

Appendix 14 Using Spreadsheet Data to Create CARS XML Invoices

This appendix describes processes that are useful for extracting data from spreadsheet files, converting the data to XML-usable content, and using the built-in Excel® XML functions to automatically generate tagged data for repeating blocks of invoice details such as direct labor or costs. The tagged data can then be copied and carefully pasted into an XML invoice that can be imported into CARS. Use this process for repeating blocks of details such as direct labor and direct costs in actual cost (cost plus) or specific rate invoices. You can enter non-repeating data for the invoice directly in the XML document. Invoices for lump sum and cost per unit contracts as a whole are simple enough that you can either edit the sample XML invoice directly or enter them manually in CARS.

This appendix is intended as guidance. It does not provide all-inclusive solutions. Every consultant firm uses its own unique system for gathering and storing invoice data, and there may be peculiarities with your data that are not covered in this appendix. For shorter invoices, you may also find that it is just as fast to manually key invoice data directly into CARS, at least until you become comfortable with using Excel functions.

Because most data systems can export data to a spreadsheet and Excel is a common and robust spreadsheet program, the examples in this appendix use Excel 2007 (version 12). The steps needed for the Excel conversions may vary slightly if you use a different version of Excel. There may be slight differences in the text manipulation functions, and there are differences in the XML function capabilities in different versions.

The goal is to take information in a spreadsheet and format it so that it can be used to automatically generate blocks of tagged invoice data. The tagged data must exactly conform to the CARS schema for content and format. Each cost line, such as an actual cost line item, has several non-repeating lines which can be easily entered into the XML document, including the payment type key, DBE key, and others. However, the repeating blocks of detail lines that include name, date, hours, cost, and other data can be tedious and time-consuming to enter manually. For example, an invoice might have 50 lines of direct labor details. For each detail line, you must include 11 individual data items. If you properly format an Excel spreadsheet and apply the CARS schema, you can automatically generate the 550 lines of tagged code and paste these lines into a CARS XML invoice document. This appendix is intended to show you how to accomplish the steps needed to generate XML code using Excel and manually paste it into the correct location in the CARS XML invoice.

Using a spreadsheet to partially automate the process of creating an XML invoice as described in this appendix is only one of many possibilities. XML invoices can be created by typing plain text, but manual editing could be tedious and error-prone for longer invoices. Invoice data could be entered directly into a spreadsheet and exported as XML. Spreadsheets are not the only program type useful for data manipulation; database applications may also be used. It is possible to fully automate the process. Again, each consultant firm has the option to use XML for invoicing in CARS, but each system used to generate the invoices will be unique.



General Process Steps

The general process steps for converting spreadsheet data into XML include the following:

1. Evaluate data exported from the financial system to a spreadsheet, either as a formatted report or as unformatted data. Ordinarily, you need to be most concerned with repetitive data such as direct labor and direct costs.
2. Convert data into the required parts as needed for the XML based on the cost basis for the invoice. Refer to the CARS schema to determine the necessary data.
3. Copy the prepared data to a worksheet using one column per data element. If you have more than one block of data to export, you need a separate worksheet for each data block. For example if you have both direct labor and direct costs for an invoice, you will have at least 3 worksheets:
 - a. The data exported from your financial software system
 - b. A worksheet for direct labor
 - c. A worksheet for direct costs

Note: Steps 2 and 3 can be incorporated into an Excel macro to speed up the invoicing process.

4. Associate the CARS XML format with the spreadsheet.
5. Apply the CARS XML format to the data.
6. Export tagged XML to a document.
7. Edit a sample invoice for the appropriate cost basis with your contract data. This edited document will become your XML invoice. You will edit the header, invoice information, and the line item blocks, and then paste in the tagged direct labor and direct cost data exported from Excel.
8. Copy the tagged data into the appropriate block in the XML document.

You will need the following items for this process. Items 2 and 3 are available on the [CARS Page](http://www.dot.wisconsin.gov/business/engrserv/cars.htm) (<http://www.dot.wisconsin.gov/business/engrserv/cars.htm>).

1. An Excel spreadsheet with your financial data. This data may be entered as unformatted data, or it could be formatted as a printable report from which the formatting will be removed.
2. *CARS Invoice XML for Excel Export* as an XML file (also called the “CARS XML file” in this appendix).
3. A sample invoice for the appropriate cost basis that you will use as a starting point to enter your data.
4. Optional but strongly recommended: an XML editor that offers features that help you easily read the text. Notepad++ is a free code editor that offers color-differentiated content, line numbers and code block tools. It is available for download from <http://notepad-plus-plus.org/>.
5. Basic understanding of XML. Suggested resources include W3 (<http://www.w3schools.com/xml/default.asp>), sitepoint (<http://www.sitepoint.com/really-good-introduction-xml/>), Wikipedia (<http://en.wikipedia.org/wiki/XML>), and appendix 13 of the CARS manual (for CARS-specific details).



Consultants: Evaluate Data

Commonly, data exported from financial systems to a spreadsheet are as a formatted report or invoice, or as raw data in an unformatted spreadsheet. Note that report or invoice formatting can add to the data cleanup processes. The exported data you can get will vary with the financial software system you use

Formatted Spreadsheets

Red arrows: Look for collapsed or merged rows and columns. These areas may contain hidden data or formulas.

Green arrows: Look for related data in separate rows or columns. Separating related information makes it more difficult to make sure the invoice cost lines are accurate.

Blue arrows: Look for data items combined in one cell, possibly with unneeded data. The data items will need to be separated into items according to CARS schema specifications. See *Convert Data* on page 217 for details on how to use Excel formulas to quickly separate data elements.

Purple arrows: Look for unneeded data. This data will need to be removed before applying the CARS XML file *CARS Invoice XML for Excel Export*.

The screenshot shows an Excel spreadsheet with the following data:

Invoice Information:

- Invoice number: 3242
- Date: 10/31/2014
- Project: 0687-22-63 STH 00, Spring Hollow -
- Professional Services: October 1st, 2014 - October 31st, 2014

Invoice Summary:

Description	Contract	Prior	Current	Total	Remaining
Services	124,985.65	91,341.45	7,633.06	98,974.51	26,011.14
Total	124,985.65	91,341.45	7,633.06	98,974.51	26,011.14

Professional Fees:

	Hours	Cost	Cost	Billed	
Assistant Project Leader					
Marshall G. Logan	10.00	38.50	385.00		
Real Estate Specialist					
Linda B. Anderson	37.25	35.00	1,303.75		
Team Leader					
Perry S. Grant	7.00	51.27	358.89		
Accountant					
David M. Pollard	3.75	21.05	78.94		
Project Leader					
Javier Garcia-Rivera	23.00	32.00	736.00		
Subtotal	81.00		2,862.58		
Overhead		136.75%	2,862.58	3,914.58	6,777.16
Fixed Fee		7.25%	6,777.16	491.34	7,268.50
Professional Fees subtotal					7,268.50

Reimbursables:

	Units	Rate	Billed
Javier Garcia-Rivera			
Mileage - Employee Vehicle	505.00	0.56	282.80
Linda B. Anderson			
Mileage - Employee Vehicle	146.00	0.56	81.76
Reimbursables subtotal			364.56

Invoice total: 7,633.06

If you have any questions, please contact Perry Grant @ (800) 555-4845. All invoices are due upon receipt. There will be a monthly



Consultants: Evaluate the Spreadsheet Data - continued

Formatted Spreadsheets - continued

Orange Arrows: Look for indications of code changes. The heading indicated by the orange arrow on this image marks a change in hours type.

Green arrows: Look for related data in separate rows or columns. Separating related information makes it more difficult to make sure the invoice cost lines are accurate.

Blue arrows: Look for data items combined in one cell, possibly with unneeded data. The data items will need to be separated into items according to CARS schema specifications. See *Convert Data* on page 217 for details on how to use Excel formulas to quickly separate data elements.

Purple arrows: Look for unneeded data. This data will need to be removed before applying the CARS XML file *CARS Invoice XML for Excel Export*.

Billing Detail						Friday, December 19, 2014	
Osage & Co.						Invoice 00395031 Dated 12/18/2014	
Osage Project: 701-0014 0687-00-77 USH 00							
Phase: 001 0687-00-77 Madison DAY SHIFT 6AM- 6PM							
Direct Labor:						Hours	Rate
272 - Inspection - General Field							
000147	2720 -Bennett, James	10/27/2014		12.00	47.31	579.71	
274 - Materials Quality Control							
000421	2740 -Richardson,	10/28/2014		4.50	29.93	143.66	
881.1 - Project Administration							
000641	881A - Kelly, Myra	11/5/2014		1.00	19.90	20.60	
740 - Project Dev - Admin/Coordination							
000071	7400 -Michaels,	11/5/2014		.25	60.18	15.62	
Total Direct Labor:						375.50	
Total Direct and Indirect:						2.3234 times	13,457.55
Subcontractors:							
Reimbursable Consultant Costs							
AP	00014358	10/31/2014	Acme Corporation/ Services rendered from			1,152.21	
Total Subcontractor Labor:						1.0 times	1,152.21
Direct Expenses:							
Reimbursable Mileage							
EX	00019037	10/27/2014	Miller, Arnold / drive to job (19) miles / drive to			10.64	
Total Direct Expenses:						1.0 times	1,417.92
Reimbursable Expenses:							
Co Vehicle Mileage							
10/27/2014	Co Vehicle Mileage 2014	108.0	Springfield-MI @ 0.61			65.88	
Total Reimbursable Expenses:						1.0 times	575.84
Total this Phase							\$34,413.24
Phase: 002 0687-00-77 Madison NIGHT SHIFT 6PM - 6AM							
Direct Labor:						Hours	Rate
272 - Inspection - General Field							
Const Services - OVT / Saturday Premium							
000152	C020 - Shelley, Paul	10/28/2014		8.00	31.00	248.00	
000152	C020 - Shelley, Paul	10/30/2014		8.00	31.00	248.00	
Total Direct Labor:						74.00	2,634.39
Total Direct and Indirect:						2.3234 times	2,634.39
Direct Expenses:							
Reimbursable Mileage							
EX	00019082	10/28/2014	Shelley, Paul / Osage - Madison - Osage /			34.16	
EX	00019093	11/3/2014	Hutton, Frederico / Travel to Madison, WI /			58.24	
Total Direct Expenses:						1.0 times	247.52
Total this Phase							\$6,368.26
Total this Project							\$40,781.50



Consultants: Evaluate the Spreadsheet Data - continued

Raw Data in Unformatted Spreadsheets

Blue arrows: Look for data items combined in one cell, possibly with unneeded data. The data items will need to be separated into items according to CARS schema specifications. See *Convert Data* on page 217 for details on how to use Excel formulas to quickly separate data elements.

Purple arrows: Look for unneeded data. This data will need to be removed before applying the CARS XML file *CARS Invoice XML for Excel Export*.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	Transaction Date	Transaction ID	Employee Full Nam	Rate Code	Rate Value	Billable Time	Billable Cost	Billing Sta	Project Nickname	Quantity	Unbillable	Unbillable	Activity Nickname	Price	
1	9/17/2014	298713	RJS Smith-1	16	\$4.64	0:30:00	0.50	\$23.32	Billable	LPM7 0687-22-57	0	0:00:00	\$0.00	106/Env Docs	\$0.00
2	9/24/2014	298671	CAM Murphy-1	16	\$118.66	1:30:00	1.50	\$177.99	Billable	LPM7 0687-22-57	0	0:00:00	\$0.00	R/E	\$0.00
3	9/24/2014	298814	CAM Murphy-1	16	\$118.66	1:00:00	1.00	\$118.66	Billable	LPM7 0687-22-57	0	0:00:00	\$0.00	R/E	\$0.00
4	10/5/2014	298948	CAM Murphy-1	16	\$118.66	1:30:00	1.50	\$177.99	Billable	LPM7 0687-22-57	0	0:00:00	\$0.00	R/E	\$0.00
5	10/14/2014	299099	CAM Murphy-1	16	\$118.66	0:30:00	0.50	\$59.33	Billable	LPM7 0687-22-57	0	0:00:00	\$0.00	R/E	\$0.00
6	9/24/2014	298739	JMG Green-1	16	\$132.63	6:00:00	6.00	\$795.78	Billable	LPM7 0687-22-57	0	0:00:00	\$0.00	PS&E	\$0.00
7	10/5/2014	298888	JMG Green-1	16	\$132.63	2:00:00	2.00	\$265.26	Billable	LPM7 0687-22-57	0	0:00:00	\$0.00	PS&E	\$0.00
8	9/24/2014	298645	KRB Billingsly-1	16	\$106.95	1:00:00	1.00	\$106.95	Billable	LPM7 0687-22-57	0	0:00:00	\$0.00	106/Env Docs	\$0.00
9	9/24/2014	298650	KRB Billingsly-1	16	\$106.95	4:30:00	4.50	\$481.28	Billable	LPM7 0687-22-57	0	0:00:00	\$0.00	PS&E	\$0.00
10	9/24/2014	298787	KRB Billingsly-1	16	\$106.95	1:00:00	1.00	\$106.95	Billable	LPM7 0687-22-57	0	0:00:00	\$0.00	Project Design	\$0.00
11	9/24/2014	298786	KRB Billingsly-1	16	\$106.95	10:30:00	10.50	\$1,122.98	Billable	LPM7 0687-22-57	0	0:00:00	\$0.00	PS&E	\$0.00
12	10/5/2014	298927	KRB Billingsly-1	16	\$106.95	1:00:00	1.00	\$106.95	Billable	LPM7 0687-22-57	0	0:00:00	\$0.00	Project Design	\$0.00
13	10/5/2014	298934	KRB Billingsly-1	16	\$106.95	6:00:00	6.00	\$641.70	Billable	LPM7 0687-22-57	0	0:00:00	\$0.00	PS&E	\$0.00
14	10/14/2014	298069	KRB Billingsly-1	16	\$106.95	1:30:00	1.50	\$160.43	Billable	LPM7 0687-22-57	0	0:00:00	\$0.00	PS&E	\$0.00
15	9/24/2014	298752	MJW Wilson-2	16	\$99.56	1:00:00	1.00	\$99.56	Billable	LPM7 0687-22-57	0	0:00:00	\$0.00	Agreement	\$0.00
16	10/14/2014	299032	MJW Wilson-2	16	\$99.56	0:30:00	0.50	\$49.78	Billable	LPM7 0687-22-57	0	0:00:00	\$0.00	R/E	\$0.00
17	9/24/2014	298844	AGC Collins-1	16	\$99.75	1:30:00	1.50	\$149.63	Billable	LPM7 0687-22-57	0	0:00:00	\$0.00	PS&E	\$0.00
18															



Consultants: Convert Data

Remove Formatting

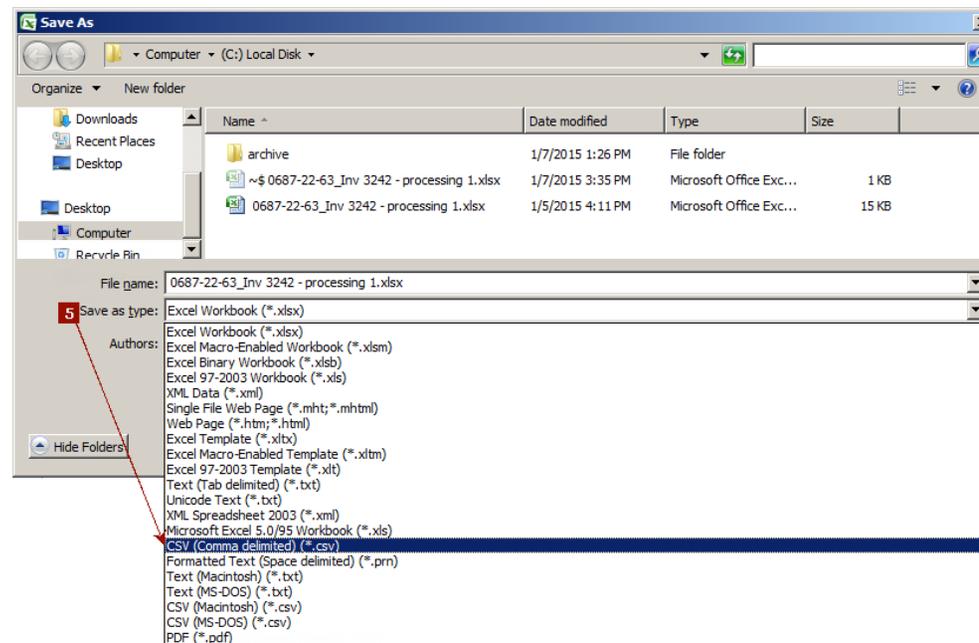
The simplest method for removing formatting from an Excel spreadsheet is to save it as a comma-separated value (CSV) document.

The steps used to access the **Save As** command may vary by Excel version.

1. Click the **Office** button .
2. Point to **Save As**.
3. Hover over the down arrow on the bottom of the **Save a copy of the document** list to show more options.
4. Click **Other Formats** at the bottom of the list. Excel opens the **Save As** dialog box.

Note: The keyboard shortcut to the **Save As** dialog box is ALT + F, then A.

5. In the **Save as type** field, select **CSV (Comma delimited) (*.csv)**. Do not use the CSV formats for Macintosh or MS-DOS.



Consultants: Convert Data - continued

Split Data into Required Parts

This process uses added columns to manipulate the data. You may also be able to use the Excel [Convert Text to Columns wizard](https://support.office.com/en-au/article/Split-names-by-using-the-Convert-Text-to-Columns-Wizard-39f7b055-6b39-4cb5-9512-13cc19b3a807) (https://support.office.com/en-au/article/Split-names-by-using-the-Convert-Text-to-Columns-Wizard-39f7b055-6b39-4cb5-9512-13cc19b3a807).

Extract Employee First Name

1. Adjust the width of the column that contains the employee names so that you can clearly see the names.
2. If needed, insert four new empty columns after the column containing the full name. Two columns will contain the Excel formulas, and the other two will contain the extracted name data. In the examples shown, there was enough room in the spreadsheet without inserting new columns. You may also use rows if that helps keep the data organized better.
3. In the first new column beside the first name (B18 in the example), enter “=LEFT(A18, (FIND(" ",A18)-1))” where “A18” is the cell reference for the original column, and there is a space between the quotation marks in the FIND function to find the first space in the data. See *Excel Formulas* on page 223 for more formulas you can use to extract data in cells. The formula is editable in the formula bar.
4. Click the check mark on the formula bar to approve the formula. Excel displays the result of the formula that is now in this cell. In this case, the result is “Marshall”.

Note: The CARS schema requires a value for employee first name, but you can simply use the first initial.

5. Paste the formula from B18 in the other cells that contain employee names. Excel automatically updates the cell references. Be aware that you are copying the formula, not the resulting name data.

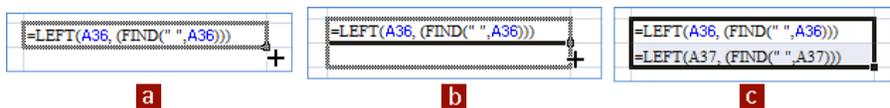
Note: You can use Excel’s fill-down function to quickly copy data in a column, including formulas.

- a. Select the cell and hover over the lower right corner until the cursor changes to a black “+”.
- b. Click and drag down.
- c. Release the mouse button to fill in the data or formulas.

	A	B	C
17	Assistant Project Leader		
18	Marshall G. Logan	=LEFT(A18, (FIND(" ",A18)))	
19	Real Estate Specialist		
20	Linda B. Anderson		

	A	B	C
17	Assistant Project Leader		
18	Marshall G. Logan	Marshall	
19	Real Estate Specialist		
20	Linda B. Anderson		

	A	B	C
17	Assistant Project Leader		
18	Marshall G. Logan	Marshall	
19	Real Estate Specialist		
20	Linda B. Anderson	Linda	



Consultants: Convert Data - continued

- Copy all of the cells that contain the Excel formulas as a block.
- Click the cell next to the first cell containing a formula.
- Right-click and select **Paste Special...** Excel opens the **Paste Special** dialog box.
- Click **Values**.
- Click **OK**. Excel pastes the first name values in the column.
- Review the copied values for accuracy and make any needed corrections directly in the copied value cells.

You can now delete the formula column or delete the existing formulas and replace them with new formulas to extract other data, such as last names. Repeat the steps for copying formulas and pasting the resulting values into new columns.

Note: You can show and hide all of the formulas (Formula View) in the worksheet at once using the key combination **CRTL + `** (grave accent, which is the key left of numeric 1 above TAB).

Dates require specific date functions in Excel because they are not stored as month, day and year values, but as the number of days since January 1, 1900. The resulting number is referred to as a serial number. For example, the serial number for October 31, 2014 is 41943. Handling dates in this manner enables accurate date calculations. Cells containing dates must be formatted as dates, as opposed to text or general data. The cell format is shown on the ribbon in the **Number** group of the **Home** tab.

	A	B	C
17	Assistant Project Leader		
18	Marshall G. Logan	Marshall	
19	Real Estate Specialist		
20	Linda B. Anderson	Linda	
21	Team Leader		
22	Perry S. Grant	Perry	
23	Accountant		
24	David M. Pollard	David	
25	Project Leader		
26	Javier Garcia-Rivera	Javier	
27			
28			
29	Overhead		
30	Fixed Fee		
31	Professional Fees subtotal		
32	Reimbursables		
33			
34	Javier Garcia-Rivera	Javier	
35	Mileage - Employee Vehicle		
36	Linda B. Anderson	Linda	
37	Mileage - Employee Vehicle		



Excel Formulas

Replace A1 with the appropriate cell reference in each formula. These formulas may be adjusted based on the text patterns in the cells.

You can use these formulas for both formatted and unformatted spreadsheets.

For explanations of Excel text functions, visit <http://office.microsoft.com/en-001/excel-help/text-and-data-functions-HP005203209.aspx>.

Extract Text	Sample Formula	Data Example	Result Example
First character in a cell	=LEFT(A1, 1)	Linda B. Anderson	L
First text in a cell	=LEFT(A1, (FIND(" ",A1)-1)) This formula finds the characters before the first space	Marshall G. Logan	Marshall
First text in a cell	=LEFT(A1, (FIND(",",A1)-1)) This formula finds the characters after the first comma.	Bennett, James	Bennett
Last text in a cell	=TRIM(RIGHT(A1,(LEN(A1)-FIND(".", A1)))) This formula finds the characters after the first period.	Marshall G. Logan	Logan
Last text in a cell	=TRIM(RIGHT(A1,(LEN(A1)-FIND(" ", A1)))) This formula finds the characters after the first space.	Javier Garcia-Rivera	Garcia-Rivera
Last text in a cell	=RIGHT(A1,(LEN(A1)-FIND(" ", A1)))	Bennett, James	James
Text in the middle of a cell	=MID(A1,(FIND(" ", A1)+2),(LEN(A1)-(FIND(" ", A1)+2)-(LEN(A1)-(FIND(",",A1)))))	2720 -Bennett, James	Bennett
Day of date	=Day(A1) This formula finds the day value when the cell is formatted as "DATE".	10/31/2014	31
Month of date	=Month(A1)	10/31/2014	10
Year of date	=Year(A1)	10/31/2014	2014



Consultants: Copy Prepared Data

For this process, you need to know what data is needed for the cost basis specified for your contract. Refer to the sample invoice or the *CARS XML Detailed Element Descriptions* spreadsheet to determine what elements are included in the block and which data is required in each block.

Note: All elements are required in the XML invoice, even if they do not contain data. However, only one line item block is required - actual cost, lump sum, cost per unit, or specific rate. Use the sample invoices as examples.

For this example, the direct labor block for a specific rate basis invoice is used. The XML tags for this block are described on rows 126 to 135 on the *CARS XML Detailed Element Descriptions* spreadsheet. The tags for this block are listed below. Data is required for every tag except employeeNumber.

- | | |
|---------------------------------------|-----------------------|
| 1. employeeFirstName | 6. hours |
| 2. employeeLastName | 7. activity |
| 3. employeeNumber (data not required) | 8. laborCalendarMonth |
| 4. employeeClassification | 9. laborCalendarYear |
| 5. hoursType | 10. laborCostAmount |

Begin the copying process by creating a new worksheet in the existing workbook.

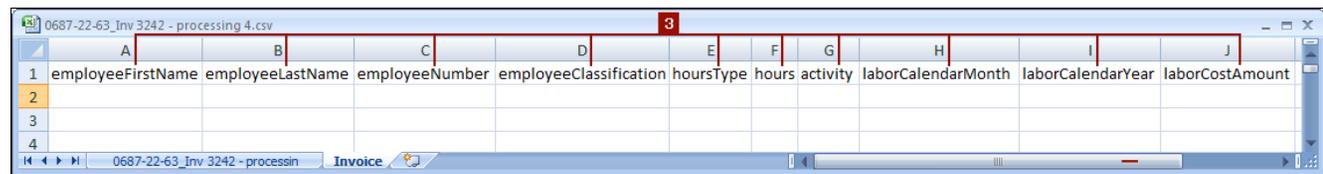
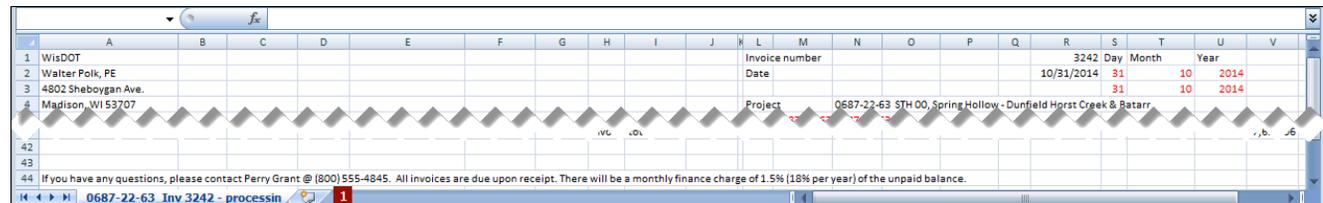
1. Click **Insert Worksheet**



to insert a new worksheet in the workbook.

2. Rename the new worksheet as needed. Right-click the "Sheet1" tab and click **Rename**. Then type the new worksheet name.

3. Enter tag names for each column in the first row (or copy and paste from the CARS XML file). Be sure to enter each tag as shown and in the order given in the XML invoice sample or the *CARS XML Detailed Element Descriptions* spreadsheet. Capitalization and spelling must match the CARS schema exactly.



Consultants: Copy Prepared Data - continued

- On the spreadsheet containing the data, copy the information from the data columns and paste it into the appropriate column on the "Invoice" worksheet. The data you need may be in raw form or it may be converted. The values that were converted are shown in red.
- Delete any blank rows. In this example, rows 3, 5, 7 and 9 will be removed.

Note: Be sure to copy only direct labor costs onto this invoice worksheet. The direct costs and overhead shown in rows 29 and following do not belong in the direct labor block.

- Save the workbook as an Excel workbook (.xlsx) file to keep both worksheets. If you save as a CSV, only one worksheet is saved.

	Formula	Fname	Formula	Lname	Hours	Cost Rate	Cost Amount	
16								
17								
18	Assistant Project Leader	Marshall	Marshall	Logan	Logan	10	38.5	385
19	Marshall G. Logan							
20	Real Estate Specialist	Linda	Linda	Anderson	Anderson	37.25	35	1303.75
21	Linda B. Anderson							
22	Team Leader	Perry	Perry	Grant	Grant	7	51.27	358.89
23	Perry S. Grant							
24	Accountant	David	David	Pollard	Pollard	3.75	21.05	78.94
25	David M. Pollard							
26	Project Leader	Javier	Javier	Garcia-Rivera	Garcia-Rivera	23	32	736
27	Subtotal					81		2862.58
28								
29	Overhead					2862.58	3914.58	6777.16
30	Fixed Fee					6777.16	491.34	7268.5
31	Professional Fees subtotal							
32	Reimbursables							
33								Units
34	Javier Garcia-Rivera	Javier	Javier	Garcia-Rivera	Garcia-Rivera			
35	Mileage - Employee Vehicle							505
36	Linda B. Anderson	Linda	Linda	Anderson	Anderson			
37	Mileage - Employee Vehicle							146
38	Reimbursables subtotal							

	A	B	C	D	E	F	G	H	I	J	K
1	employeeFirstName	employeeLastName	employeeNumber	employeeClassification	hoursType	hours	activity	laborCalendarMonth	laborCalendarYear	laborCost	Amount
2	Marshall	Logan		Assistant Project Leader		10					385
3											
4	Linda	Anderson		Real Estate Specialist		37.25					1303.75
5											
6	Perry	Grant		Team Leader		7					358.89
7											
8	David	Pollard		Accountant		3.75					78.94
9											
10	Javier	Garcia-Rivera		Project Leader		23					736
11											



Copy Prepared Data - continued

7. Enter all other required data. For repeating values, you can use the Excel fill-down function, which is described on page 219. You may also copy and paste values. For this example the following values are entered and shown in blue:
 - a. employeeNumber is supposed to be blank in all cases for this example. This value is not required in CARS or in the XML invoice. However, when no data is entered in the spreadsheet, Excel will not include the element in the exported XML. Remember that all elements in a block are required, even if they are empty. A zero or a period is added for each employee number for this example to make sure the element is exported by Excel. This character can be deleted from the XML generated by Excel by global find and replace if you use a text editor with that capability.
 - b. hoursType is set to "Hours Worked". This value is validated by lookup in CARS and must match the values stored in CARS exactly, including capitalization. Possible values are "Hours Worked" or "Premium". Premium pay is used only rarely.
 - c. activity is set to "9999" in each case. Activity codes are not currently required, but will be in a future release. When you enter other values for activities, enter only the 1 to 4 digit activity code. A list of current activity codes and descriptions can be found on the [WisDOT CARS Page](http://www.dot.wisconsin.gov/business/engrserv/cars.htm) (<http://www.dot.wisconsin.gov/business/engrserv/cars.htm>).
 - d. calendarMonth and calendarYear are set to "02" and "2015" respectively. Be sure to use the actual month and year the work was performed. Format the cells in the "calendarMonth" column as text to ensure that the required leading zero for a 2 digit date is retained.

	A	B	a C	D	b E	F	c G	H	d I	J
1	employeeFirstName	employeeLastName	employeeNumber	employeeClassification	hoursType	hours	activity	laborCalendarMonth	laborCalendarYear	laborCostAmount
2	Marshall	Logan	0	Assistant Project Leader	Hours Worked	10	9999	2	2015	385
3	Linda	Anderson	0	Real Estate Specialist	Hours Worked	37.25	9999	2	2015	1303.75
4	Perry	Grant	.	Team Leader	Hours Worked	7	9999	2	2015	358.89
5	David	Pollard	.	Accountant	Hours Worked	3.75	9999	2	2015	78.94
6	Javier	Garcia-Rivera	0	Project Leader	Hours Worked	23	9999	2	2015	736
7										

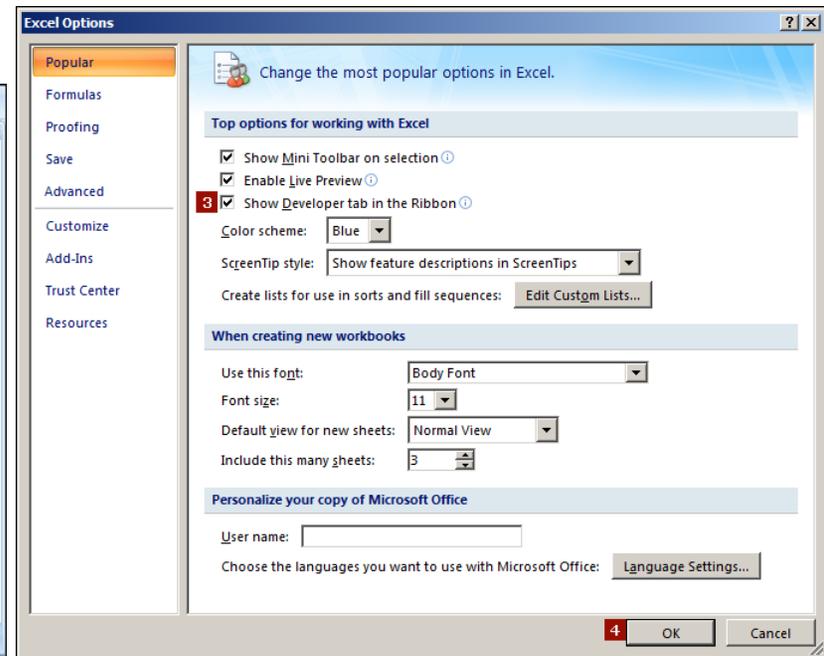


Consultants: Associate CARS XML Format

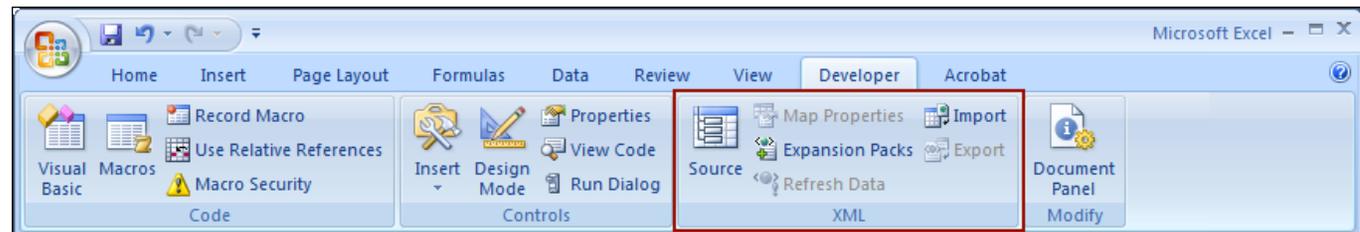
Show the Developer Tab

The next step is to associate the CARS XML file *CARS Invoice XML for Excel Export* with the Excel workbook. Begin this process by showing the **Developer** tab in Excel, which is where the XML tools reside.

 **Important:** Do not associate the CARS schema (XSD) with the spreadsheet. Excel cannot export repeating lists when you associate the spreadsheet with the XSD file, but it can if you associate the XML file.



1. Click the **Office** button .
2. Click **Excel Options**. Excel opens the **Excel Options** dialog box.
3. Select the **Show Developer tab in the Ribbon** check box.
4. Click **OK**. Excel shows the **Developer** tab, which contains the **XML** group.



Consultants: Associate CARS XML Format - continued

The next step is to associate the CARS XML file *CARS Invoice XML for Excel Export* with the workbook.

1. On the **Developer** tab in the **XML** group, click **Source**. Excel opens the **XML Source** pane.
2. Click **XML Maps....** Excel opens the **XML Maps** dialog box.
3. Click **Add**. Excel opens the **Select XML Source** dialog box.

The screenshot displays the Microsoft Excel interface with the **Developer** tab selected. The **XML Source** pane is open on the right, showing a list of XML maps in the workbook. The **XML Maps** dialog box is also open, showing a table with columns for Name, Root, and Namespace. The **Add...** button is highlighted with a red '3'. The **XML Source** pane has a red '2' next to the **XML Maps...** button. The **Source** button in the **Developer** tab has a red '1' next to it.

employeeFirstName	employeeLastName	employeeNumber	employeeClassification	hoursType	hours	activity	laborCalendarMonth	laborCalendarYear	laborCostAmount
Marshall	Logan		Assistant Project Leader	Hours Worked	10	9999 02	2015		385
Linda	Anderson		Real Estate Specialist	Hours Worked	37.25	9999 02	2015		1303.75
Perry	Grant		Team Leader	Hours Worked	7	9999 02	2015		358.89
David	Pollard		Accountant	Hours Worked	3.75	9999 02	2015		78.94
Javier	Garcia-Rivera		Project Leader	Hours Worked	23	9999 02	2015		736

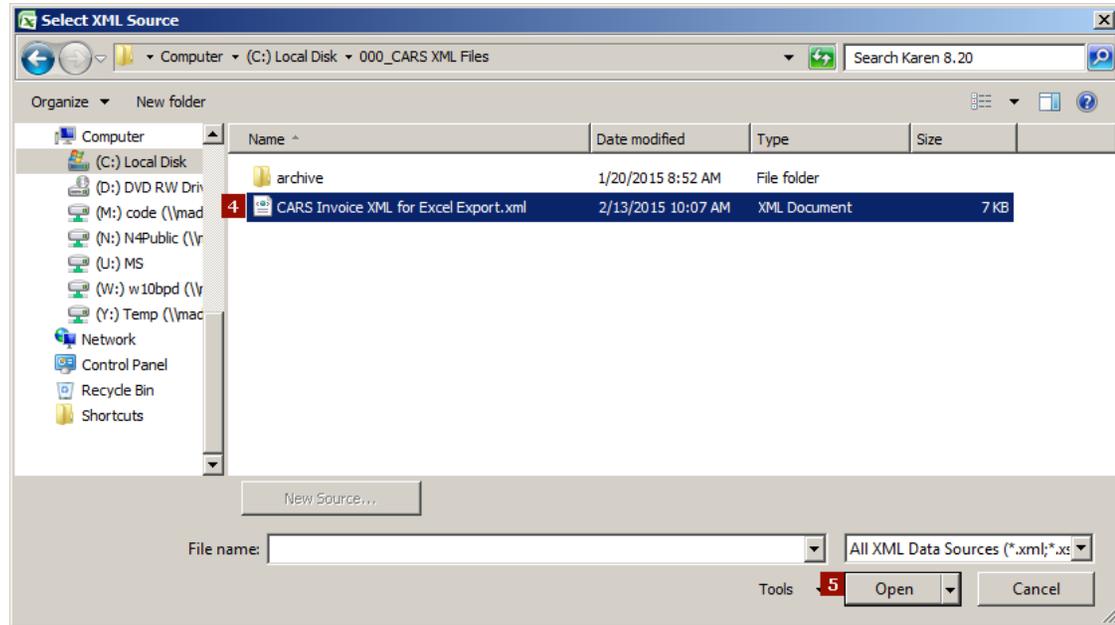


Consultants: Associate CARS XML Format - continued

4. Navigate to location of the CARS XML file *CARS Invoice XML for Excel Export* on your computer and select this file.
5. Click **Open**. Excel displays the **Microsoft Office Excel** message.
6. Click **OK**. Excel lists the CARS XML file as an XML source.
7. Click **OK** in the **XML Maps** dialog box. Excel displays the CARS XML file details in the **XML Source** pane.

The CARS XML file contains definitions for all 4 cost basis types. You can expand and collapse sections. The default view shows all blocks fully expanded.

Every invoice will include the header and invoice information blocks plus one of the invoice details blocks based on the cost basis for your project. When you are preparing the XML invoice, be certain that you are using the appropriate cost basis section.



The screenshot shows the Excel interface with the following elements:

- Spreadsheet:** A table with columns A through J. The data includes employee names (Marshall, Linda, Perry, David, Javier), last names (Logan, Anderson, Grant, Pollard, Garcia-Rivera), employee numbers, classifications (Assistant Project Leader, Real Estate Specialist, Team Leader, Accountant, Project Leader), hours worked, activity codes (9999), labor calendar months (2), labor calendar years (2015), and labor cost amounts (385, 1303.75, 358.89, 78.94, 736).
- XML Source Pane:** Located on the right, it shows a tree view of the XML source structure:
 - invoice_Map
 - invoice
 - okResponse
 - header
 - invoiceInformation
 - invoiceDetails
 - totalInvoiceAmount
 - actualCostLineItem
 - lumpSumLineItem
 - costPerUnitLineItem
 - specificRateLineItem

- XML Maps Dialog Box:** A dialog box with a table showing the mapping of XML elements to the spreadsheet:

Name	Root	Namespace
invoice_Map	invoice	<No Namespace>
- Microsoft Office Excel Message Box:** A message box with an information icon and the text: "The specified XML source does not refer to a schema. Excel will create a schema based on the XML source data." There is a checkbox for "In the future, do not show this message." and "OK" and "Help" buttons.


Consultants: Apply CARS XML Format

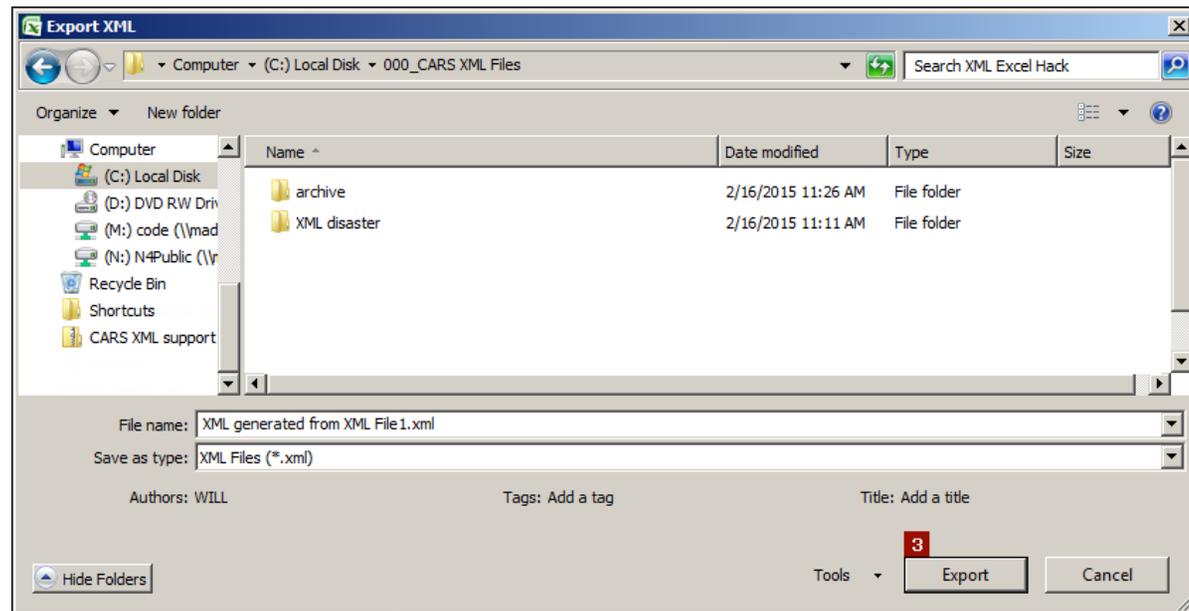
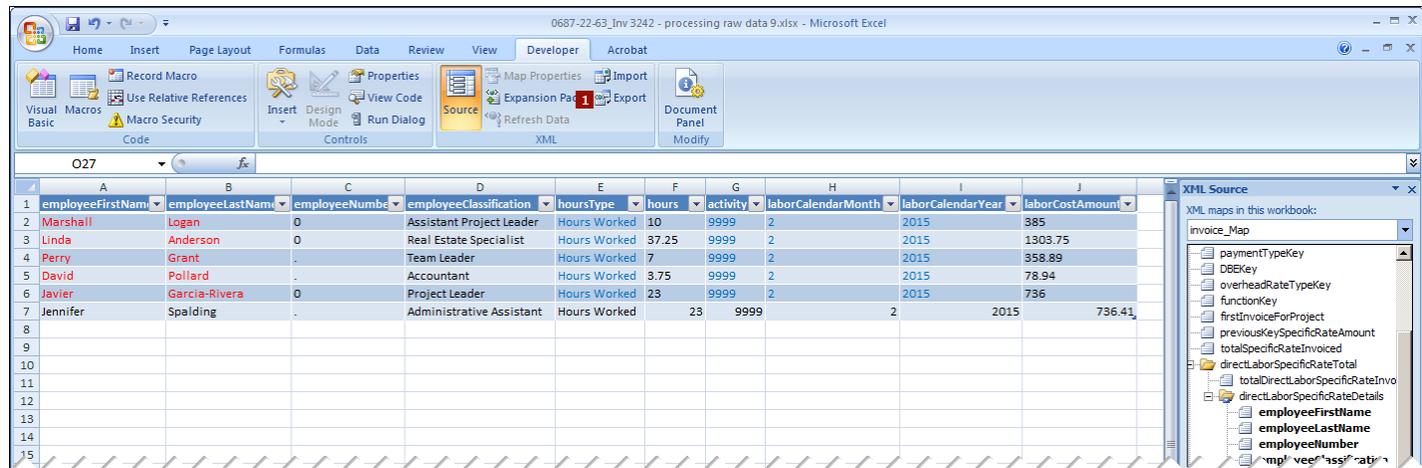
1. Expand the appropriate cost basis line item section in the **XML Source** pane. For this example, the `specificRateLineItem` section is needed.
2. Select the repeating parent element that contains the same XML elements as the worksheet. Excel selects the entire block.
3. Drag and drop the repeating parent element to cell A1, which contains the tag name for the first data element in the block. In this example, the tag name for the first element is "employeeFirstName". Excel maps the entire block to the tag names and data. Mapped elements are shown in bold in the XML source pane.

Note: If you add data to the spreadsheet after you complete this mapping step, the new data will be tagged and exported with the original data.



Consultants: Export Tagged Data

1. On the ribbon in the XML group, click **Export**.
2. In the **Export XML** dialog box, specify the location where the new XML file should be stored.
3. Click **Export**. Excel generates tagged XML data from the data in the worksheet.



Consultants: Export Tagged Data - continued

4. Navigate to the exported XML, open the file, and proofread or spot check the data.
5. If your XML editor application has find and replace capabilities, be sure to search and replace using at least part of the tag for the element to ensure that data you don't intend to change is not altered. For example to correct month values, search for "<laborCalendarMonth>2" and replace with "<laborCalendarMonth>02".

```

62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
<activity>9999</activity>
5 <laborCalendarMonth>2</laborCalendarMonth>
  <laborCalendarYear>2015</laborCalendarYear>
  <laborCostAmount>736</laborCostAmount>
</directLaborSpecificRateDetails>
<directLaborSpecificRateDetails>
  <employeeFirstName>Jennifer</employeeFirstName>
  <employeeLastName>Spalding</employeeLastName>
5 <employeeNumber>.</employeeNumber>
  <employeeClassification>Administrative Assistant</employeeClassification>
  <hoursType>Hours Worked</hoursType>
  <hours>23</hours>
  <activity>9999</activity>
5 <laborCalendarMonth>2</laborCalendarMonth>
  <laborCalendarYear>2015</laborCalendarYear>
  <laborCostAmount>736.41</laborCostAmount>
</directLaborSpecificRateDetails>

```

Exported Data as XML Before Editing

Places to find common errors:

- Missing elements: Make sure each detail block has the correct number of elements. For this example, each specific rate direct labor detail item has 10 elements.
- Dates: Month values should have 2 digits. The detail items still import, but CARS will notify you that lines having the incorrect month values are in error.
- Data added to export tags, such as employee numbers

```

61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
<hours>23</hours>
<activity>9999</activity>
5 <laborCalendarMonth>02</laborCalendarMonth>
  <laborCalendarYear>2015</laborCalendarYear>
  <laborCostAmount>736</laborCostAmount>
</directLaborSpecificRateDetails>
<directLaborSpecificRateDetails>
  <employeeFirstName>Jennifer</employeeFirstName>
  <employeeLastName>Spalding</employeeLastName>
5 <employeeNumber>.</employeeNumber>
  <employeeClassification>Administrative Assistant</employeeClassification>
  <hoursType>Hours Worked</hoursType>
  <hours>23</hours>
  <activity>9999</activity>
5 <laborCalendarMonth>02</laborCalendarMonth>
  <laborCalendarYear>2015</laborCalendarYear>
  <laborCostAmount>736.41</laborCostAmount>
</directLaborSpecificRateDetails>
</directLaborSpecificRateTotal>

```

Exported Data as XML After Editing



Consultants: Edit Sample Invoice

Exporting invoice data from Excel data as XML is the first part of creating the CARS XML invoice. The next part is to create the full XML invoice that is valid for CARS. One simple method for creating an XML invoice with data details that were exported from Excel begins with an “IMPORT” invoice for the appropriate cost basis. Using an existing invoice is easier than starting with an empty page because all the tags are already in the invoice. The “IMPORT” invoices provided on the [CARS Page](http://www.dot.wisconsin.gov/business/engrserv/cars.htm) (<http://www.dot.wisconsin.gov/business/engrserv/cars.htm>) also show the data needed to import an invoice. The data in these “IMPORT” invoices is meant to serve as either an examples or an explanation of the requirements. “BLANK” or an explanation within an element tag indicates that data is either required or conditionally required. If information is not required at all for an element, the element is left empty. The intent is that you edit a copy of one of these invoices with your data, and then import the XML invoice into CARS as described in the CARs manual.

Note: Editing the XML invoice is easier when you use an XML editor that offers enhancements to help you read it, such as using different colors of text for tags and data, line numbers, and the ability to collapse (hide) blocks. Notepad++ is a free code editor available for download from <http://notepad-plus-plus.org/>.

Consultants: Tags

When editing an XML invoice, pay close attention to the element tags. Be careful not to change or delete any of the tags. The spelling of the tags must match the CARS schema exactly, including case. Be aware that tags have two parts: a start and end. End tags have a slash after the first bracket, and start tags do not. Tags can be used to contain data elements, or they can contain blocks of other grouped data elements, or both. XML blocks can be nested, but they can never overlap.

To edit an element, enter data between the start and end tags.

```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <invoice>
3 <okResponse>
4   <header>
5     <vendorCompanyName></vendorCompanyName>
6     <vendorAddress></vendorAddress>
7     <vendorCity></vendorCity>
8     <vendorState></vendorState>
9     <vendorZip></vendorZip>
10  </header>
11  <invoiceInformation>
12    <invoiceTotalAmount></invoiceTotalAmount>
13    <projectInformation>
14      <vendorInvoiceNumber></vendorInvoiceNumber>
15      <projectDescription></projectDescription>
16      <stateProjectId></stateProjectId>
17    </projectInformation>
18  </invoiceInformation>
19 </okResponse>
20 </invoice>
21
22
```



Consultants: Complete Header

The invoice header block in the XML invoice contains some information that is consistent for every invoice you import. Your federalEmployeeIdentification (FEIN), location, company name, and address information will probably never change. The contact person and information may change. The current date needs to be updated with each invoice.

1. Enter your company's 9-digit FEIN between the federalEmployeeIdentification tags. This value is required.
2. Enter your company location in between the FEINLoc tags. This single-digit value is usually "0", but it may be another number if your company has multiple locations for the same FEIN registered in CARS. This value is required.
3. Enter your company's name, preferably as shown in CARS. Remember to substitute any special characters with XML predefined entities as described in Appendix 13 of this manual. This value is required.
4. Optionally, enter your company's address information in the 6 address elements. None of these elements are required.
5. Enter the name, phone number and email address of the person WisDOT should contact with questions about this invoice between the vendorContactName, vendorContactPhone, and vendorContactEmail tags. Phone numbers must contain 10 digits with dashes separating the area code, exchange and number. All of these values are required.
6. Subconsultants may want to enter prime vendor information for reference. None of these elements are required.
7. Enter the current date between the currentDate tags. This value is required. Dates are formatted as YYYYMMDD in CARS XML invoices. The date in the "Before" image is November 25, 2010.

```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <invoice>
3 <okResponse>
4 <header>
5 <federalEmployeeIdentification>99999999</federalEmployeeIdentification> 1
6 <FEINLoc>0</FEINLoc> 2
7 <currentDate>20101125</currentDate> 7
8 <vendorCompanyName>REQUIRED</vendorCompanyName> 3
9 <vendorAddress1></vendorAddress1> 4
10 <vendorAddress2></vendorAddress2>
11 <vendorCity></vendorCity>
12 <vendorState></vendorState>
13 <vendorZip></vendorZip>
14 <vendorZipExtn></vendorZipExtn>
15 <vendorContactName>REQUIRED</vendorContactName> 5
16 <vendorContactPhone>000-000-0000</vendorContactPhone>
17 <vendorContactEmail>email</vendorContactEmail>
18
19 <primeVendorCompanyName></primeVendorCompanyName> 6
20 <primeVendorCity></primeVendorCity>
21 <primeVendorState></primeVendorState>
22 <primeVendorZip></primeVendorZip>
23 </header>
```

"IMPORT" XML Invoice Header Before Editing

```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <invoice>
3 <okResponse>
4 <header>
5 <federalEmployeeIdentification>123456789</federalEmployeeIdentification>
6 <FEINLoc>0</FEINLoc>
7 <currentDate>20150501</currentDate>
8 <vendorCompanyName>Osage & Co.</vendorCompanyName>
9 <vendorAddress1></vendorAddress1>
10 <vendorAddress2></vendorAddress2>
11 <vendorCity></vendorCity>
12 <vendorState></vendorState>
13 <vendorZip></vendorZip>
14 <vendorZipExtn></vendorZipExtn>
15 <vendorContactName>Max Brewer</vendorContactName>
16 <vendorContactPhone>800-123-4567</vendorContactPhone>
17 <vendorContactEmail>max@osage.com</vendorContactEmail>
18
19 <primeVendorCompanyName>Acme Inc.</primeVendorCompanyName>
20 <primeVendorCity>Sauk City</primeVendorCity>
21 <primeVendorState>WI</primeVendorState>
22 <primeVendorZip>53583</primeVendorZip>
23 </header>
```



Consultants: Complete Invoice Information

The invoice information block in the XML invoice contains contract-specific information. The required elements are:

- vendorInvoiceNumber
 - timePeriodCoveredFrom
 - timePeriodCoveredTo
 - finalInvoiceFlag
 - zeroInvoiceFlag
 - stateProjectId **-OR-** masterContract plus workOrderNumber
1. Enter your firm's invoice reference number between the vendorInvoiceNumber tags. This number must be unique for invoices for a given project.
 2. Enter the beginning and ending dates of the period which the invoice covers between the timePeriodCoveredFrom and timePeriodCoveredTo tags. Dates are formatted as YYYYMMDD in CARS XML invoices.
 3. If this invoice is the final invoice for the project, enter "Y" between the finalInvoiceFlag tags. Otherwise, enter "N".
 4. If this invoice has no costs, enter "Y" between the zeroInvoiceFlag tags. Otherwise, enter "N".
 5. Enter the project ID with hyphens between the stateProjectId tags when you are billing for a regular contract.
-OR-
Enter the master contract ID without hyphens between the masterContract tags and the work order number between the workOrderNumber tags when you are billing for a work order.
 6. Optional: You may enter values between the projectDescription or the projectCounty tags for your reference. These values are not required.
 7. The subContractInvoiceFlag is also not required, and this tag is currently not in use.

```
<invoiceInformation>
  <vendorInvoiceNumber>BLANK</vendorInvoiceNumber>
  <projectDescription></projectDescription>
  <projectCounty></projectCounty>
  <subContractInvoiceFlag></subContractInvoiceFlag>
  <timePeriodCoveredFrom>20101101</timePeriodCoveredFrom>
  <timePeriodCoveredTo>20101130</timePeriodCoveredTo>
  <finalInvoiceFlag>Y or N</finalInvoiceFlag>
  <zeroInvoiceFlag>Y or N</zeroInvoiceFlag>
  <stateProjectId>0000-00-00</stateProjectId>
  <masterContract>M00000000 OPTIONAL</masterContract>
  <workOrderNumber>9 OPTIONAL</workOrderNumber>
</invoiceInformation>
```

"IMPORT" XML Invoice Information Before Editing

```
<invoiceInformation>
  <vendorInvoiceNumber>25302</vendorInvoiceNumber>
  <projectDescription>I39 Project</projectDescription>
  <projectCounty>Various</projectCounty>
  <subContractInvoiceFlag></subContractInvoiceFlag>
  <timePeriodCoveredFrom>201510101</timePeriodCoveredFrom>
  <timePeriodCoveredTo>201501</timePeriodCoveredTo>
  <finalInvoiceFlag>N</finalInvoiceFlag>
  <zeroInvoiceFlag>N</zeroInvoiceFlag>
  <stateProjectId>0687-11-25</stateProjectId>
  <masterContract></masterContract>
  <workOrderNumber></workOrderNumber>
</invoiceInformation>
```

"IMPORT" XML Invoice Information After Editing



Consultants: Complete Invoice Details

The Invoice details block contains 2 main parts:

- An element that may contain the total amount for the entire invoice
- A *Line Item* block that contains the other tags and blocks, which includes:
 - Line item data consisting of 7 tags
 - One or more *Total* blocks, which vary depending on the cost basis for the invoice.

For this example of a specific rate invoice, only the *Line Item* data is of concern for this part of the process. Refer to Appendix 13: *Import XML Invoice* for details about the four-part line item key.

1. Make sure the value shown between the `paymentTypeKey` tags is correct. For this example, the correct value is "Specific Rate".
2. Change the value between the `DBEKey` tags to "Y" or "N".
3. Enter the overhead rate type from the contract in CARS between the `overheadRateTypeKey` tags. This value must match the value in CARS exactly, including capitalization and spaces.

```
<invoiceDetails>
  <totalInvoiceAmount></totalInvoiceAmount> 7
  <specificRateLineItem>
    <paymentTypeKey>Specific Rate</paymentTypeKey> 1
    <DBEKey>Y or N</DBEKey> 2
    <overheadRateTypeKey>MUST MATCH VALID TYPE IN CARS</overheadRateTypeKey> 3
    <functionKey>MUST MATCH VALID TYPE IN CARS</functionKey> 4
    <firstInvoiceForProject>Y or N</firstInvoiceForProject> 5
    <previousKeySpecificRateAmount></previousKeySpecificRateAmount> 6
    <totalSpecificRateInvoiced></totalSpecificRateInvoiced> 8
```

"IMPORT" XML Invoice Details Before Editing

```
<invoiceDetails>
  <totalInvoiceAmount></totalInvoiceAmount>
  <specificRateLineItem>
    <paymentTypeKey>Specific Rate</paymentTypeKey>
    <DBEKey>N</DBEKey>
    <overheadRateTypeKey>Home Office</overheadRateTypeKey>
    <functionKey>Traffic Analysis Services</functionKey>
    <firstInvoiceForProject>Y</firstInvoiceForProject>
    <previousKeySpecificRateAmount>8654.61</previousKeySpecificRateAmount>
    <totalSpecificRateInvoiced></totalSpecificRateInvoiced>
```

"IMPORT" XML Invoice Details After Editing



Consultants: Complete Invoice Details - continued

4. Enter the contract function from the contract in CARS between the functionKey tags. This value must match the value in CARS exactly, including capitalization and spaces.
5. Change the value between the firstInvoiceForProject tags to “Y” or “N”.
6. If this is the first invoice in CARS and there are existing invoices outside CARS, enter the amount previously invoiced outside CARS between the previousKeySpecificRateAmount tags.

This value will affect the total amount you get paid for the contract. Be sure it is correct.

7. Optional: Enter a dollar value for the total invoice amount between the totalInvoiceAmount tags for your reference. This value is not required.
8. Optional: Enter a dollar value for the total specific rate amount between the totalSpecificRateInvoiced tags for your reference. This value is not required.

```

34 <invoiceDetails>
35 <totalInvoiceAmount></totalInvoiceAmount> 7
36 <specificRateLineItem>
37 <paymentTypeKey>Specific Rate</paymentTypeKey> 1
38 <DBEKey>Y or N</DBEKey> 2
39 <overheadRateTypeKey>MUST MATCH VALID TYPE IN CARS</overheadRateTypeKey> 3
40 <functionKey>MUST MATCH VALID TYPE IN CARS</functionKey> 4
41 <firstInvoiceForProject>Y or N</firstInvoiceForProject> 5
42 <previousKeySpecificRateAmount></previousKeySpecificRateAmount> 6
43 <totalSpecificRateInvoiced></totalSpecificRateInvoiced> 8
44

```

“IMPORT” XML Invoice Details Before Editing

```

33 <invoiceDetails>
34 <totalInvoiceAmount></totalInvoiceAmount>
35 <specificRateLineItem>
36 <paymentTypeKey>Specific Rate</paymentTypeKey>
37 <DBEKey>N</DBEKey>
38 <overheadRateTypeKey>Home Office</overheadRateTypeKey>
39 <functionKey>Traffic Analysis Services</functionKey>
40 <firstInvoiceForProject>Y</firstInvoiceForProject>
41 <previousKeySpecificRateAmount>8654.61</previousKeySpecificRateAmount>
42 <totalSpecificRateInvoiced></totalSpecificRateInvoiced>
43

```

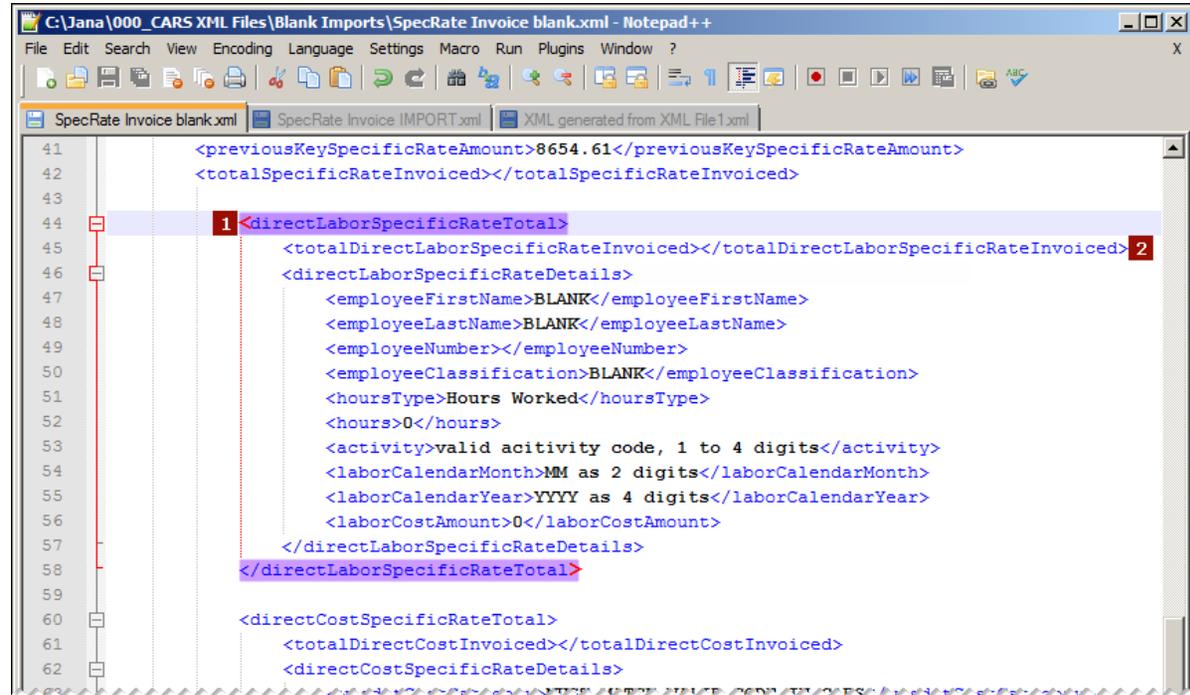
“IMPORT” XML Invoice Details After Editing



Consultants: Copy Tagged Data

The next step is to copy the tagged data generated from the Excel data into the XML invoice. For this example, the data from Excel is direct labor costs for a specific rate project. There is only one block for direct labor specific rate details in the IMPORT copy of the invoice. However, the CARS schema does not limit the number of direct labor specific rate details blocks. The same is true of direct cost specific rate blocks. Refer to the *CARS XML Detailed Element Descriptions* document for other blocks that may repeat in the invoice.

1. Locate the `directLaborSpecificRateTotal` block in the XML invoice you have been editing.
2. Optional: enter a value for the total direct labor cost for this invoice between the `totalDirectLaborSpecificRateInvoiced` tags for your reference. This value is not required.

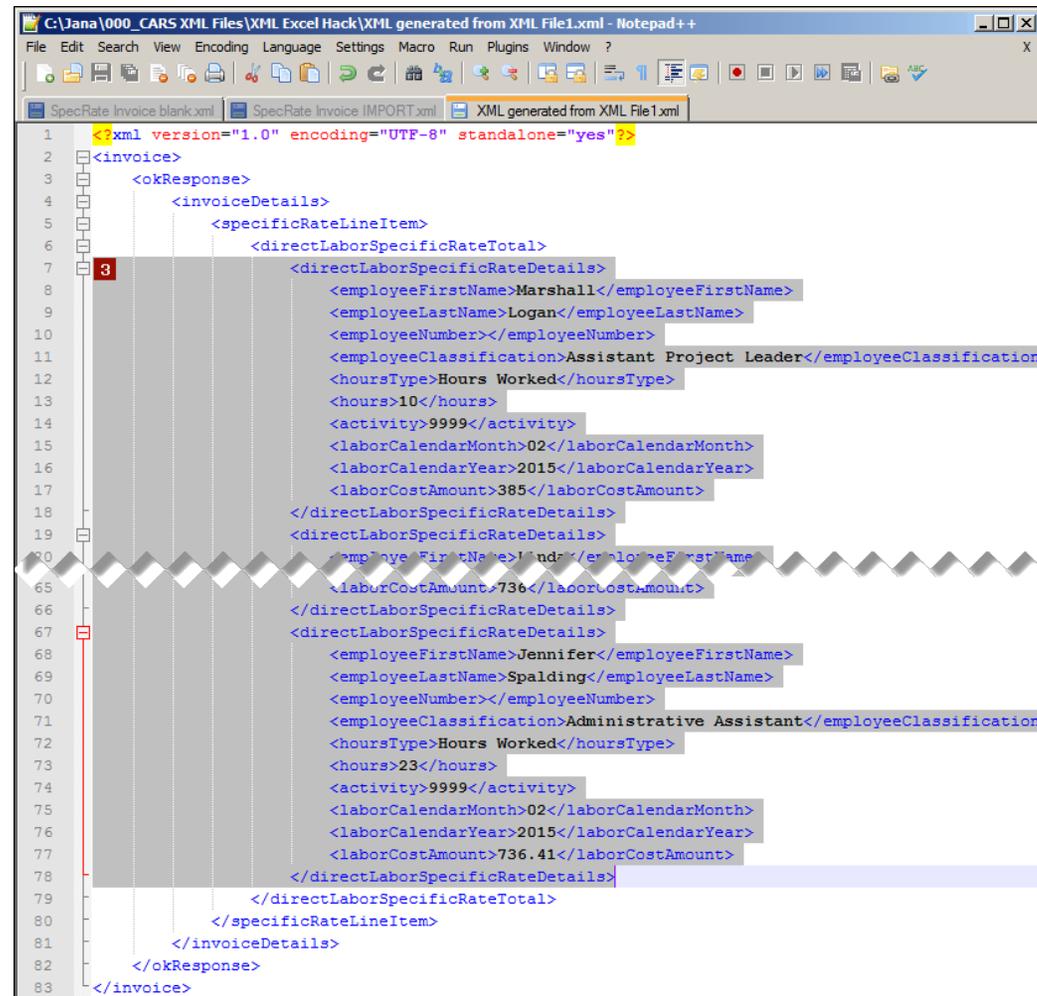


```
41 <previousKeySpecificRateAmount>8654.61</previousKeySpecificRateAmount>
42 <totalSpecificRateInvoiced></totalSpecificRateInvoiced>
43
44 1 <directLaborSpecificRateTotal>
45   <totalDirectLaborSpecificRateInvoiced></totalDirectLaborSpecificRateInvoiced> 2
46   <directLaborSpecificRateDetails>
47     <employeeFirstName>BLANK</employeeFirstName>
48     <employeeLastName>BLANK</employeeLastName>
49     <employeeNumber></employeeNumber>
50     <employeeClassification>BLANK</employeeClassification>
51     <hoursType>Hours Worked</hoursType>
52     <hours>0</hours>
53     <activity>valid activity code, 1 to 4 digits</activity>
54     <laborCalendarMonth>MM as 2 digits</laborCalendarMonth>
55     <laborCalendarYear>YYYY as 4 digits</laborCalendarYear>
56     <laborCostAmount>0</laborCostAmount>
57   </directLaborSpecificRateDetails>
58 </directLaborSpecificRateTotal>
59
60 <directCostSpecificRateTotal>
61   <totalDirectCostInvoiced></totalDirectCostInvoiced>
62   <directCostSpecificRateDetails>
```



Consultants: Copy Tagged Data - continued

3. Locate and select the directLaborSpecificRateDetails blocks in the XML generated from Excel. Be sure to select the start and end tags. Do not select any other tags or information.
4. Copy the selected data.

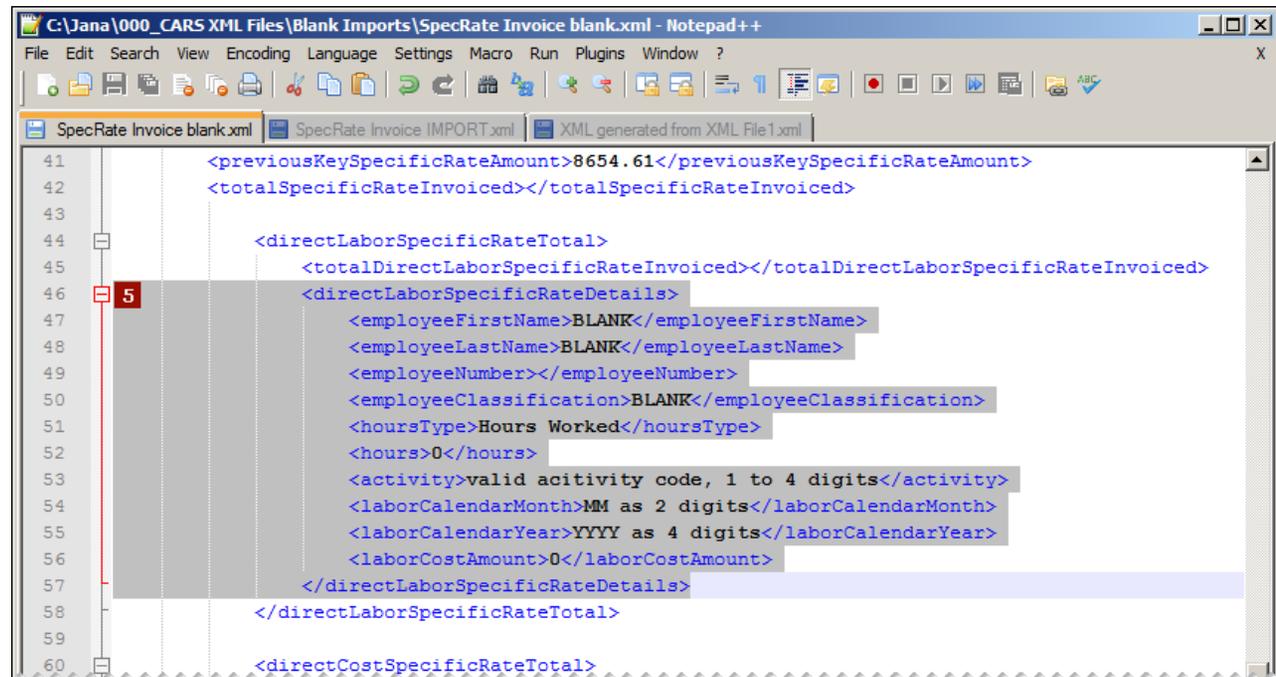


```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<invoice>
  <okResponse>
    <invoiceDetails>
      <specificRateLineItem>
        <directLaborSpecificRateTotal>
          <directLaborSpecificRateDetails>
            <employeeFirstName>Marshall</employeeFirstName>
            <employeeLastName>Logan</employeeLastName>
            <employeeNumber></employeeNumber>
            <employeeClassification>Assistant Project Leader</employeeClassification>
            <hoursType>Hours Worked</hoursType>
            <hours>10</hours>
            <activity>9999</activity>
            <laborCalendarMonth>02</laborCalendarMonth>
            <laborCalendarYear>2015</laborCalendarYear>
            <laborCostAmount>385</laborCostAmount>
          </directLaborSpecificRateDetails>
        </directLaborSpecificRateTotal>
      </specificRateLineItem>
    </invoiceDetails>
  </okResponse>
</invoice>
```



Consultants: Copy Tagged Data - continued

5. Locate and select the `directLaborSpecificRateDetails` block in the XML invoice you have been editing. Be sure to select the start and end tags. Do not select any other tags or information.



The screenshot shows a Notepad++ window with the following XML code. The `<directLaborSpecificRateDetails>` block is highlighted in blue, and a red box with the number '5' is placed over the start tag on line 46.

```
41 <previousKeySpecificRateAmount>8654.61</previousKeySpecificRateAmount>
42 <totalSpecificRateInvoiced></totalSpecificRateInvoiced>
43
44 <directLaborSpecificRateTotal>
45 <totalDirectLaborSpecificRateInvoiced></totalDirectLaborSpecificRateInvoiced>
46 <directLaborSpecificRateDetails>
47 <employeeFirstName>BLANK</employeeFirstName>
48 <employeeLastName>BLANK</employeeLastName>
49 <employeeNumber></employeeNumber>
50 <employeeClassification>BLANK</employeeClassification>
51 <hoursType>Hours Worked</hoursType>
52 <hours>0</hours>
53 <activity>valid acitivity code, 1 to 4 digits</activity>
54 <laborCalendarMonth>MM as 2 digits</laborCalendarMonth>
55 <laborCalendarYear>YYYY as 4 digits</laborCalendarYear>
56 <laborCostAmount>0</laborCostAmount>
57 </directLaborSpecificRateDetails>
58 </directLaborSpecificRateTotal>
59
60 <directCostSpecificRateTotal>
```

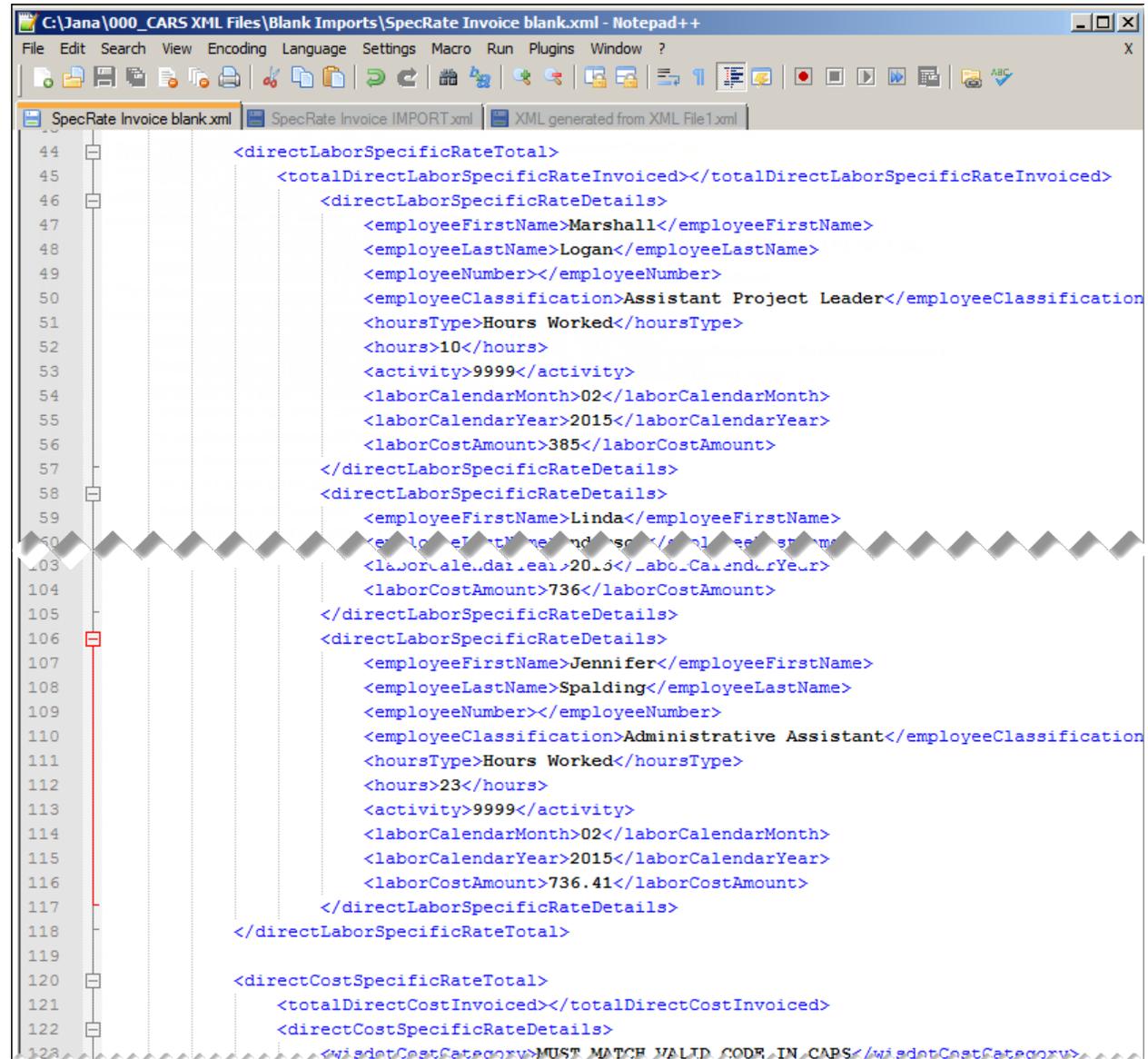


Consultants: Copy Tagged Data - continued

- Paste the data from the XML generated from Excel.

Note: Spacing between lines and block alignment is not critical for XML invoices. The line spacing and alignment is ignored when the data is imported. However, formatting is very helpful when you read and edit the XML document.

- Continue to edit the invoice. Edit other sections and paste in other details sections that were exported from Excel as needed.
- Import the XML invoice into CARS. Review the invoice in CARS for accuracy, and edit it manually as necessary. Remember to upload supporting documents, and submit the invoice as described in the CARS training manual.



```
C:\Jana\000_CARS XML Files\Blank Imports\SpecRate Invoice blank.xml - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
SpecRate Invoice blank.xml SpecRate Invoice IMPORT.xml XML generated from XML File1.xml
44 <directLaborSpecificRateTotal>
45 <totalDirectLaborSpecificRateInvoiced></totalDirectLaborSpecificRateInvoiced>
46 <directLaborSpecificRateDetails>
47 <employeeFirstName>Marshall</employeeFirstName>
48 <employeeLastName>Logan</employeeLastName>
49 <employeeNumber></employeeNumber>
50 <employeeClassification>Assistant Project Leader</employeeClassification>
51 <hoursType>Hours Worked</hoursType>
52 <hours>10</hours>
53 <activity>9999</activity>
54 <laborCalendarMonth>02</laborCalendarMonth>
55 <laborCalendarYear>2015</laborCalendarYear>
56 <laborCostAmount>385</laborCostAmount>
57 </directLaborSpecificRateDetails>
58 <directLaborSpecificRateDetails>
59 <employeeFirstName>Linda</employeeFirstName>
60 <employeeLastName> </employeeLastName>
61 <laborCalendarYear>2015</laborCalendarYear>
62 <laborCostAmount>736</laborCostAmount>
63 </directLaborSpecificRateDetails>
64 <directLaborSpecificRateDetails>
65 <employeeFirstName>Jennifer</employeeFirstName>
66 <employeeLastName>Spalding</employeeLastName>
67 <employeeNumber></employeeNumber>
68 <employeeClassification>Administrative Assistant</employeeClassification>
69 <hoursType>Hours Worked</hoursType>
70 <hours>23</hours>
71 <activity>9999</activity>
72 <laborCalendarMonth>02</laborCalendarMonth>
73 <laborCalendarYear>2015</laborCalendarYear>
74 <laborCostAmount>736.41</laborCostAmount>
75 </directLaborSpecificRateDetails>
76 </directLaborSpecificRateTotal>
77
78 <directCostSpecificRateTotal>
79 <totalDirectCostInvoiced></totalDirectCostInvoiced>
80 <directCostSpecificRateDetails>
81 <directCostCategory>MUST MATCH VALID CODE IN CARS</directCostCategory>
```

