Using BI Publisher Excel Templates with PeopleTools

Randy Groncki
9/21/2020

Summary

Starting with the release of PeopleTools 8.58, PeopleTools can now use Excel Spreadsheets as BI Publisher templates.   Before this release, if we wanted to create a spreadsheet using BI Publisher, the fastest way was to use an RTF template with a table and force the output to Excel through the report definition.   However, this is more of a "Work Around" than a solution and left a few concepts to be desired.

Now using the Excel templates, we can work directly in Excel to define the report which provides many of the features and tools that the users are requesting.

This document will introduce the basics of using Excel Templates with BI Publisher.

What and Why Excel Templates

Excel templates allow us to generate Excel spreadsheets in BI Publisher using Excel.

Excel templates are not new to BI Publisher.   They are just new to PeopleTools' implementation of BI Publisher.   Luckily, there is plentiful documentation, demonstrations, and examples of Excel Templates available.   We just need leverage these within our PeopleTools environment.

Just like we used MS Word to create RTF templates, we use Excel to create XLS templates with the same desktop helper (Template Builder).

Now we can create multiple page spreadsheets natively formatted in Excel and delivered through the PeopleTools framework where and when they are needed.

This is a link to the Oracle Excel Template Documentation:
   docs.oracle.com/html/E22254_03/create_excel_tmpl.htm

# BI Publisher Excel Template Basics

Installing the Template Builder

Download the latest version of BI Publisher Desktop Template Builder as per your version of PeopleTools.   The easiest way is through your PeopleSoft Application

> Home > Reporting Tools > BI Publisher > Setup > Design Helper

Make sure you install the correct 32/64-bit version as per your version of MS Office. Installing the wrong version will create problems.   If this does happen, completely uninstall before re-installing the correct version.

 If you installed the Desktop helper for MS Word, then you have already installed the Desktop Helper for Excel.   Just look for the BI Publisher heading on your ribbon in Excel just an in Word.

Word
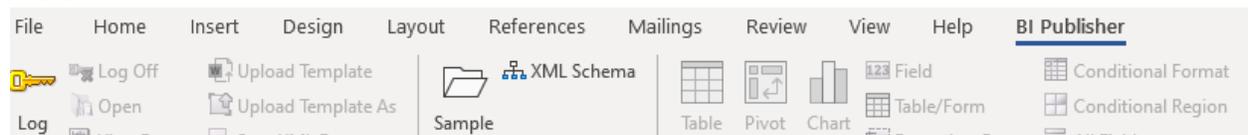
Excel

Notes on the Excel Template Builder

XLS, Not XLSX

> BI Publisher uses the .xls spreadsheets (1997-2003 Workbook), not the modern .xlsx spreadsheets.   When coding spreadsheets, you can only use capabilities and functions available to this version.   Advanced formatting and functions from the current version will not work.

Hidden XDO_METADATA Sheet

> The Template Builder will create a hidden sheet to the Excel workbook when the first field is added.   BI Publisher uses this hidden sheet for mapping between the data and the spreadsheet. You might have to edit some of these references and formulas.
>
> To unhide the sheet
> - Right click on the
> - Select UnHide
> - Select XDO_MEDATA
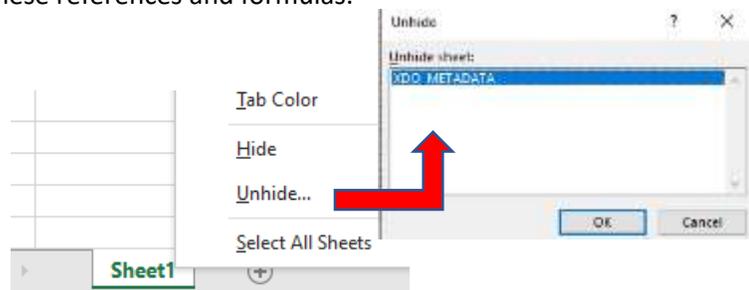
The top section (rows 1-8) are basic identification information needed for the template.

	A	B
1	Version	
2	ARU-dbdrv	
3	Extractor Version	
4	Template Code	
5	Template Type	TYPE_EXCEL_TEMPLATE
6	Preprocess XSLT File	
7	Last Modified Date	
8	Last Modified By	
9		
10	Data Constraints:	
11	XDO_?XDOFIELD1?	<?fld_STATE_DESCR?>

Our work is done in the "Data Constraints" section starting with row 10.

Best practice is to re-hide the XDO_METADATA sheet after development of the report.

Not all Excel Functions, formatting and formulas will work in the template

The template and report are Excel 2003 objects (.xls). Formatting and features are limited to that Excel version's capabilities. An example of this Excel 2003 allows only three conditional formatting rules per cell where the current version allows more.

Additionally, some features such as data filtering will not flow through to the final report.

If incorporating images in a multi-sheet report, the images will only appear on the first sheet. Not subsequent sheets.

The Template Builder for Excel provides basic design capability. More advanced designs will require manual XSL coding.

Creating a basic Template

For the purposes of all our examples, we are creating a spreadsheet listing all taxes for a specific payroll. This spreadsheet will be a basic, one-page spreadsheet listing all the taxes with a total at the bottom.

Create a sample XML file

Create a sample XML File using PeopleTools through PSQuery or another method such as RowSets or XMLDoc objects. Creating the XML file is beyond the scope of this document.

The sample XML file must contain the exact same data structure and tag names as the file produced by the production system for this report.

Generally, I choose to create all the PeopleTools objects needed to support the report (components, pages, PeopleCode, PSQueries, etc.) before creating the report and have those objects generate the example XML file. This guarantees the sample file is an exact representation of the report data.

Sort and enrich the data as needed before creating the XML file. The data manipulation tools in the Excel template are limited.

The Excel Spreadsheet

Open a new Excel spreadsheet and save it as an .xls file (Excel 97-2003 Workbook).

This will avoid later issues using formatting or functions not available to this version of Excel.

SampleBIPubExcelTemplate

Excel 97-2003 Workbook (*.xls)

More options...
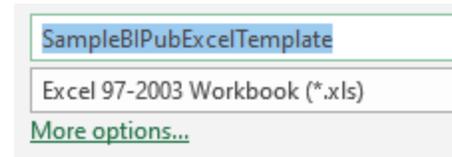
On the BI Publisher section of the ribbon, upload your sample XML file using the "Sample XML" icon in the Load Data group.
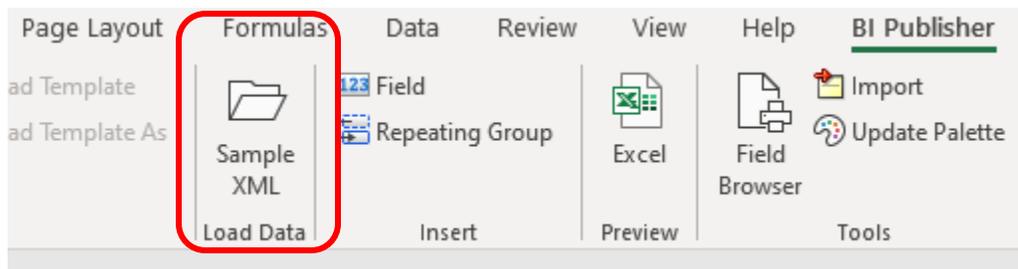

Add fields and titles to the spreadsheet

Use the "Field" tool to add fields from the XML File to your spreadsheet.
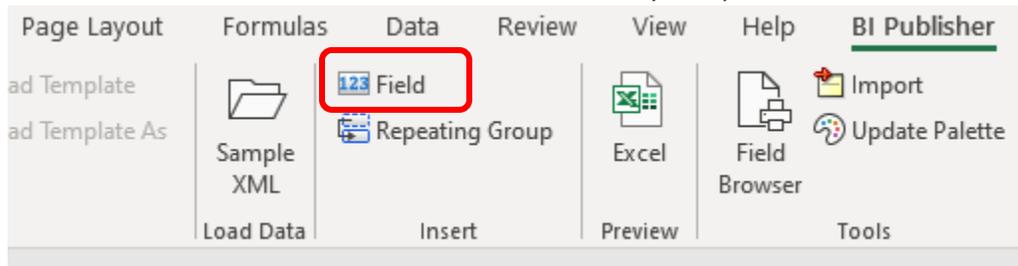
This will open a dialog box showing the data structure and fields of the sample XML File.

The first time clicking this control, you will receive a popup warning from Excel:" Meta data sheet will be created"

We will cover the meta data sheet later in this paper.

Meta data sheet will be created.

OK   Cancel
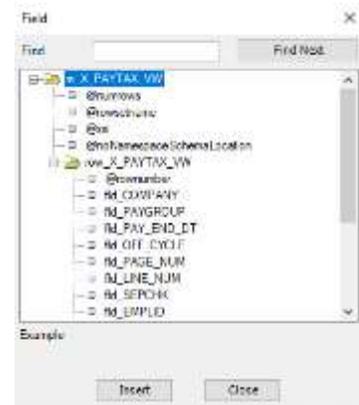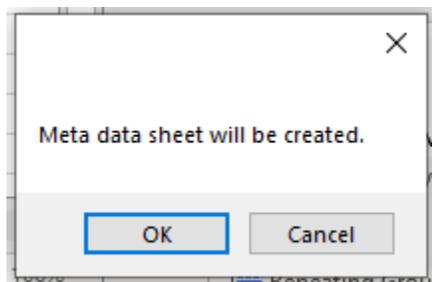
Add fields and titles to your spreadsheet.

	A	B	C	D	E	F	G
1	State	Locality	Tax Class	EE/ER	Current Tax	Emplid	Name
2	Federal	LEXINGTON-FAYETTE	OASDI/Disability - EE	E	162.21	KU0087	Lopez,Wilma
3							

Template Builder will show data from your example XML file in the cells rather than the XML Tag names for those fields.   It will not add titles to the columns for those fields.

Add titles and Do your basic field/column formatting such column width, alignment, and number formatting.

Add repeating groups

Select the cells in the row that will contain the repeating data (i.e. data rows of your XML file). Then click on the "Repeating Group" tool on the ribbon to bring up the Properties box.

Choose the Row of the repeating data from the XML data structure options:

Test the template
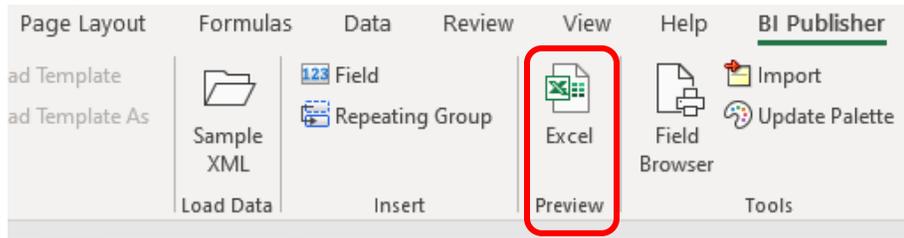
Click on the Excel tool on the ribbon. A new spreadsheet should generate with the data from the sample XML file neatly formatted in columns.

	A	B	C	D	E	F	
1	State	Locality	Tax Class	EE/ER	Current Tax	Emplid	Name
2	Federal		OASDI/Disability - EE	E	162.21	KU0087	Lopez,Wilma
3	Federal		OASDI/Disability - ER	R	162.21	KU0087	Lopez,Wilma
4	Federal		FICA Med Hospital Ins / E	E	37.94	KU0087	Lopez,Wilma
5	Federal		Withholding	E	351.69	KU0087	Lopez,Wilma
6	Federal		FICA Med Hospital Ins / E	R	37.94	KU0087	Lopez,Wilma
7	Federal		Unemployment ER	R	0.00	KU0087	Lopez,Wilma
8	Utah		FUTA Credit Reduction	E	0.00	KU0087	Lopez,Wilma
9	Utah		Withholding	E	126.44	KU0087	Lopez,Wilma
10	Utah		Unemployment ER	R	176.60	KU0087	Lopez,Wilma
11	California		FUTA Credit Reduction	E	14.31	KU0058	Votava,Teresa
12	California		OASDI/Disability - EE	E	11.92	KU0058	Votava,Teresa
13	California		Withholding	E	18.10	KU0058	Votava,Teresa

Add analytic / aggregate functions

For a simple, single page spreadsheet without data groupings, we can sum the Tax column by just putting a SUM() function in a cell.

Notice the range for the SUM function contains only the one cell in the repeating row. Excel translates the cell address to "`XDO_?XDOFIELD5?`". This address is Named Space Template Builder uses to map data to the spreadsheet. "`=SUM(XDO_?XDOFIELD5?)`" will sum all the values of this field when the spreadsheet is created.

Names Spaces are viewed and controlled by the Names Manager on the "Formulas" Ribbon.

Click on the Preview icon to see if your formula worked correctly.

Adding a sum using the XDO_METADATA Sheet

A spreadsheet summing function can also be added on the XDO_METADATA spreadsheet using XSL commands

In cell below the repeating row, add the field to be summed

Notice the formula field name "XDO_?XDOFIELD8?" in the Name Box.

Switch to the "XDO_METADATA" sheet (unhide it if necessary) and find the field mapping for that named cell.

Change the field reference to a sum. For example, in this case it will be `<?sum(fld_TAX_CUR)?>`

| XDO_GROUP_?XDOG1? | `<xsl:for-each select=".//row_X_PAYTAX` |
| XDO_?XDOFIELD8? | `<?sum(fld_TAX_CUR)?>` |

Test the template again using the Excel control and see that the sum field is working correctly.

	A	B	C	D	E	F	G
1	State	Locality	Tax Class	EE/ER	Current Tax	Emplid	Name
55	New York		Family Leave Insurance -	E	14.43	KU0005	Aliverdi,Reza K
56	New York		Unemployment ER	R	0.00	KU0005	Aliverdi,Reza K
57	New York	NEW YORK	Withholding	E	192.07	KU0005	Aliverdi,Reza K
58	New York	MTA PAYROLL TAX (MCTM1	Local ER	R	18.03	KU0005	Aliverdi,Reza K
59							
60			Total		11,526.64		

Re-hide the XDO_METADATA sheet before publishing.

At this point the spreadsheet is ready for loading into PeopleSoft as a BI Publisher Template.

Create a Grouped Data Template

Notes on Grouping in Excel Templates

The data manipulation tools within the BI Publisher Excel template are not robust nor extensive. The basic functionality will take a flat XML file and group the data. However, the ability for meaningful analysis with Excel features such as aggregate functions is limited.

For this example, we are using the same payroll tax data, but this time we want to group and sum the report by taxing state.

To create grouped data with aggregate functions, the XML data must be in a hierarchical structure within those State groups. Using the same data, we created a new XML file with a parent-child relationship.

X_PAYTAX_ST_VW
⮡ X_PAYTAX_ST_DVW

X_PAYTAX_ST_VW contains a row for each unique state
X_PAYTAX_ST_DVW contains all tax detail information for that state

Grouping Setup

Create a new spreadsheet and load your XML sample file into the BI Publisher Template Builder.

Add the state field to the top.
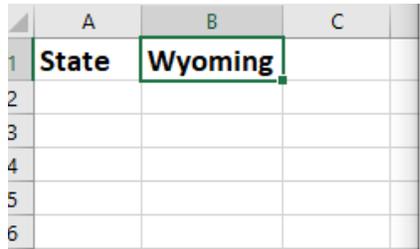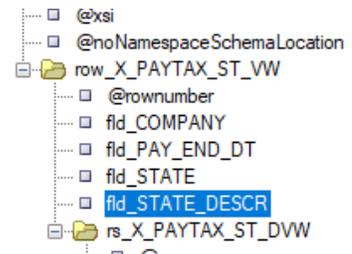
Highlight a block of cells high enough for the structure of each group.

Choose the "Repeating Group" control from the BI Publisher Ribbon Menu and choose the Parent Row Structure in the "For Each" box.   Since the data is already grouped in the data structure, we do not click the "On Grouping" checkbox or complete the "Group By" option.
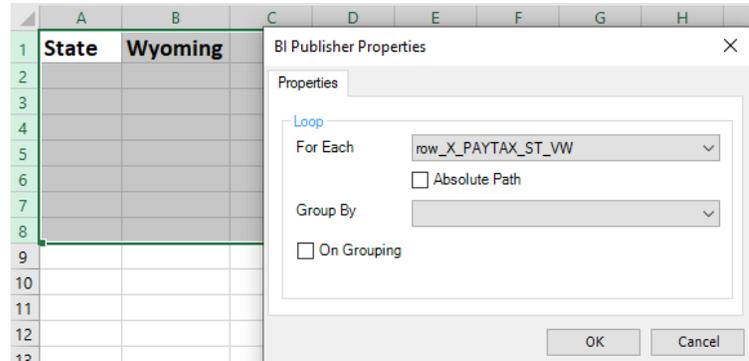
Test out this much by clicking on the Excel icon before continuing.

Group Detail Setup

Add in the fields from the detail/child row of the XML data structure. Add column headings and formatting for numbers and dates.

Create a repeating row for the detail/child row. Select just the data fields on the spreadsheet from the child row and then click on the "Repeating Group" control. Ensure that the Child Row data structure is selected.

State	Wyoming					
	Tax	Locality	EE/ER	Current Tax	Emplid	Name
	FUTA Credit Reduction	WORKERS COMP	E	0.00	KU0061	Wynne,Elbert

BI Publisher Properties   ✕

Properties

Loop

For Each   row_X_PAYTAX_ST_DVW   ⌄

☐ Absolute Path

Group By   ⌄

☐ On Grouping

Test out the detail report by clicking on the Excel icon before continuing.

	A	B	C	D	E	F	G
1	State	Wyoming					
2		Tax	Locality	EE/ER	Current Tax	Emplid	Name
3		FUTA Credit Reduction		E	0.00	KU0061	Wynne,Elbert
4		Unemployment ER		R	190.65	KU0061	Wynne,Elbert
5							
6							
7							
8							
9							
10	State	Utah					
11		Tax	Locality	EE/ER	Current Tax	Emplid	Name
12		FUTA Credit Reduction		E	0.00	KU0087	Lopez,Wilma
13		Unemployment ER		R	176.60	KU0087	Lopez,Wilma
14		Withholding		E	126.44	KU0087	Lopez,Wilma
15							

PT3

Group Totals

We need to add the total field to sum within the group instead of sum for then entire data. This is done using the XDO_METADATA sheet.

Add the field to be summed below your detail data using the field control. Note the Name Box in the left-hand corner for the Template Mapping name (XDO_?XDOFIELD8?).

Select the XDO_METADATA sheet (unhide if necessary) to edit the field mapping of XDO_?XDOFIELD8?.

Change the mapping field to <?sum(.//fld_TAX_CUR)?>. Notice the "//" in front of the fieldname. This forces the template to sum only within that group instead for the entire data set.

	A	B	C
13	XDO_?XDOFIELD2?	<?fld_PY_DESCR30?>	
14	XDO_?XDOFIELD3?	<?fld_LOCALITY_NAME?>	
15	XDO_?XDOFIELD4?	<?fld_EE_ER_TAX_TYPE?>	
16	XDO_?XDOFIELD5?	<?fld_TAX_CUR?>	
17	XDO_?XDOFIELD6?	<?fld_EMPLID?>	
18	XDO_?XDOFIELD7?	<?fld_NAME?>	
19	XDO_GROUP_?XDOG2?	<xsl:for-each select=".//row_X_PAYTAX_ST_DVW">	</xsl:for-each>
20	XDO_?XDOFIELD8?	<?sum(.//fld_TAX_CUR)?>	

Test detail report by clicking on the Excel icon before continuing.

Grouping on Separate worksheets in the workbook

A feature of the Excel template is to split groups into multiple worksheets: one group per sheet.

An important point here is that the group/spilt is not handled through the "Repeating Group" control on the ribbon as the previous example, but through the XDO_METADATA sheet with an XDO mappings.

XDO_SHEET_?	Defines which record/row to split the data into sheets
XDO_SHEET_NAME_?	Defines what to name the sheet

Both XDO commands are required to make the split on the group work.

Using the previous taxes grouped by state example:
- We will split the data into a different sheet for each state
- The sheet name will be the State

First, remove the high-level STATE grouping from the previous example. The only repeating group control will be on the tax detail lines for each State

To remove the grouping:
- Click on the "Field Browser" control on the BI Publisher Ribbon
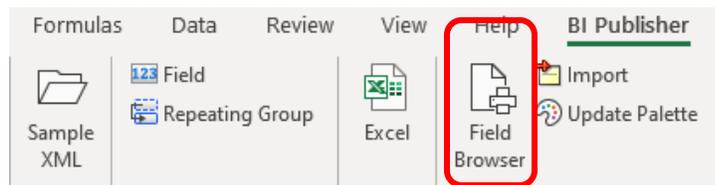
- Highlight the state grouping
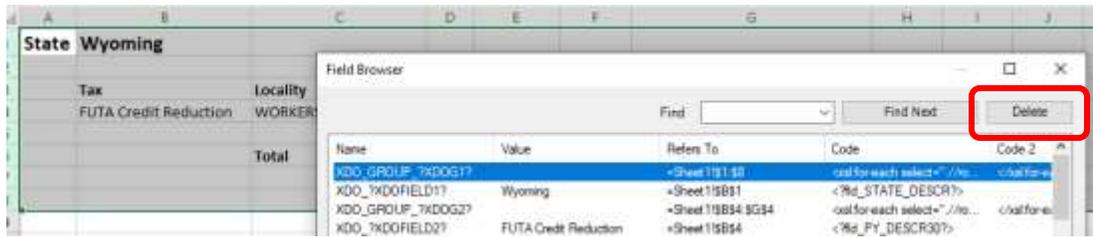
- Click the delete button (Upper right corner)
- Close the field browser

Open the XDO_METADATA sheet, navigate to the "Constraints" section and add the following rows

Column A	Column B
XDO_SHEET_?	<?.//row_X_PAYTAX_ST_VW?>
XDO_SHEET_NAME_?	<?.//fld_STATE_DESCR?>

"XDO_SHEET_?" controls which repeating ROW BI Publisher will split into different sheets. Not the grouping field. This will force the grouping, which is why we don't want to have a "Repeating Group" control at this level.

"XDO_SHEET_NAME_?" determines the unique name for each split sheet of the group.   You can use XLT functions here such as "concat()" and aggregate functions on fields for a meaningful sheet name.

The detail data row still requires the "Repeating Field" control to show each data row of the group.

Test detail report by clicking on the Excel icon before continuing.

Advanced Ideas

XDO Functions for Excel Templates

[Advanced Excel Template Functions](#)

This Oracle document lists many of the XDO functions available for Excel templates.

Field Browser
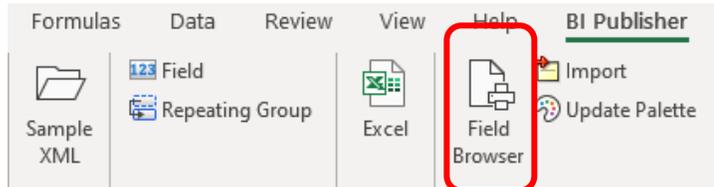
The "Field Browser" control on the BI Publisher ribbon displays all currently mapped fields and repeating groups in the current Excel Template.
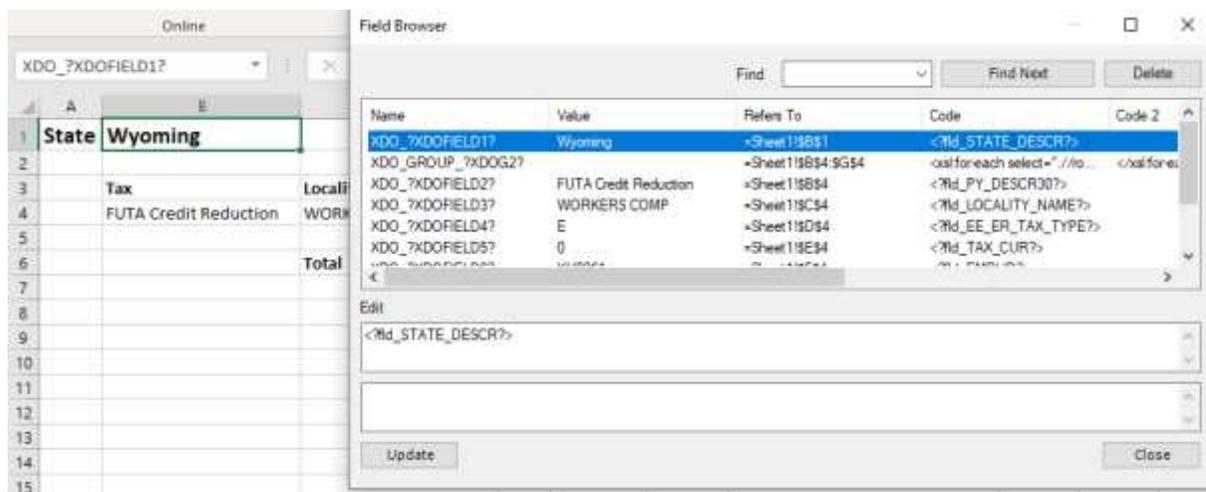
Field mapping can be viewed and updated and deleted through this control.

As a note, this does not show all XDO functions such as the sheet spitting controls, which are available on the XDO_METADATA sheet.

Name Manager

The Name Manager is available on the "Formulas" Ribbon menu.   This shows the cell, or the cell range associated with a template mapping.
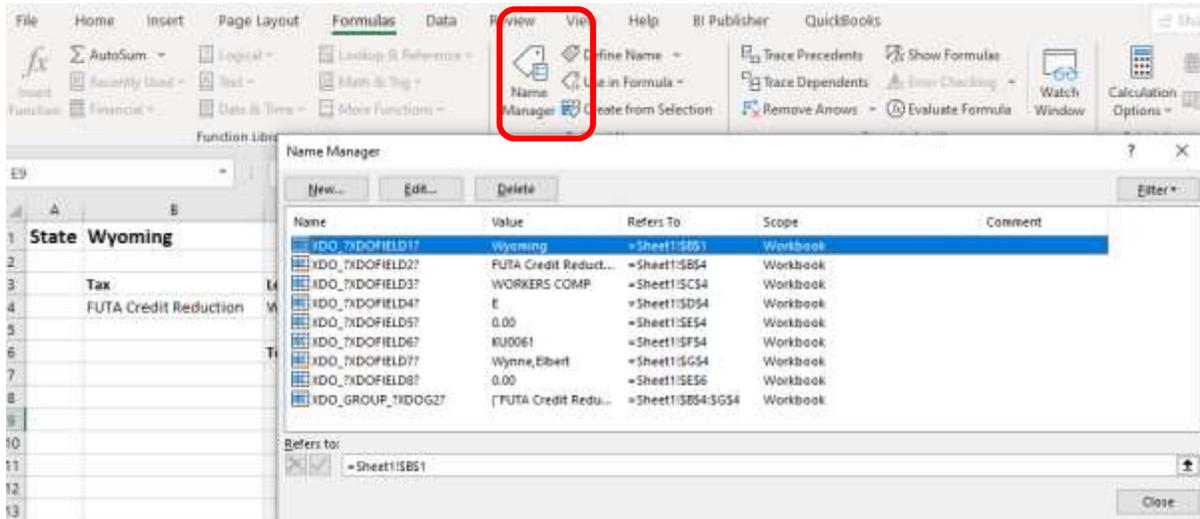
The "Field Browser" will show the mapping between the XML datafile and a Named Space, but editing and sometimes seeing the cell locations and ranges referenced by that Named Space must be done here.

Appendix A – Flat Tax XML File Example

Note: Data shortened to only two rows to show structure

```xml
<?xml version="1.0" encoding="UTF-8"?>
<rs_X_PAYTAX_VW xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" numrows="3"
rowsetname="X_PAYTAX_VW" xsi:noNamespaceSchemaLocation="">
    <row_X_PAYTAX_VW rownumber="1">
        <fld_COMPANY>GBI</fld_COMPANY>
        <fld_PAYGROUP>KU3</fld_PAYGROUP>
        <fld_PAY_END_DT>2020-03-15</fld_PAY_END_DT>
        <fld_OFF_CYCLE>N</fld_OFF_CYCLE>
        <fld_PAGE_NUM>2</fld_PAGE_NUM>
        <fld_LINE_NUM>1</fld_LINE_NUM>
        <fld_SEPCHK>0</fld_SEPCHK>
        <fld_STATE>GA</fld_STATE>
        <fld_EMPLID>KU0107</fld_EMPLID>
        <fld_NAME>Doyle,Emma</fld_NAME>
        <fld_STATE_DESCR>Georgia</fld_STATE_DESCR>
        <fld_LOCALITY />
        <fld_LOCALITY_NAME />
        <fld_TAX_CLASS>6</fld_TAX_CLASS>
        <fld_PY_DESCR30>FUTA Credit Reduction</fld_PY_DESCR30>
        <fld_EE_ER_TAX_TYPE>E</fld_EE_ER_TAX_TYPE>
        <fld_TAX_CUR>0</fld_TAX_CUR>
    </row_X_PAYTAX_VW>
    <row_X_PAYTAX_VW rownumber="2">
        <fld_COMPANY>GBI</fld_COMPANY>
        <fld_PAYGROUP>KU3</fld_PAYGROUP>
        <fld_PAY_END_DT>2020-03-15</fld_PAY_END_DT>
        <fld_OFF_CYCLE>N</fld_OFF_CYCLE>
        <fld_PAGE_NUM>2</fld_PAGE_NUM>
        <fld_LINE_NUM>1</fld_LINE_NUM>
        <fld_SEPCHK>0</fld_SEPCHK>
        <fld_STATE>GA</fld_STATE>
        <fld_EMPLID>KU0107</fld_EMPLID>
        <fld_NAME>Doyle,Emma</fld_NAME>
        <fld_STATE_DESCR>Georgia</fld_STATE_DESCR>
        <fld_LOCALITY />
        <fld_LOCALITY_NAME />
        <fld_TAX_CLASS>U</fld_TAX_CLASS>
        <fld_PY_DESCR30>Unemployment ER</fld_PY_DESCR30>
        <fld_EE_ER_TAX_TYPE>R</fld_EE_ER_TAX_TYPE>
        <fld_TAX_CUR>42.32</fld_TAX_CUR>
    </row_X_PAYTAX_VW>
</rs_X_PAYTAX_VW>
```

Appendix B – Complex Tax XML File Example

Note: Data shortened to only two rows to show structure

```xml
<?xml version="1.0" encoding="UTF-8"?>
<rs_X_PAYTAX_ST_VW xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" numrows="1"
rowsetname="X_PAYTAX_ST_VW" xsi:noNamespaceSchemaLocation="">
    <row_X_PAYTAX_ST_VW rownumber="1">
        <fld_COMPANY>GBI</fld_COMPANY>
        <fld_PAY_END_DT>2020-03-15</fld_PAY_END_DT>
        <fld_STATE>GA</fld_STATE>
        <fld_STATE_DESCR>Georgia</fld_STATE_DESCR>
        <rs_X_PAYTAX_ST_DVW numrows="3" rowsetname="X_PAYTAX_ST_DVW">
            <row_X_PAYTAX_ST_DVW rownumber="1">
                <fld_COMPANY>GBI</fld_COMPANY>
                <fld_PAY_END_DT>2020-03-15</fld_PAY_END_DT>
                <fld_STATE>GA</fld_STATE>
                <fld_PAYGROUP>KU3</fld_PAYGROUP>
                <fld_OFF_CYCLE>N</fld_OFF_CYCLE>
                <fld_PAGE_NUM>2</fld_PAGE_NUM>
                <fld_LINE_NUM>1</fld_LINE_NUM>
                <fld_SEPCHK>0</fld_SEPCHK>
                <fld_EMPLID>KU0107</fld_EMPLID>
                <fld_NAME>Doyle,Emma</fld_NAME>
                <fld_LOCALITY />
                <fld_LOCALITY_NAME />
                <fld_TAX_CLASS>6</fld_TAX_CLASS>
                <fld_PY_DESCR30>FUTA Credit Reduction</fld_PY_DESCR30>
                <fld_EE_ER_TAX_TYPE>E</fld_EE_ER_TAX_TYPE>
                <fld_TAX_CUR>0</fld_TAX_CUR>
            </row_X_PAYTAX_ST_DVW>
            <row_X_PAYTAX_ST_DVW rownumber="2">
                <fld_COMPANY>GBI</fld_COMPANY>
                <fld_PAY_END_DT>2020-03-15</fld_PAY_END_DT>
                <fld_STATE>GA</fld_STATE>
                <fld_PAYGROUP>KU3</fld_PAYGROUP>
                <fld_OFF_CYCLE>N</fld_OFF_CYCLE>
                <fld_PAGE_NUM>2</fld_PAGE_NUM>
                <fld_LINE_NUM>1</fld_LINE_NUM>
                <fld_SEPCHK>0</fld_SEPCHK>
                <fld_EMPLID>KU0107</fld_EMPLID>
                <fld_NAME>Doyle,Emma</fld_NAME>
                <fld_LOCALITY />
                <fld_LOCALITY_NAME />
                <fld_TAX_CLASS>H</fld_TAX_CLASS>
                <fld_PY_DESCR30>Withholding</fld_PY_DESCR30>
                <fld_EE_ER_TAX_TYPE>E</fld_EE_ER_TAX_TYPE>
                <fld_TAX_CUR>79.34</fld_TAX_CUR>
            </row_X_PAYTAX_ST_DVW>
        </rs_X_PAYTAX_ST_DVW>
    </row_X_PAYTAX_ST_VW>
</rs_X_PAYTAX_ST_VW>
```