ROWOVERFLOW=PON attribute in the StyleSheet activates the overflow feature, and the ROWLIMIT=250 sets the maximum number of rows in each worksheet to approximately 250. Without this attribute, one worksheet would have been generated. The PRODUCT sort phrase specifies a page break.

```
TABLE FILE GGSALES
-* ****Report Heading****
ON TABLE SUBHEAD
"SALES BY REGION, CATEGORY, AND PRODUCT"
" "
PRINT DOLLARS UNITS BUDDOLLARS BUDUNITS
BY REGION
BY HIGHEST CATEGORY
BY PRODUCT PAGE-BREAK
BY DATE
WHERE DATE GE '19971001'
-* ****Page Heading****
HEADING
" Product: <PRODUCT in Category: <CATEGORY for Region: <REGION"
-* ****Page Footing****
FOOTING
" "
-* ****Report Footing****
ON TABLE SUBFOOT
" "
"END OF REPORT"
ON TABLE SET BYDISPLAY ON
ON TABLE PCHOLD FORMAT XLSX
ON TABLE SET STYLE *
INCLUDE=endeflt, TITLETEXT=EXLOVER, ROWOVERFLOW=PON, ROWLIMIT=250,
$
ENDSTYLE
END
```

Overcoming the Excel 2007/2010 Row Limit Using Overflow Worksheets

The report heading displays on the first worksheet only, the page heading, footing, and column titles display on each worksheet and at each WebFOCUS page break (each time the product changes), and the subhead and subfoot display whenever the associated sort field changes value. The following image shows the top of the first worksheet.

SALES BY REGION, CATEGORY, AND PRODUCT

Product: Coffee Grinder in Category: Gifts for Region: Midwest

Region	Category	Product	Date	Dollar Sales	Unit Sales	Budget Dollars	Budget Units
Midwest	Gifts	Coffee Grinder	1997/10/01	14494	95.6	1173.4	1029
Midwest	Gifts	Coffee Grinder	1997/10/01	5452	49.5	501.9	502
Midwest	Gifts	Coffee Grinder	1997/10/01	1750	12.5	217.5	145
Midwest	Gifts	Coffee Grinder	1997/11/01	12525	114.0	1009.7	1191
Midwest	Gifts	Coffee Grinder	1997/11/01	12190	110.5	1457.1	1041
Midwest	Gifts	Coffee Grinder	1997/11/01	9552	39.5	1016.1	547
Midwest	Gifts	Coffee Grinder	1997/12/01	47455	550.9	4503.5	3757
Midwest	Gifts	Coffee Grinder	1997/12/01	7355	49.0	550.1	530
Midwest	Gifts	Coffee Grinder	1997/12/01	4075	37.0	513.5	513
Midwest	Gifts	Coffee Grinder	1997/12/01	1957	17.9	354.0	295

Product: Coffee Pot in Category: Gifts for Region: Midwest

Region	Category	Product	Date	Dollar Sales	Unit Sales	Budget Dollars	Budget Units
Midwest	Gifts	Coffee Pot	1997/10/01	12525	107.0	1533.5	1032
Midwest	Gifts	Coffee Pot	1997/10/01	11509	115.1	1542.5	1215
Midwest	Gifts	Coffee Pot	1997/10/01	1375	11.5	112.4	55
Midwest	Gifts	Coffee Pot	1997/11/01	5345	59.5	1099.1	545
Midwest	Gifts	Coffee Pot	1997/11/01	5151	51.5	910.1	530
Midwest	Gifts	Coffee Pot	1997/11/01	3722	34.5	235.4	199
Midwest	Gifts	Coffee Pot	1997/12/01	10592	70.5	530.5	534
Midwest	Gifts	Coffee Pot	1997/12/01	9257	51.7	955.0	544
Midwest	Gifts	Coffee Pot	1997/12/01	5500	44.5	422.1	352

Product: Mug in Category: Gifts for Region: Midwest

Region	Category	Product	Date	Dollar Sales	Unit Sales	Budget Dollars	Budget Units
Midwest	Gifts	Mug	1997/10/01	14492	111.5	1555.2	1044
Midwest	Gifts	Mug	1997/10/01	14325	95.5	1254.2	542
Midwest	Gifts	Mug	1997/10/01	11275	112.7	1175.1	1175
Midwest	Gifts	Mug	1997/10/01	10745	75.5	1102.9	725
Midwest	Gifts	Mug	1997/10/01	7319	73.2	954.1	702
Midwest	Gifts	Mug	1997/10/01	4955	33.2	352.5	395
Midwest	Gifts	Mug	1997/11/01	11450	33.5	972.5	749
Midwest	Gifts	Mug	1997/11/01	9295	72.2	777.5	707
Midwest	Gifts	Mug	1997/11/01	5551	33.5	1107.5	729
Midwest	Gifts	Mug	1997/11/01	4917	49.2	592.5	495
Midwest	Gifts	Mug	1997/11/01	1950	13.1	322.1	255
Midwest	Gifts	Mug	1997/11/01	1955	13.0	390.1	250
Midwest	Gifts	Mug	1997/12/01	15255	109.0	1555.2	1029
Midwest	Gifts	Mug	1997/12/01	14912	114.7	1452.0	1215
Midwest	Gifts	Mug	1997/12/01	14555	112.7	1790.9	1194
Midwest	Gifts	Mug	1997/12/01	14327	110.2	1555.5	1125

As of WebFOCUS Reporting Server Release 7.7.04, distribution of burst workbooks that have been generated using ROWOVERFLOW is supported in XLSX.

Excel Compound Reports

As of WebFOCUS Reporting Server Release 7.7.05, the Excel compound reports feature is supported. Excel compound reports generate compound workbooks that can contain multiple worksheet reports using the XLSX output format. The syntax of Excel compound reports is identical to that of PDF compound reports. By default, each of the component reports from the compound report is placed in a new Excel worksheet (analogous to a new page in PDF).

The components of an Excel compound report can include standard tables, Table of Content (BYTOC) reports, and ROWOVERFLOW.

Reference: Usage Notes for Excel Compound Reports

- ❑ Coordinated compound reports that generate individual instances of the overall report for each unique primary key are not available in XLSX.
- ❑ Graphs and images are not supported.

Note: Since multiple tables are generated, WebFOCUS will ensure that each tab name is unique.

Example: Compound Layout with Overflow Worksheets in XLSX

```
SET PAGE-NUM=OFF
COMPOUND LAYOUT PCHOLD FORMAT XLSX
SECTION=Example, LAYOUT=ON, MERGE=OFF,$
PAGELAYOUT=1,$
COMPONENT=R1, TYPE=REPORT,$
COMPONENT=R2, TYPE=REPORT,$
COMPONENT=R3, TYPE=REPORT,$
COMPONENT=R4, TYPE=REPORT,$
END
SET COMPONENT=R1
TABLE FILE GGSales
HEADING CENTER
"Gotham Grinds Sales to Information Builders"
" "
"Report 1"
"Sales Summary by Category"
" "
SUM UNITS/D12C BUDUNITS/D12C DOLLARS/D12CM BUDDOLLARS/D12CM
BY CATEGORY
ON TABLE HOLD FORMAT XLSX
ON TABLE SET STYLE *
TYPE=REPORT, TITLETEXT=Sales Summary,$
TYPE=REPORT, TOPMARGIN=1.5,BOTTOMMARGIN=1,PAGESIZE=LETTER,$
TYPE=TITLE, COLOR=WHITE, BACKCOLOR=GREY,$
TYPE=HEADING,LINE=1,COLOR=PURPLE,JUSTIFY=CENTER, STYLE=BOLD,$
TYPE=HEADING,LINE=3,OBJECT=TEXT,COLOR=BLUE, JUSTIFY=CENTER, STYLE=BOLD,$
TYPE=HEADING,LINE=4,OBJECT=TEXT,COLOR=PURPLE, JUSTIFY=CENTER, STYLE=BOLD,$
ENDSTYLE
END
```

```
SET COMPONENT=R2
TABLE FILE GGSALES
HEADING CENTER
"Gotham Grinds Sales to Information Builders"
" "
"Report 2"
"Sales Detail Report By Category"
" "
SUM UNITS/D12C BUDUNITS/D12C DOLLARS/D12CM BUDDOLLARS/D12CM
BY CATEGORY BY PRODUCT BY REGION
ON TABLE SET BYDISPLAY ON
ON TABLE HOLD FORMAT XLSX
ON TABLE SET STYLE *
TYPE=TITLE, COLOR=WHITE, BACKCOLOR=GREY,$
TYPE=HEADING,LINE=1,OBJECT=TEXT,COLOR=PURPLE, JUSTIFY=CENTER, STYLE=BOLD,$
TYPE=HEADING,LINE=3,OBJECT=TEXT,COLOR=BLUE, JUSTIFY=CENTER, STYLE=BOLD,$
TYPE=HEADING,LINE=4,OBJECT=TEXT,COLOR=PURPLE, JUSTIFY=CENTER, STYLE=BOLD,$
TYPE=REPORT, TITLETEXT=Sales Detail,$
TYPE=REPORT, TOPMARGIN=1.5, BOTTOMMARGIN=1, PAGESIZE=LETTER,$
ENDSTYLE
END

SET COMPONENT=R3
TABLE FILE GGSALES
HEADING CENTER
"Gotham Grinds Sales to Information Builders"
" "
"Report 3"
"Sales Summary by Region"
" "
SUM UNITS/D12C BUDUNITS/D12C DOLLARS/D12CM BUDDOLLARS/D12CM
BY REGION
ON TABLE HOLD FORMAT XLSX
ON TABLE SET STYLE *
TYPE=REPORT, TITLETEXT=Region Summary,$
TYPE=REPORT, TOPMARGIN=1.5, BOTTOMMARGIN=1, PAGESIZE=LETTER,$
TYPE=TITLE, COLOR=WHITE, BACKCOLOR=GREY,$
TYPE=HEADING, LINE=1, OBJECT=TEXT, COLOR=PURPLE, JUSTIFY=CENTER,
STYLE=BOLD,$
TYPE=HEADING, LINE=3, OBJECT=TEXT, COLOR=BLUE, JUSTIFY=CENTER, STYLE=BOLD,$
TYPE=HEADING, LINE=4, OBJECT=TEXT, COLOR=PURPLE, JUSTIFY=CENTER,
STYLE=BOLD,$
ENDSTYLE
END
```

Excel Compound Reports

```

SET COMPONENT=R4
TABLE FILE GGSALES
SUM UNITS/D12C DOLLARS/D12CM
BY REGION BY CATEGORY BY PRODUCT
HEADING CENTER
"Gotham Grinds Sales to Information Builders"
" "
"Report 4"
"Sales Detail By Region"
ON REGION SUBHEAD
"<REGION Region Sales"
ON TABLE PCHOLD FORMAT XLSX
ON TABLE SET BYDISPLAY ON
ON TABLE SET COMPOUND BY
ON TABLE SET STYLE *
TYPE=REPORT, TOPMARGIN=1.5,BOTTOMMARGIN=1,PAGESIZE=LETTER,$
TYPE=REPORT, TITLETEXT=Region Details,$
TYPE=TITLE, COLOR=WHITE, BACKCOLOR=GREY,$
TYPE=HEADING, LINE=1, COLOR=PURPLE, JUSTIFY=CENTER,STYLE=BOLD,$
TYPE=HEADING, LINE=3, COLOR=BLUE, JUSTIFY=CENTER, STYLE=BOLD,$
TYPE=HEADING, LINE=4, COLOR=PURPLE, JUSTIFY=CENTER, STYLE=BOLD,$
ENDSTYLE
END

```

The output is:

Report 1: Simple Summary Report

	A	B	C	D	E
1	Gotham Grinds Sales to Information Builders				
2					
3	Report 1				
4	Sales Summary by Category				
5					
6	Category	Unit Sales	Budget Units	Dollar Sales	Budget Dollars
7	Coffee	1,376,266	1,385,923	\$17,231,455	\$17,293,886
8	Food	1,384,845	1,377,564	\$17,229,333	\$17,267,160
9	Gifts	927,880	931,007	\$11,695,502	\$11,659,732
10					
11					

Ready | Sales Summary | Sales Detail | Region Summary | Region Details

Report 2: Detail Report

	A	B	C	D	E	F	G
1	Gotham Grinds Sales to Information Builders						
2							
3	Report 2						
4	Sales Detail Report By Category						
5							
6	Category	Product	Region	Unit Sales	Budget Units	Dollar Sales	Budget Dollars
7	Coffee	Capuccino	Northeast	44,785	44,432	\$542,095	\$561,491
8	Coffee	Capuccino	Southeast	73,264	75,353	\$944,000	\$956,661
9	Coffee	Capuccino	West	71,168	70,585	\$895,495	\$877,304
10	Coffee	Espresso	Midwest	101,154	101,869	\$1,294,947	\$1,258,232
11	Coffee	Espresso	Northeast	68,127	69,776	\$850,107	\$872,902
12	Coffee	Espresso	Southeast	68,030	66,785	\$853,572	\$849,465
13	Coffee	Espresso	West	71,675	72,927	\$907,617	\$923,941
14	Coffee	Latte	Midwest	231,623	233,657	\$2,883,566	\$2,827,800
15	Coffee	Latte	Northeast	222,866	221,712	\$2,771,815	\$2,818,069
16	Coffee	Latte	Southeast	209,654	213,555	\$2,617,836	\$2,625,303
17	Coffee	Latte	West	213,920	215,272	\$2,670,405	\$2,722,718
18	Food	Biscotti	Midwest	86,105	85,839	\$1,091,727	\$1,067,629
19	Food	Biscotti	Northeast	145,242	145,152	\$1,802,005	\$1,848,682
20	Food	Biscotti	Southeast	119,594	120,549	\$1,505,717	\$1,512,019
21	Food	Biscotti	West	70,436	67,780	\$863,868	\$861,804

Report 3: Summary Report

	A	B	C	D	E
1	Gotham Grinds Sales to Information Builders				
2					
3	Report 3				
4	Sales Summary by Region				
5					
6	Region	Unit Sales	Budget Units	Dollar Sales	Budget Dollars
7	Midwest	905,045	907,107	\$11,400,665	\$11,194,373
8	Northeast	916,675	914,359	\$11,392,300	\$11,576,921
9	Southeast	935,232	942,247	\$11,710,379	\$11,807,971
10	West	932,039	930,781	\$11,652,946	\$11,641,513
11					
12					

Report 4: Overflow Report

	A	B	C	D	E	F
1	Gotham Grinds Sales to Information Builders					
2						
3	Report 4					
4	Sales Detail By Region					
5	Region	Category	Product	Unit Sales	Dollar Sales	
6	Midwest Region Sales					
7	Midwest	Coffee	Espresso	101,154	\$1,294,947	
8	Midwest	Coffee	Latte	231,623	\$2,883,566	
9	Midwest	Food	Biscotti	86,105	\$1,091,727	
10	Midwest	Food	Croissant	139,182	\$1,751,124	
11	Midwest	Food	Scone	116,127	\$1,495,420	
12	Midwest	Gifts	Coffee Grinder	50,393	\$619,154	
13	Midwest	Gifts	Coffee Pot	47,156	\$599,878	
14	Midwest	Gifts	Mug	86,718	\$1,086,943	
15	Midwest	Gifts	Thermos	46,587	\$577,906	
16	Northeast Region Sales					
17	Northeast	Coffee	Capuccino	44,785	\$542,095	
18	Northeast	Coffee	Espresso	68,127	\$850,107	
19	Northeast	Coffee	Latte	222,866	\$2,771,815	
20	Northeast	Food	Biscotti	145,242	\$1,802,005	
21	Northeast	Food	Croissant	137,394	\$1,670,818	
22	Northeast	Food	Scone	70,732	\$907,171	
23	Northeast	Gifts	Coffee Grinder	40,977	\$509,200	
24	Northeast	Gifts	Coffee Pot	46,185	\$590,780	
25	Northeast	Gifts	Mug	91,497	\$1,144,211	
26	Northeast	Gifts	Thermos	48,870	\$604,098	

Reference: Guidelines for Using the Legacy OPEN, CLOSE, and NOBREAK Keywords and SET COMPOUND

The keywords OPEN, CLOSE, and NOBREAK are used to control Excel compound reports. They can be specified with the HOLD or PCHOLD command or with a separate SET COMPOUND command.

- ❑ OPEN is used on the first report of a sequence of component reports to specify that a compound report should be started.
- ❑ CLOSE is used to designate the last report in a compound report.
- ❑ NOBREAK specifies that the next report be placed on the same worksheet as the current report. If it is not present, the default behavior is to place the next report on a separate worksheet.
- ❑ When used with the HOLD or PCHOLD syntax, the compound report keywords OPEN, CLOSE, and NOBREAK must appear immediately after FORMAT XLSX. For example, you can specify:
 - ❑ ON TABLE PCHOLD FORMAT XLSX OPEN
 - ❑ ON TABLE HOLD AS MYHOLD FORMAT XLSX OPEN NOBREAK

- ❑ As with PDF compound reports, compound report keywords can be alternatively specified using SET COMPOUND:
 - ❑ `SET COMPOUND = OPEN`
 - ❑ `SET COMPOUND = 'OPEN NOBREAK'`
 - ❑ `SET COMPOUND = NOBREAK`
 - ❑ `SET COMPOUND = CLOSE`

Guidelines for Producing Excel Compound Reports

- ❑ **Naming of Worksheets.** The default worksheet tab names will be Sheet1, Sheet2, and so on. You have the option to specify a different worksheet tab name by using the TITLETEXT keyword in the StyleSheet. For example:

```
TYPE=REPORT, TITLETEXT='Summary Report ', $
```

Excel limits the length of worksheet titles to 31 characters. The following special characters cannot be used: ':', '?', '*', and '/'.

- ❑ **File Names and Formats.** The output file name (AS name, or HOLD by default) is obtained from the first report of the compound report (the report with the OPEN keyword). Output file names on subsequent reports are ignored.

The HOLD FORMAT syntax used in the first component report in a compound report applies to all subsequent reports in the compound report, regardless of their format.

- ❑ **NOBREAK Behavior.** When NOBREAK is specified, the following report appears on the row immediately after the last row of the report with the NOBREAK. If additional spacing is required between the reports, a FOOTING or an ON TABLE SUBFOOT can be placed on the report with the NOBREAK, or a HEADING or an ON TABLE SUBHEAD can be placed on the following report. This allows the most flexibility, since if blank rows were added by default, there would be no way to remove them.

Example: Creating a Simple Compound Report Using XLSX

```
SET PAGE-NUM=OFF
TABLE FILE GGSALES
HEADING
"Report 1: Coffee - Budget"
" "
SUM BUDDOLLARS BUDUNITS COLUMN-TOTAL AS 'Total'
BY REGION
ON TABLE SET STYLE *
TYPE=REPORT, TITLETEXT=Coffee Budget,$
TYPE=HEADING, SIZE=14,$
ENDSTYLE
ON TABLE PCHOLD AS EX1 FORMAT XLSX OPEN
END

TABLE FILE GGSALES
HEADING
"Report 2: Coffee - Actual "
SUM DOLLARS UNITS COLUMN-TOTAL AS 'Total'
BY REGION
ON TABLE PCHOLD FORMAT XLSX
ON TABLE SET STYLE *
TYPE=REPORT, TITLETEXT=Coffee Actual,$
TYPE=HEADING, SIZE=14,$
ENDSTYLE
END

TABLE FILE GGSALES
HEADING
"Report 3: Food - Budget"
SUM BUDDOLLARS BUDUNITS COLUMN-TOTAL AS 'Total'
BY REGION
ON TABLE SET STYLE *
TYPE=REPORT, TITLETEXT=Food Budget,$
TYPE=HEADING, SIZE=14,$
ENDSTYLE
ON TABLE PCHOLD FORMAT XLSX CLOSE
END
```


The output is:

Report 1: Coffee - Budget			Report 2: Coffee - Actual			Report 3: Food - Budget		
Region	Budget Dollars	Budget Units	Region	Dollar Sales	Unit Sales	Region	Budget Dollars	Budget Units
Midwest	11194373	907107	Midwest	11400665	905045	Midwest	11194373	907107
Northeast	11576921	914359	Northeast	11392300	916675	Northeast	11576921	914359
Southeast	11807971	942247	Southeast	11710379	935232	Southeast	11807971	942247
West	11641513	930781	West	11652946	932039	West	11641513	930781
Total	46220778	3694494	Total	46156290	3688991	Total	46220778	3694494

Example: Creating a Compound Report Using NOBREAK

In this example, the first two reports are on the first worksheet, and the last two reports are on the second worksheet, since NOBREAK appears on both the first and third reports.

```
TABLE FILE GGSales
HEADING
"Report 1: Coffee - Budget"
SUM BUDDOLLARS BUDUNITS COLUMN-TOTAL AS 'Total'
BY REGION
IF CATEGORY EQ Coffee
ON TABLE PCHOLD FORMAT XLSX OPEN NOBREAK
ON TABLE SET STYLE *
TYPE=REPORT, TITLETEXT=Coffee, FONT=ARIAL, SIZE=10, STYLE=NORMAL,$
TYPE=TITLE, STYLE=BOLD,$
TYPE=HEADING, SIZE=12, STYLE=BOLD, COLOR=BLUE,$
TYPE=GRANDTOTAL, STYLE=BOLD,$
END

TABLE FILE GGSales
HEADING
" "
"Report 2: Coffee - Actual "
SUM DOLLARS UNITS COLUMN-TOTAL AS 'Total'
BY REGION
IF CATEGORY EQ Coffee
ON TABLE PCHOLD FORMAT XLSX
ON TABLE SET STYLE *
TYPE=REPORT, FONT=ARIAL, SIZE=10, STYLE=NORMAL,$
TYPE=GRANDTOTAL, STYLE=BOLD,$
TYPE=HEADING, SIZE=12, STYLE=BOLD, COLOR=BLUE,$
END
```

Guidelines for Producing Excel Compound Reports

```
TABLE FILE GGSALES
HEADING
"Report 3: Food - Budget"
SUM BUDDOLLARS BUDUNITS COLUMN-TOTAL AS 'Total'
BY REGION
IF CATEGORY EQ Food
ON TABLE PCHOLD FORMAT XLSX NOBREAK
ON TABLE SET STYLE *
TYPE=REPORT, TITLETEXT=Food, FONT=ARIAL, SIZE=10, STYLE=NORMAL,$
TYPE=HEADING, STYLE=BOLD, SIZE=12, COLOR=BLUE,$
TYPE=TITLE, STYLE=BOLD,$
TYPE=GRANDTOTAL, STYLE=BOLD,$
END

TABLE FILE GGSALES
HEADING
" "
"Report 4: Food - Actual"
SUM DOLLARS UNITS COLUMN-TOTAL AS 'Total'
BY REGION
IF CATEGORY EQ Food
ON TABLE PCHOLD FORMAT XLSX CLOSE
ON TABLE SET STYLE *
TYPE=REPORT, FONT=ARIAL, SIZE=10, $
TYPE=TITLE, STYLE=BOLD,$
TYPE=HEADING, SIZE=12, STYLE=BOLD, COLOR=BLUE,$
TYPE=GRANDTOTAL, STYLE=BOLD,$
END
```

Report output is displayed in two separate tabs.

The image shows two side-by-side Excel spreadsheets, both titled 'WF5erviet [Read-Only]'. The left spreadsheet displays 'Report 1: Coffee - Budget' and 'Report 2: Coffee - Actual'. The right spreadsheet displays 'Report 3: Food - Budget' and 'Report 4: Food - Actual'. Each report is structured with a header row and a data table with columns for Region, Budget Dollars, Budget Units (for budget reports) or Dollar Sales, Unit Sales (for actual reports). The data is summarized by region and a total row.

Report 1: Coffee - Budget			
Region	Budget Dollars	Budget Units	
Midwest	4086032	335526	
Northeast	4252462	335920	
Southeast	4431429	355693	
West	4523963	358784	
Total	17293886	1385923	

Report 2: Coffee - Actual			
Region	Dollar Sales	Unit Sales	
Midwest	4178513	332777	
Northeast	4164017	335778	
Southeast	4415408	350948	
West	4473517	356763	
Total	17231455	1376266	

Report 3: Food - Budget			
Region	Budget Dollars	Budget Units	
Midwest	4220721	339263	
Northeast	4453907	351431	
Southeast	4409288	351509	
West	4183244	335361	
Total	17267160	1377564	

Report 4: Food - Actual			
Region	Dollar Sales	Unit Sales	
Midwest	4338271	341414	
Northeast	4379994	353368	
Southeast	4308731	349829	
West	4202337	340234	
Total	17229333	1384845	

FORMAT XLSX Limitations

Format XLSX in WebFOCUS Release 7.7.02 and higher does not support the following features, currently supported for EXL2K:

- Cell locking
- Pivot tables
- Tooltips for drill-down hyperlinks
- XLSX reports are available on a z/OS USS server but are not currently supported on a z/OS PDS server or on a z/OS USS server with the setting DYNAM TEMP ALLOC MVS.

For additional support on the implementation of features supported by the XLSX format, see *WebFOCUS XLSX Format Supported Features Roadmap*, located at the following link:

https://techsupport.informationbuilders.com/tech/wbf/wbf_rln_formatXLSX_support.html

