2: Set Tag Tutorial

Prerequisite Tutorials

Before doing this tutorial, we recommend you have done...

1. [The Start Here Tutorial](#)

Training Overview

This Training Guide walks you through the fundamentals to understand and use the Set tag.

1. What does a Set tag do?

2. How to use a Set tag

3. Designing a sample template with a Set tag

1 - What does a Set tag do?

A Set tag allows a user to store information inside of there template in the tag's variable. There is plenty of information you may want to store inside your template. Sometimes you may want to store a template specific value that is always the same so you can use the value in queries in a template. Sometimes you may want to store information that you get out of the database during report generation. Other times, you can use template variables to record some sort of state the template is in. No matter how you use it, Set tags are a useful tool for saving information in your template under a variable name so you can access it again later on.

2 - How to use a Set tag

Set tags can be placed in a document from the Tags drop down, just like every other tag.
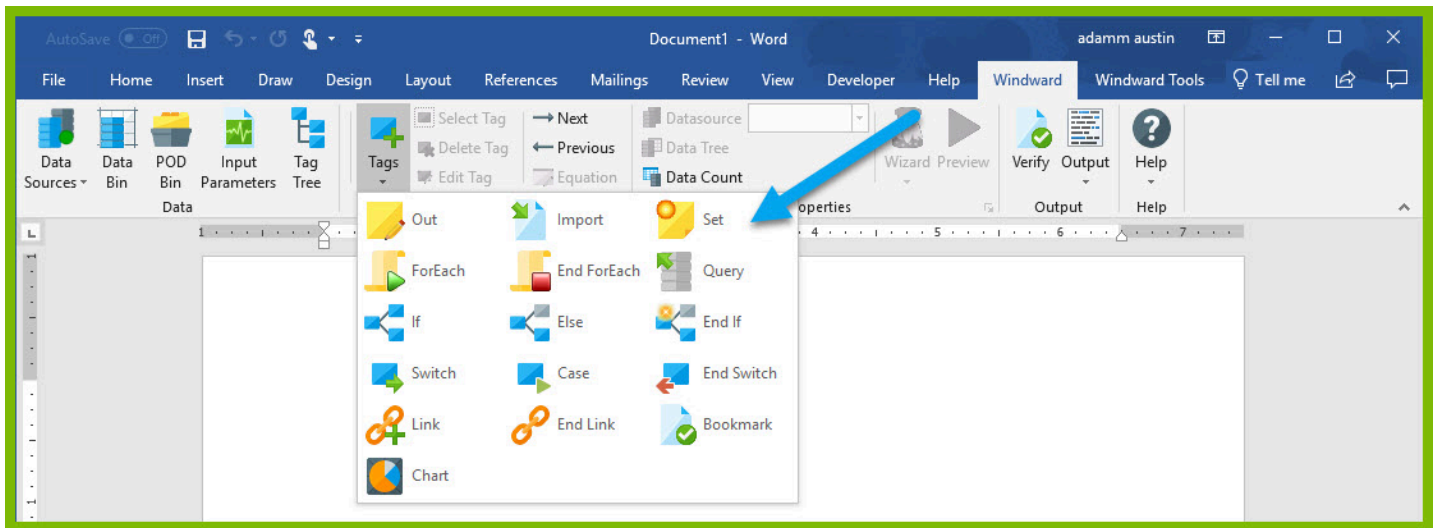
Once in the template, the Set tag can be edited with one of our many tools: the Tag Editor, Wizard, and Tag Tree are all good options.
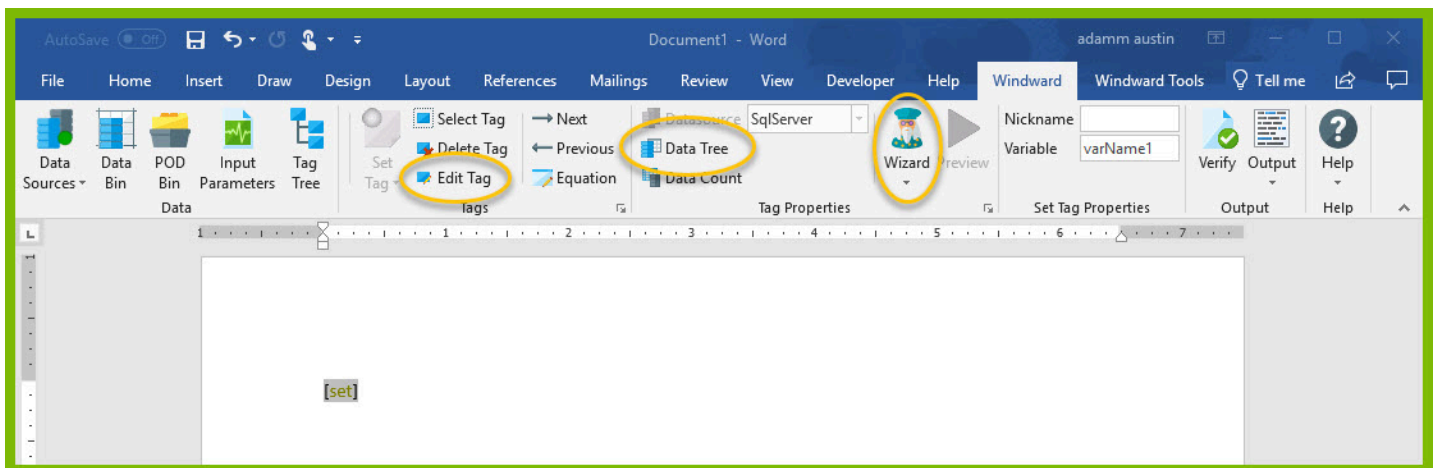

Inside the Tag Editor, we can explicitly set a value for the Set tag by entering it into the query pane of the Tag editor and selecting the "Value" option for the tag. Previewing the tag will show that the value is saved to the tag variable.
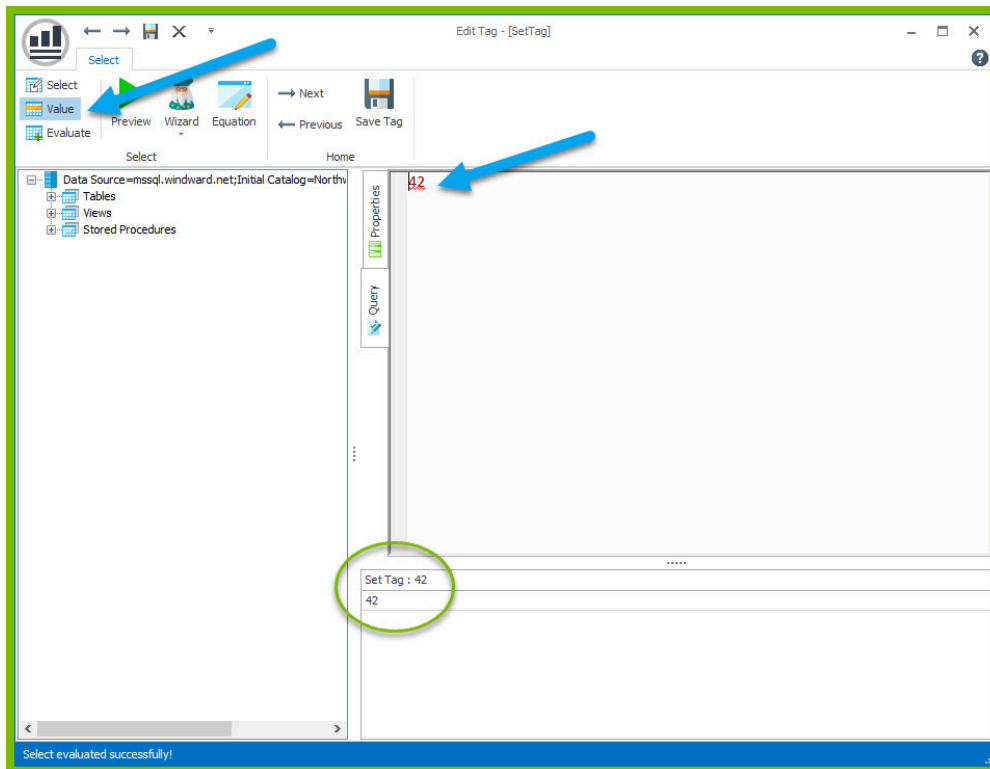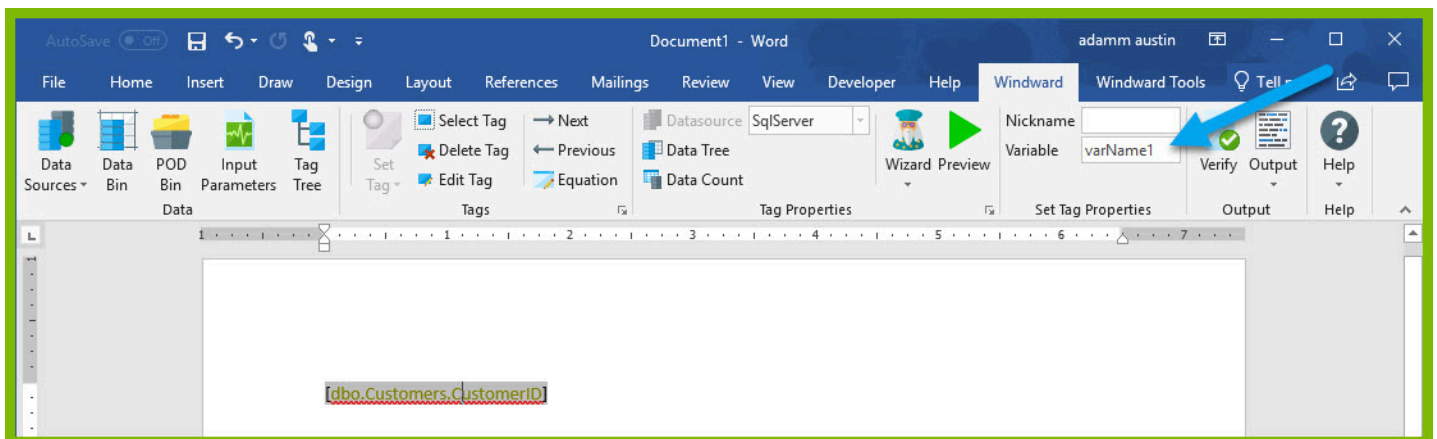
We can also use the Wizard to query specific information from our data, and store the result in the tag variable.
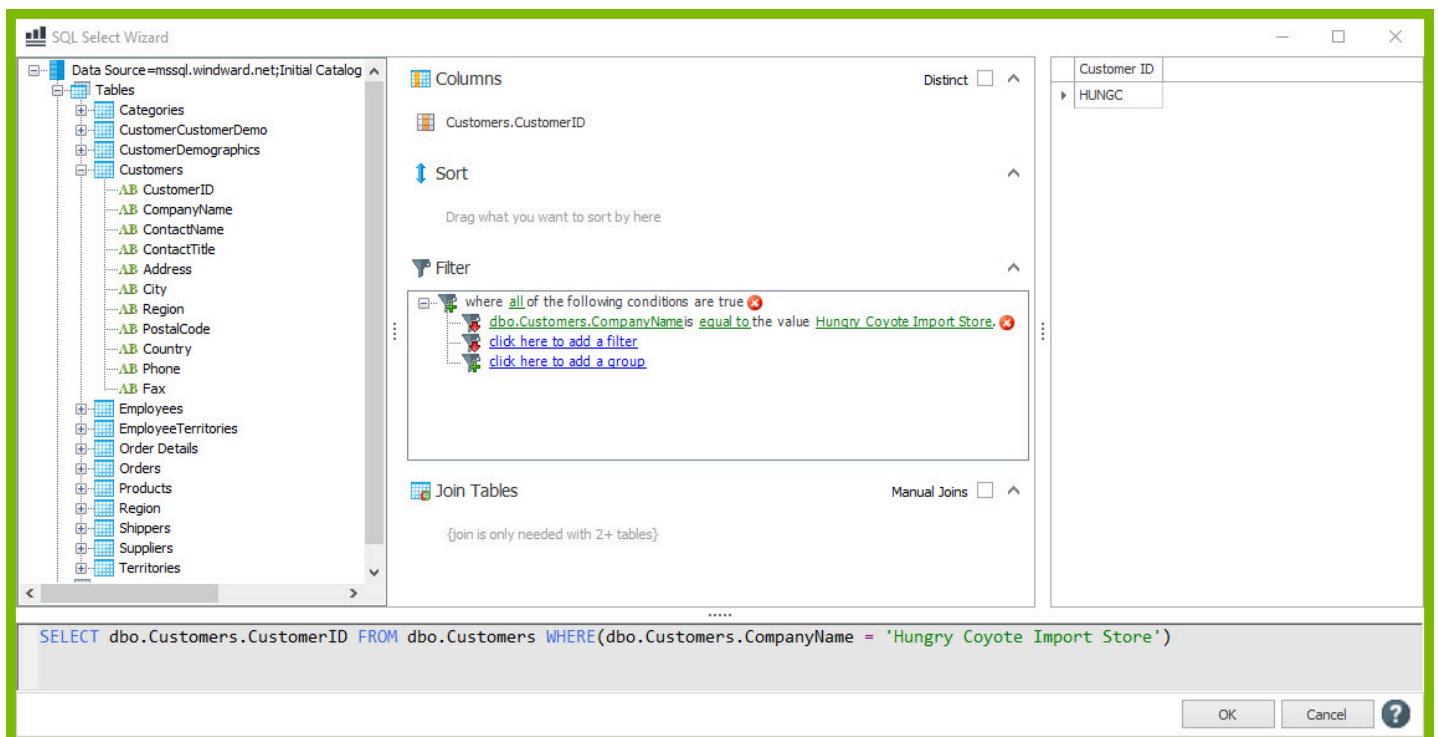
Setting the tag to Select and previewing the tag will show the queried value being saved to the select variable.
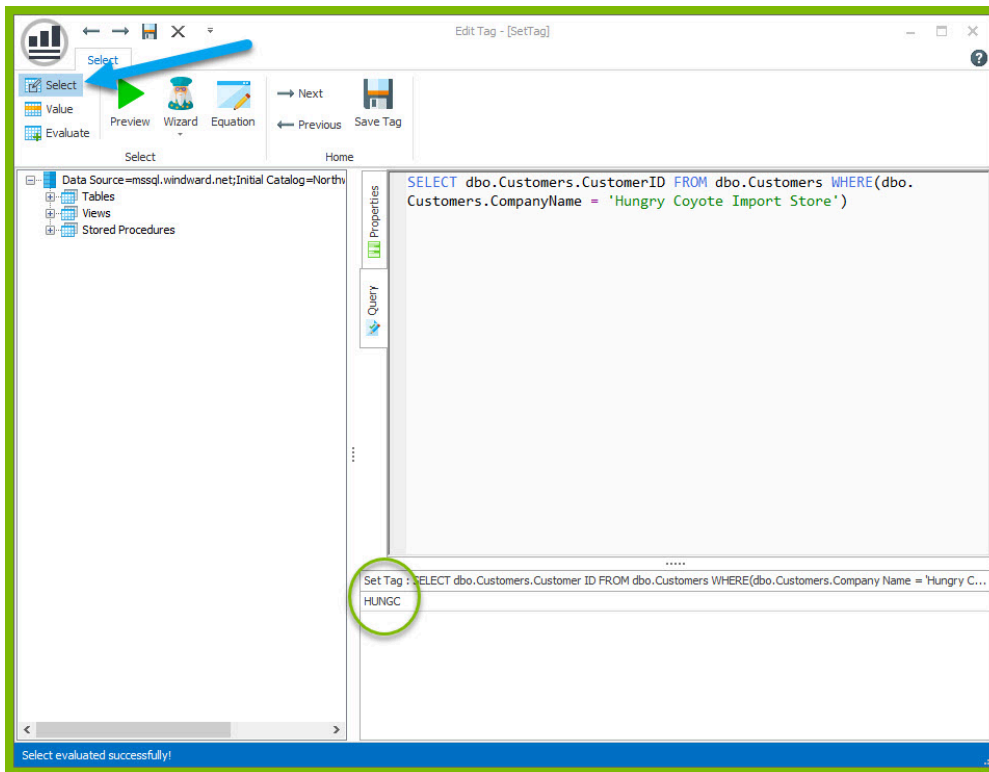
The variable name for the tag can be changed to make the variable more easily recognizable in the template later.


3 - Designing a sample template with a Set tag

To demonstrate use of the Set tag, you will make a basic template that calculates a total cost of orders by a company.

1. Open Microsoft Word. Now that you've installed Windward Designer, you will see two new tabs in the Microsoft Office Ribbon – the "**Windward**" Tab and the "**Windward Tools**" Tab.
2. Click on the "Windward" Tab and click on the "**Data Sources**" button located on the left-hand side of the ribbon. This opens the Connection Editor where you will manage all of your data connections.
3. In the Connection Editor window, select the data connection labeled "**SqlServer**" under "Recent Inactive" and click the "**Connect**" button.
4. Close the Connection Editor by clicking "Close." You have now successfully connected to the sample data!

Now you will start tagging the template. The goal of this template is, for a single CustomerID, to output every order by that customer, some details about the order, and total amount for all orders from the customer.

5. Place your cursor on the first line of the document, click Tags to open the Tag drop down, and select the Set tag.
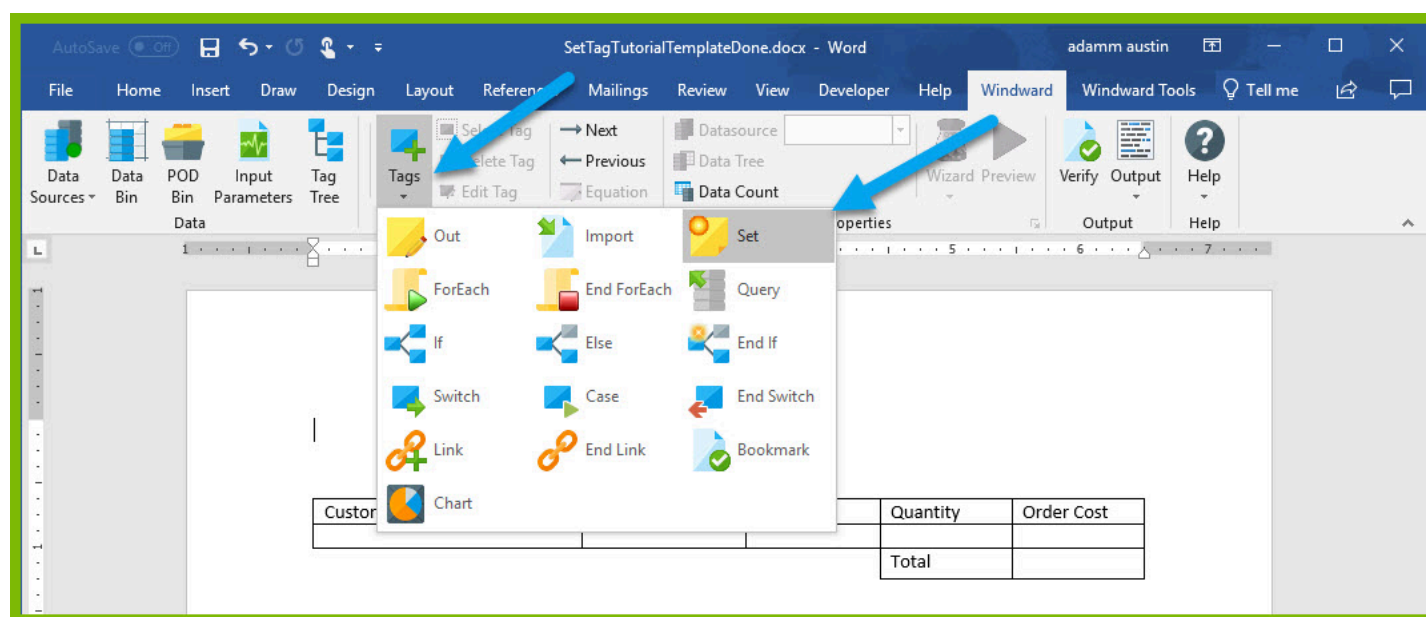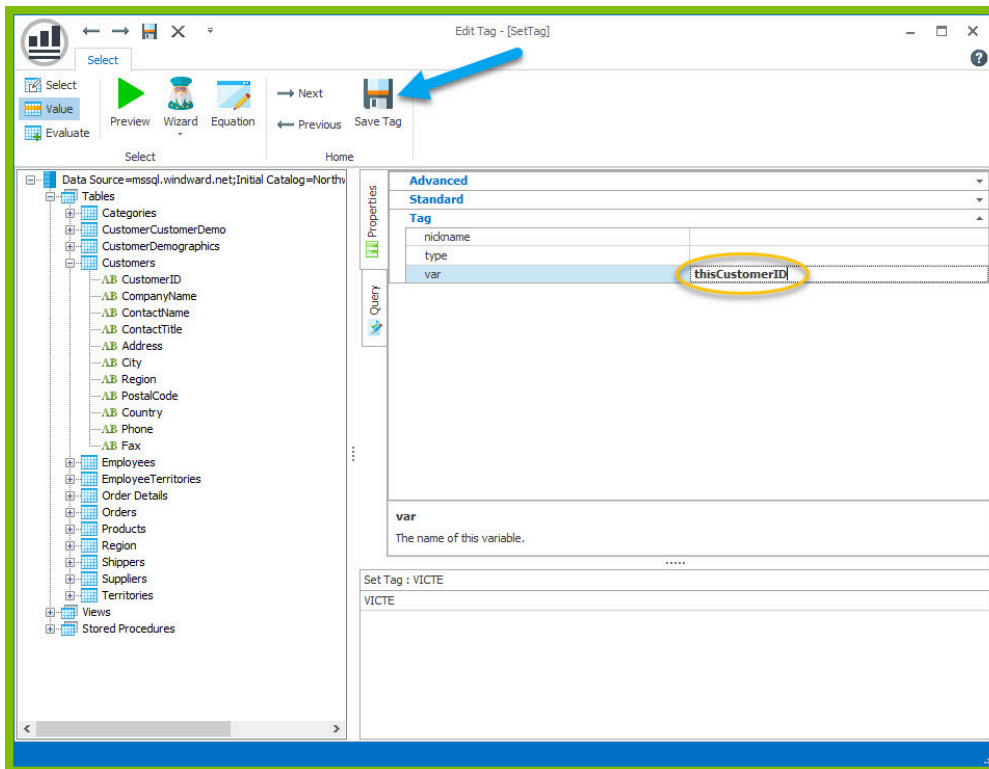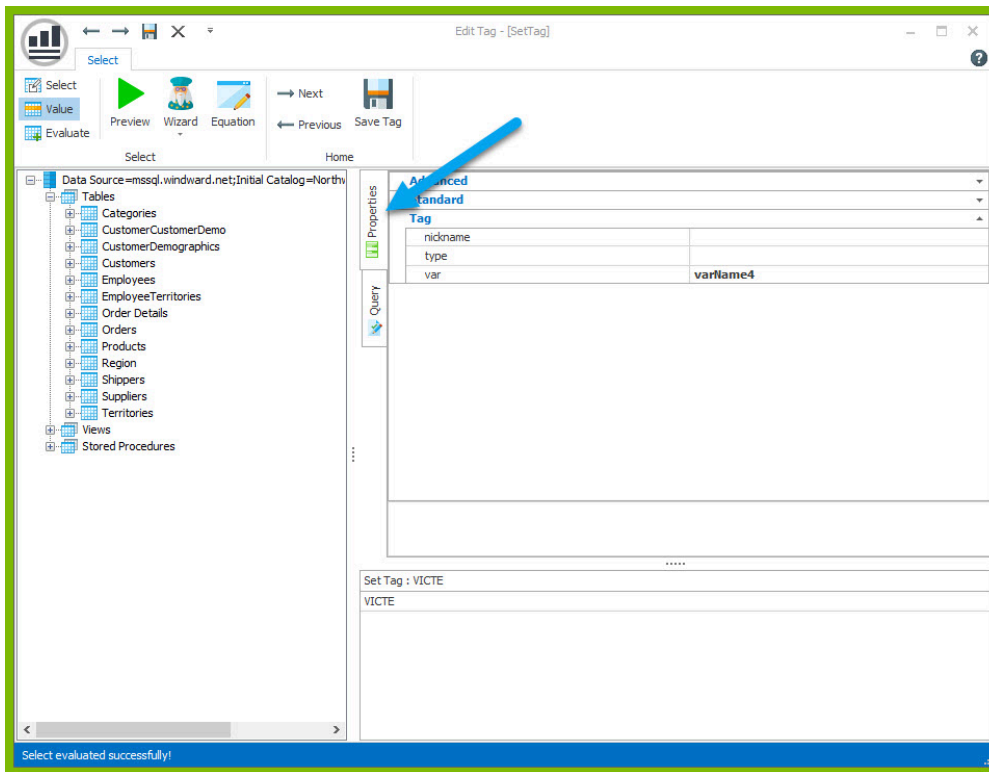

6. Place your cursor on the Set tag in the template and open the Tag Editor by clicking the Edit Tag button.
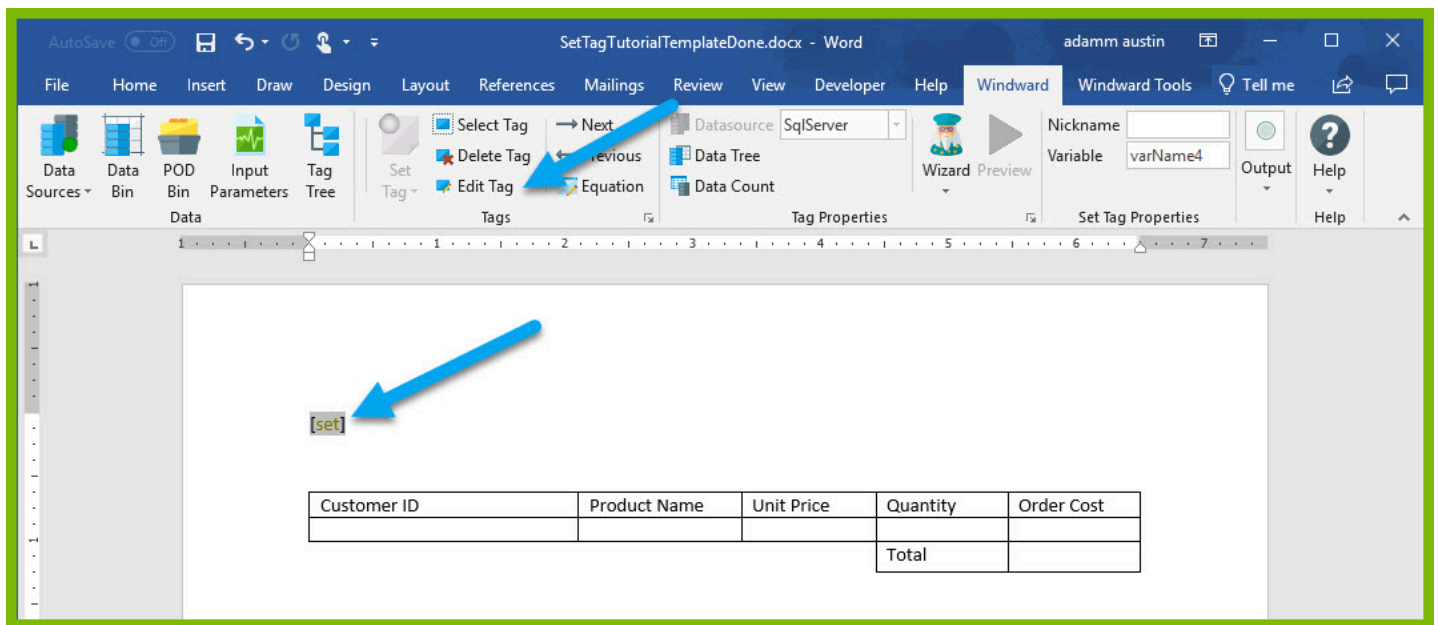
7. In the query pane of the Tag Editor pane, enter the value "VICTE" (no quotes) and click Preview. This value (one of the CustomerIDs in the Customers table) will be used later on to filter our results. If the preview shows the correct value, the tag is set correctly. If the preview is incorrect, or gives an error, make sure you have Value set for that tag in the upper left hand of the Tag Editor.
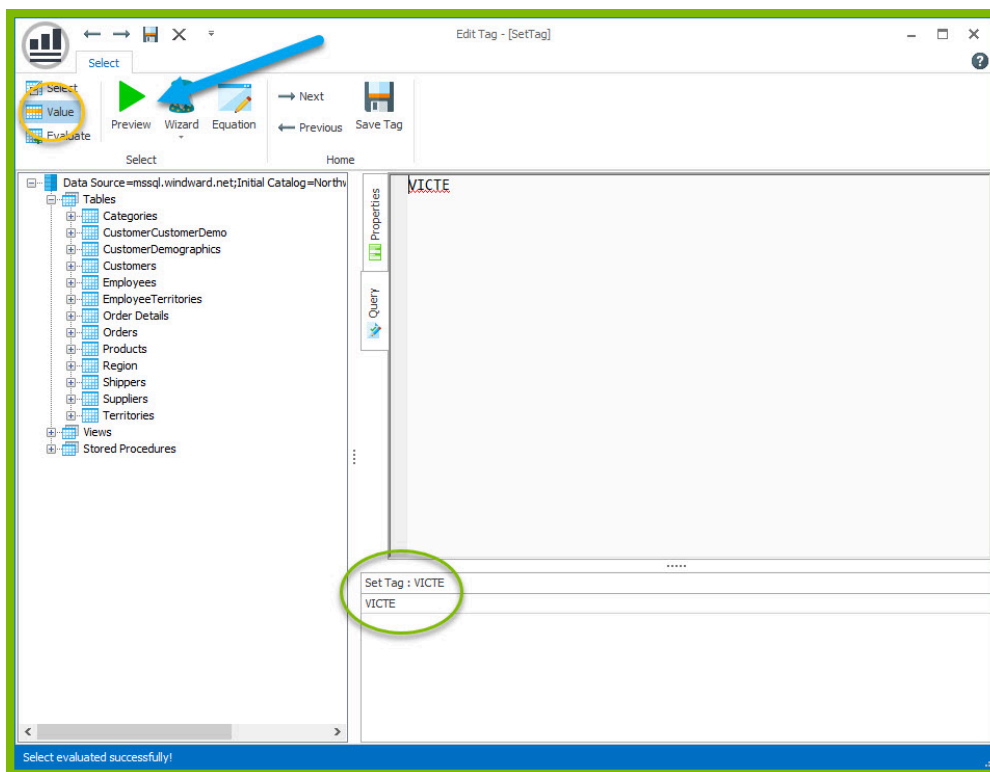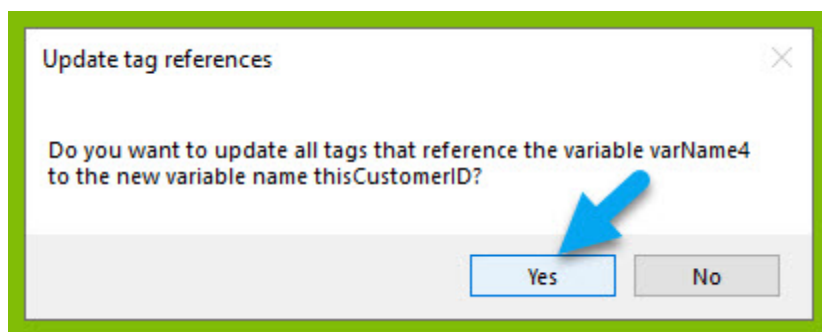

8. To set a variable name for the tag, select the Properties pane in the Tag Editor, and change the var value to "thisCustomerID". Click the Save Tag button.

9. In the "Update tag references" dialog, click Yes.
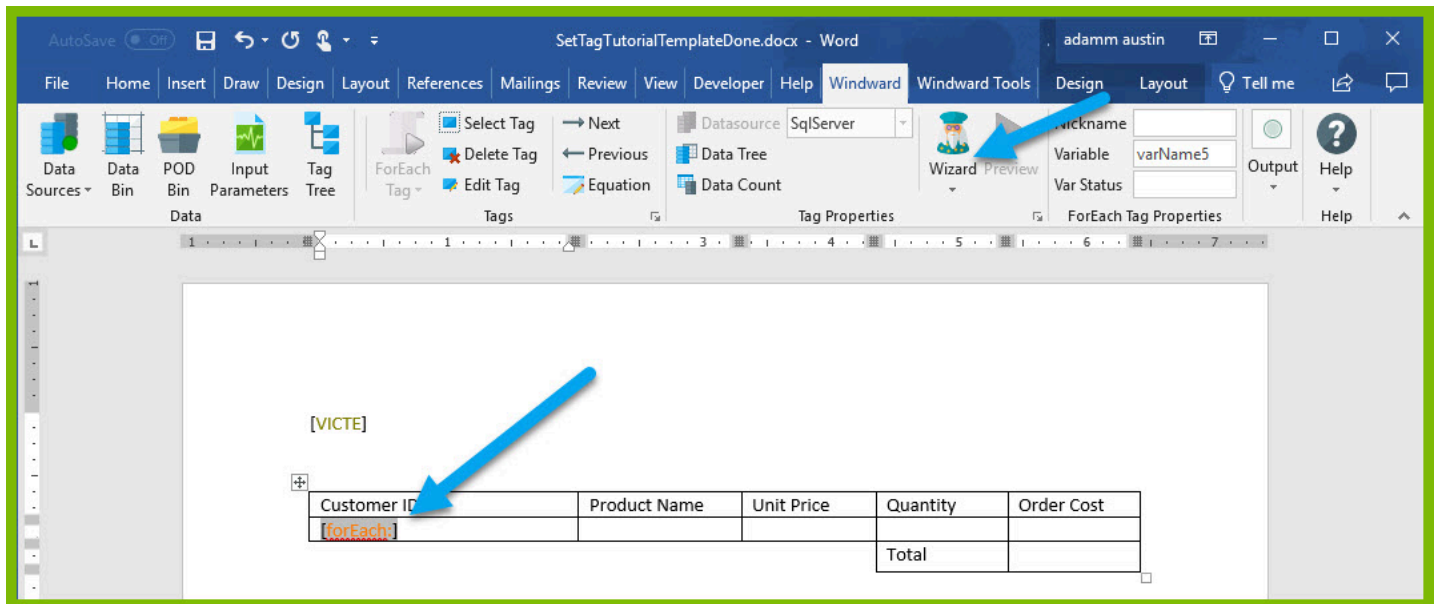
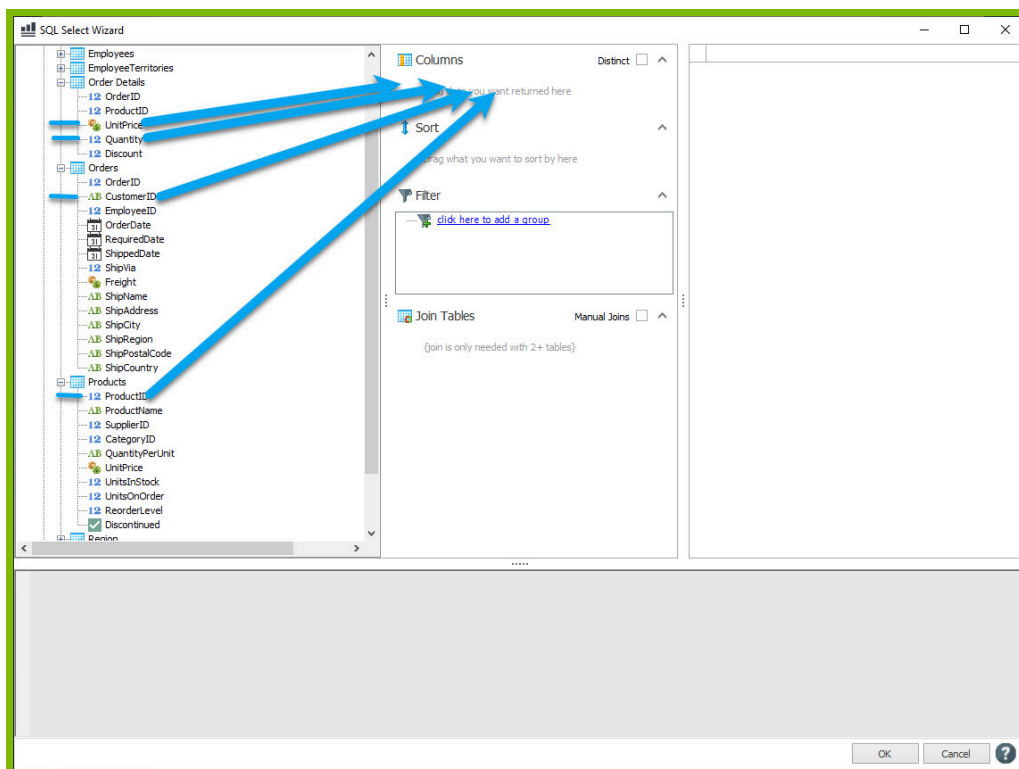10. In the cell under "Customer ID," insert a ForEach tag.
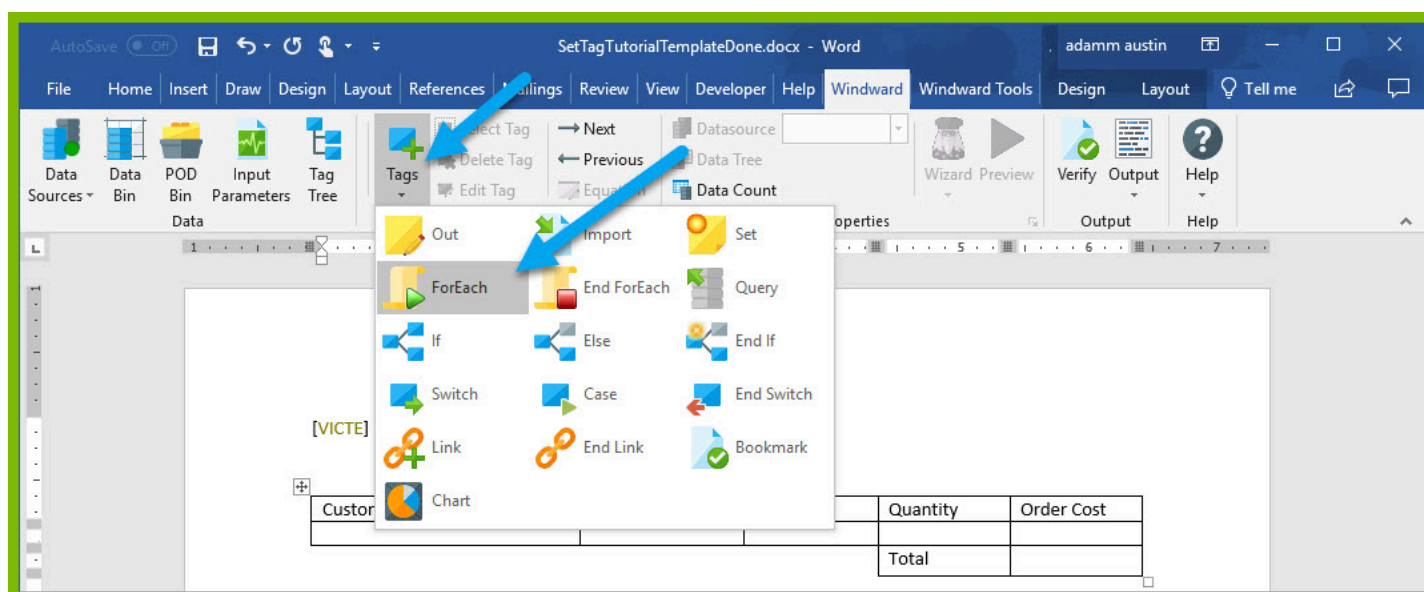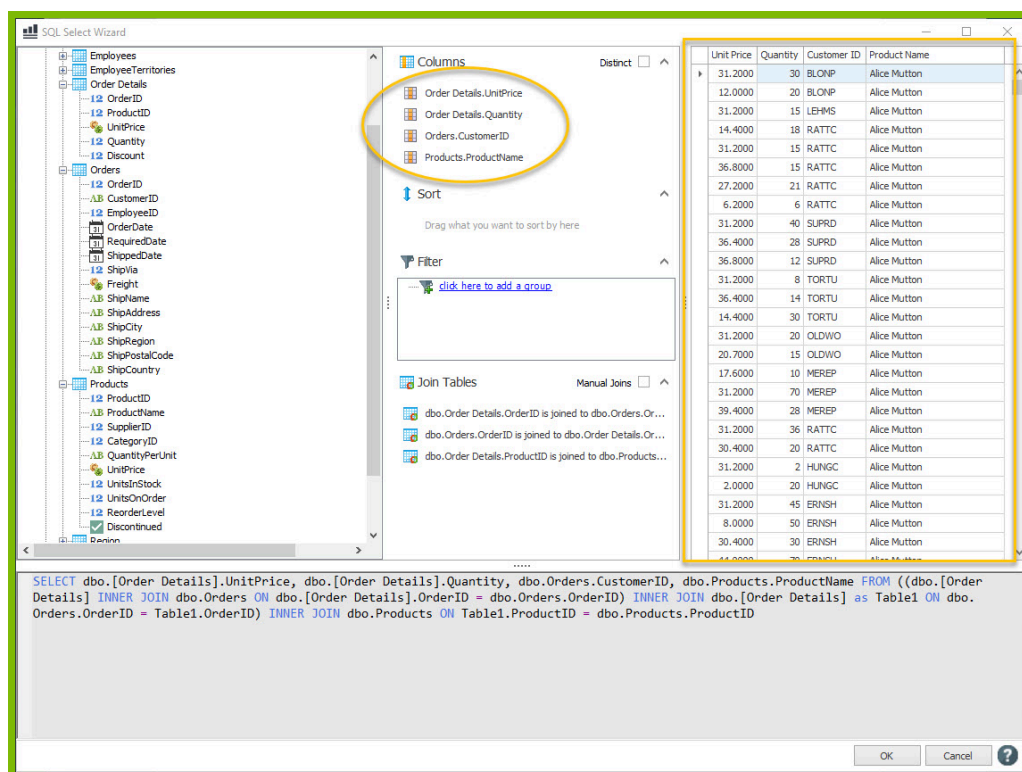

11. Place your cursor on the ForEach tag in the template and click the Wizard button to open the SQL Wizard.

12. In the Wizard, drag the 4 columns that you need from the Data Tree into the Columns section. These are the OrderDetails → UnitPrice, OrderDetails → Quantity, Orders → CustomerID, Products → ProductName

NOTE: We are able to drag columns in from three different tables (Order Details, Orders, and Products) thanks to a structure called a "join" in the SQL query language. While the SQL Wizard takes care of performing the join for you, using data from different tables like this is only possible when the involved tables can be related using a shared column. Here, the Order Details table has a column called OrderID which is the same as the OrderID in the Orders table, and the Order Details table has a column called ProductID which is the same as the ProductID in the Products table.

13. For this template, you only want to see orders for the CustomerID defined earlier. To add a filter, in the Filter section click "click here to add a group" and "click here to add a filter."
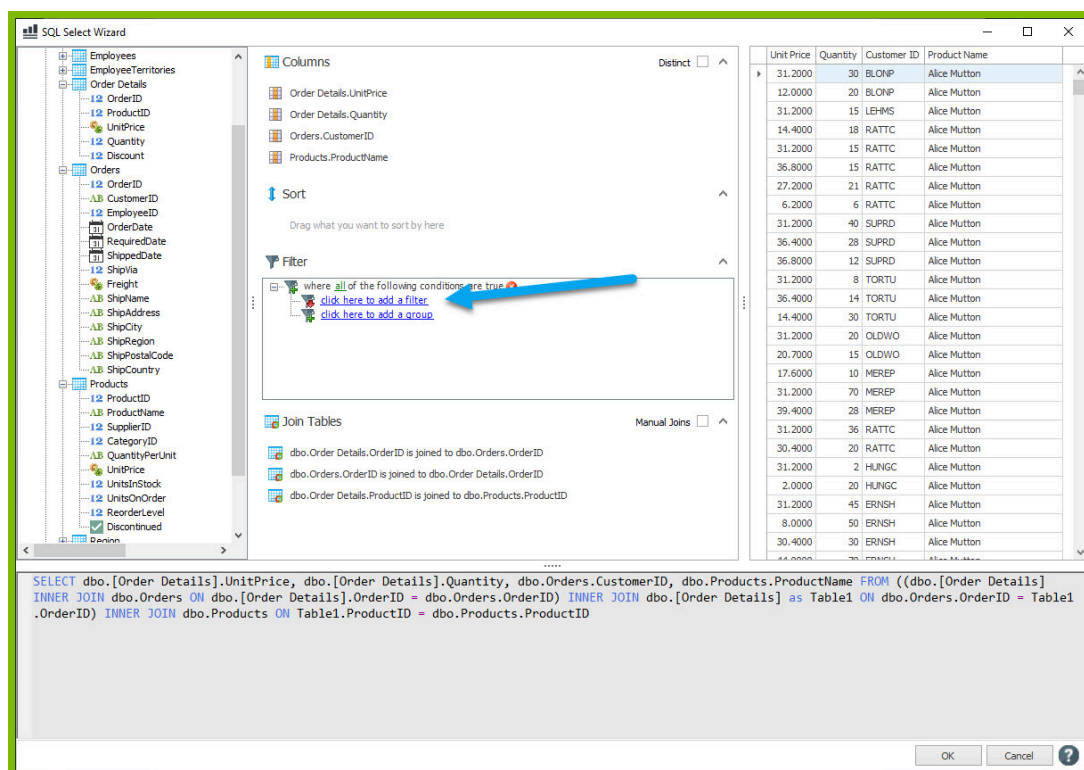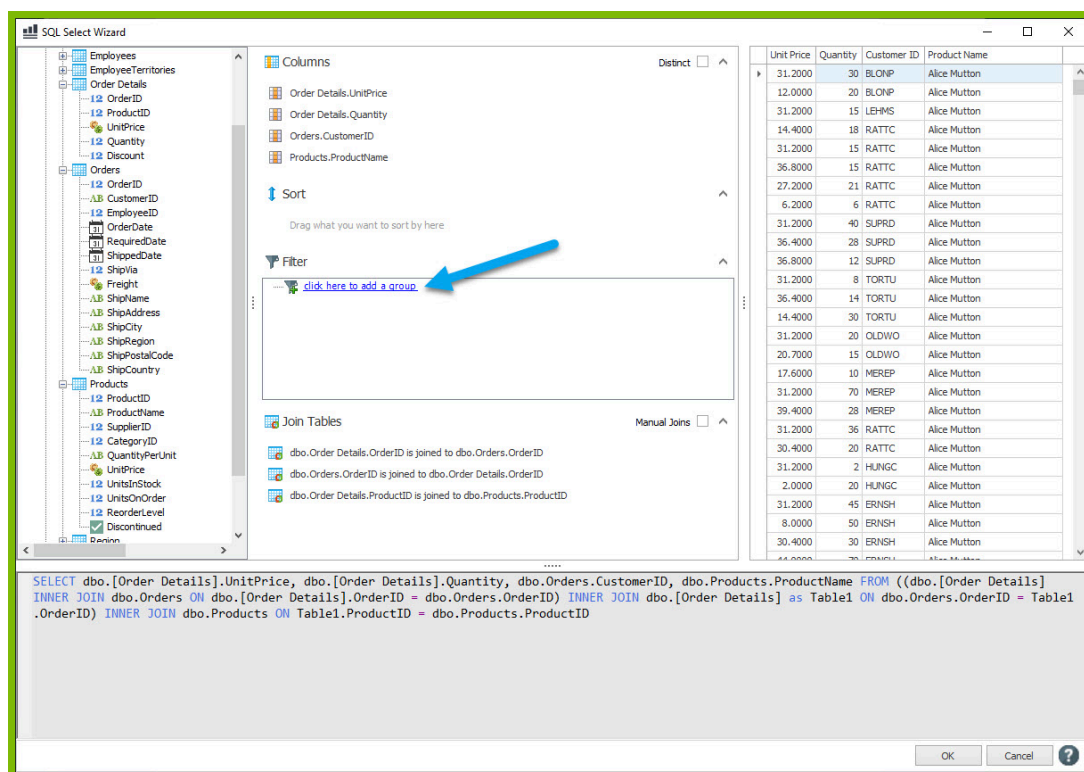
14. Click on "click here to select a node" and from the "Select the desired column" dialog, select the Orders → CustomerID node since you can compare this to the "thisCustomerID" variable defined earlier.
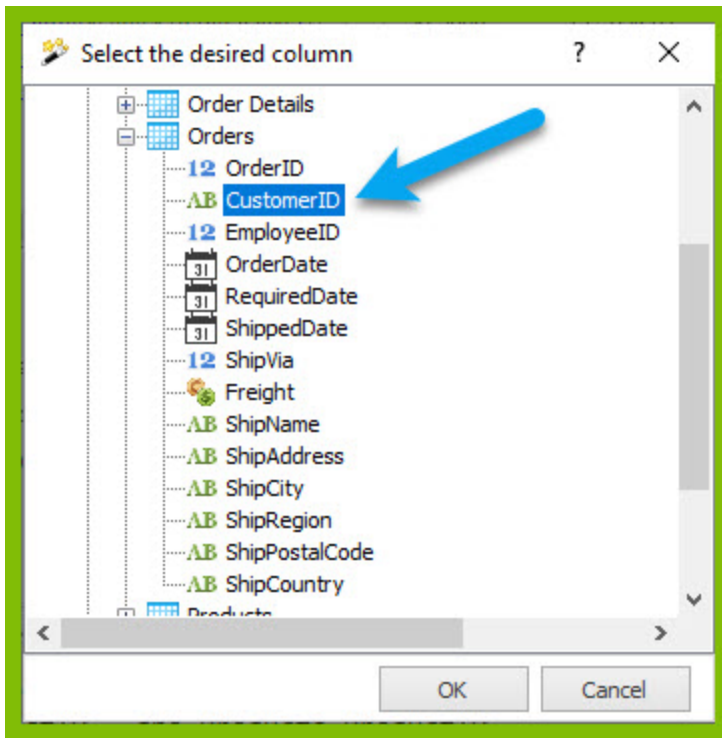
15. To compare the CustomerID column to the "thisCustomerID" variable, click "click here to set the value," click the down arrow to the side of the text field, and from the drop down of template variables, select "thisCustomerID."
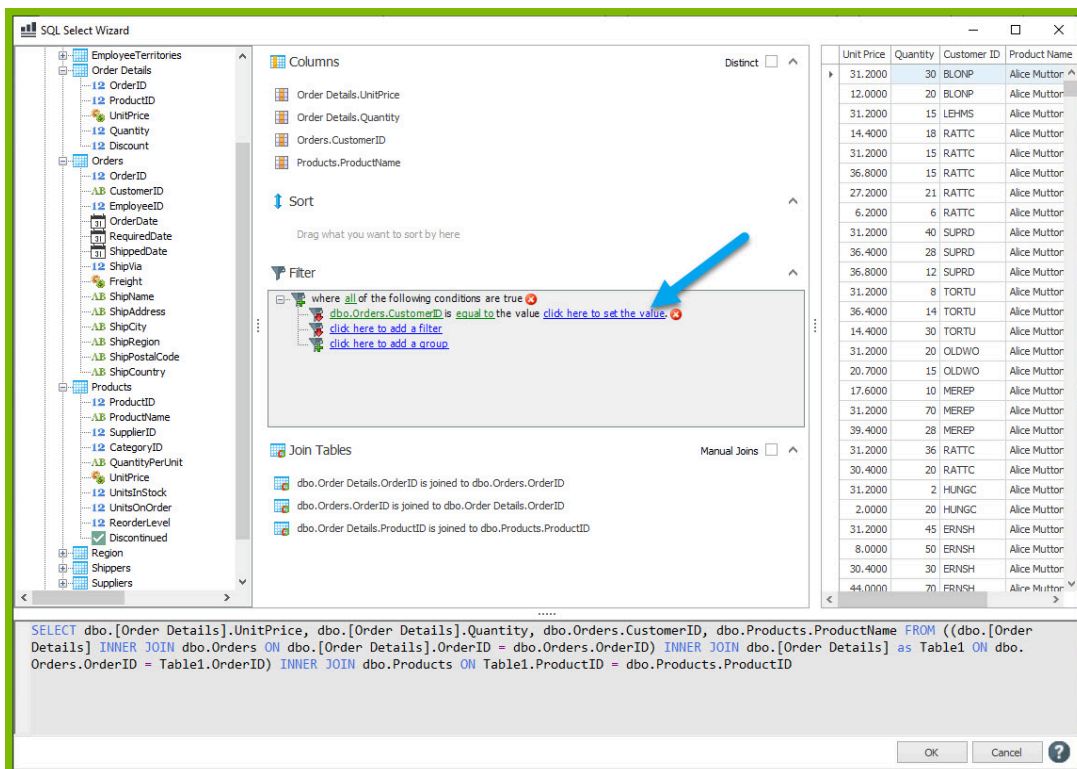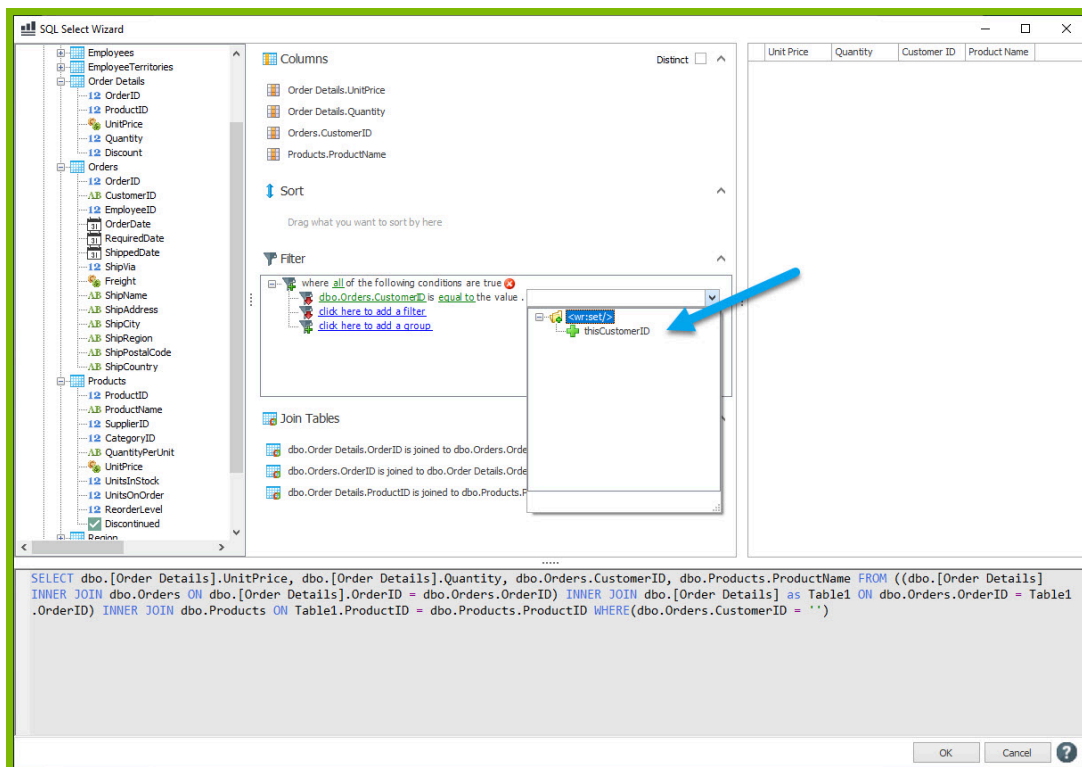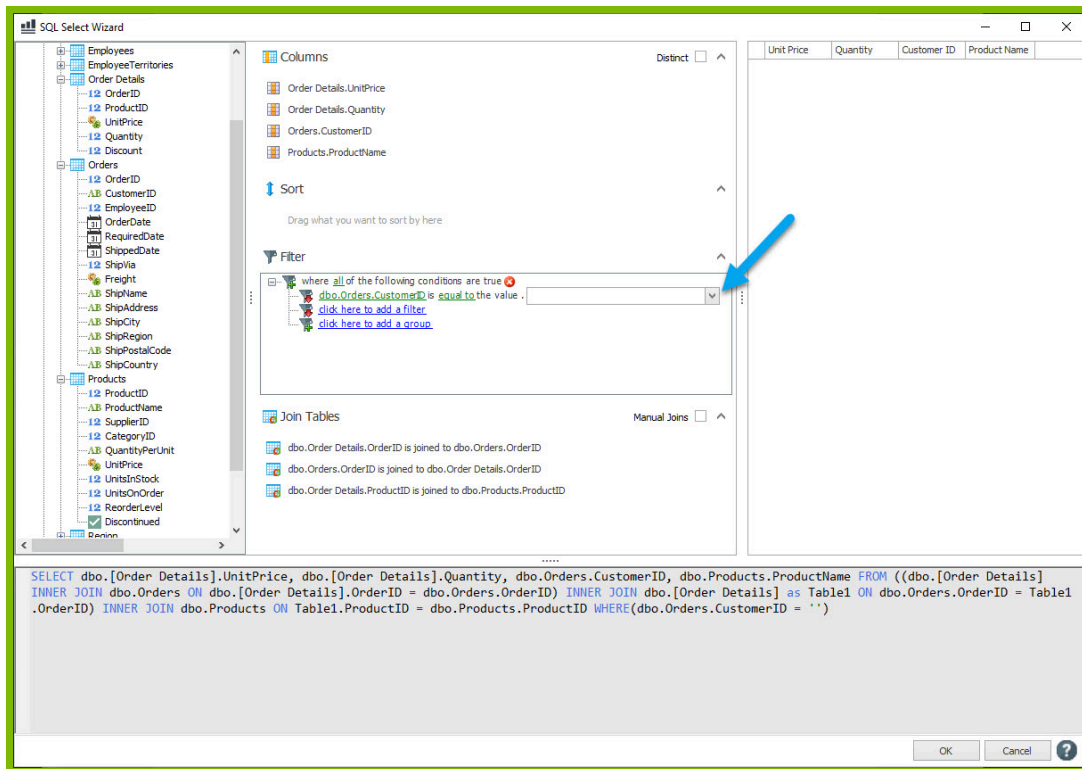
16. In the template, place your cursor on the ForEach tag and click Preview. You should see each of the 4 columns you selected in the preview, and you should only see rows that match the CustomerID "VICTE".

17. Next, add an out tag in the first column after our ForEach tag placing your cursor after the ForEach tag and clicking Tags → Out.

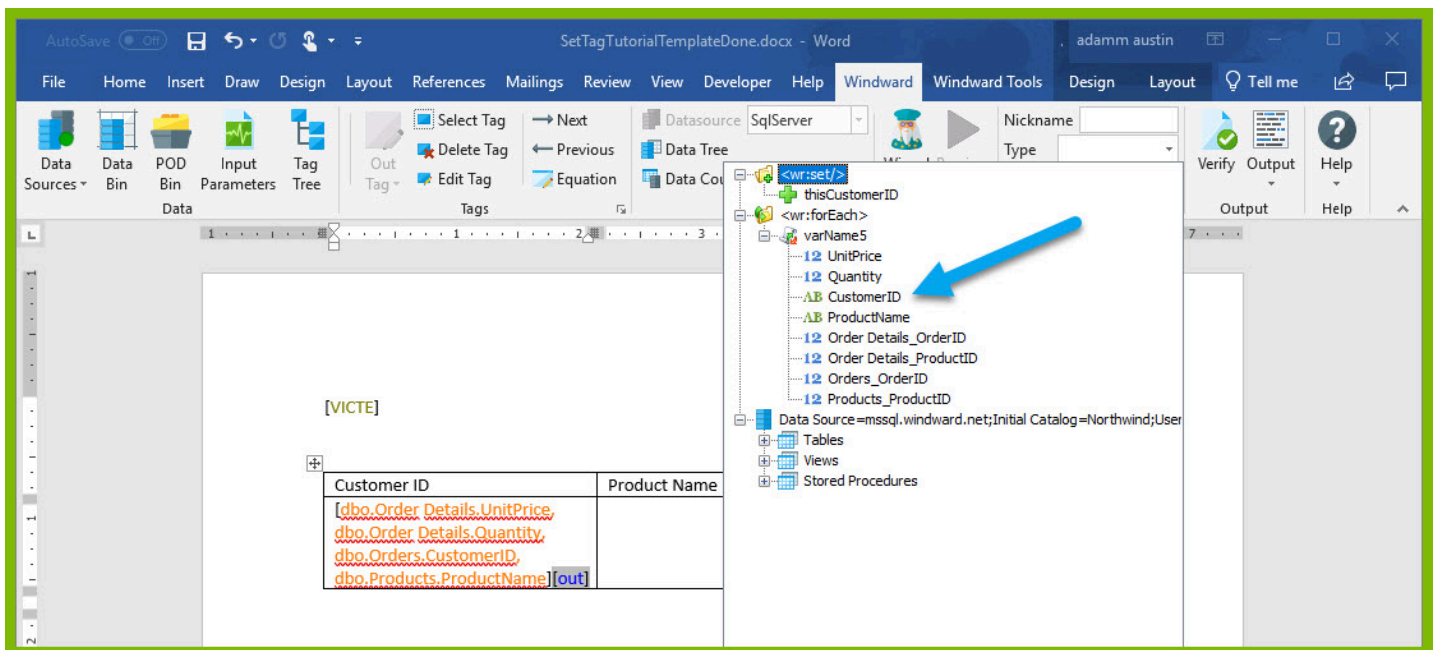18. Place your cursor on the Out tag, and open the Data Tree from the Tag Properties section. The Data Tree is a quick way to assign a select to a tag in your template.
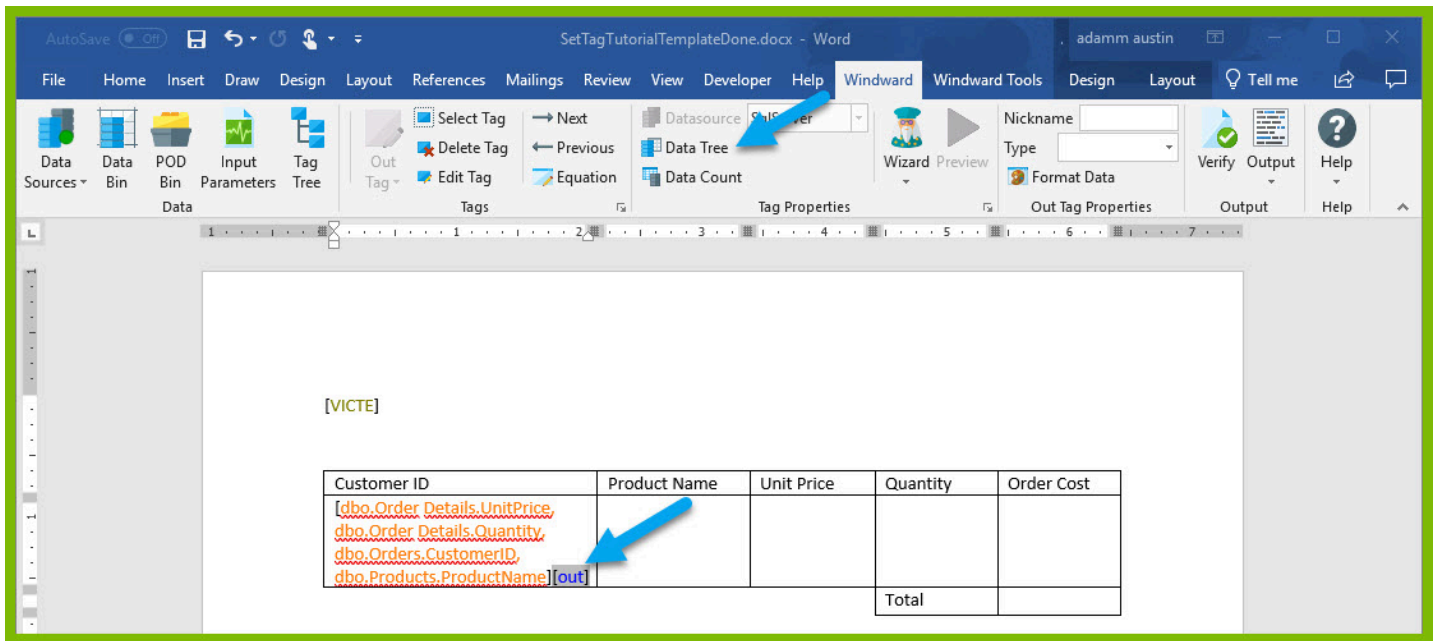
19. From the Data Tree, select the node CustomerID under your ForEach variable.
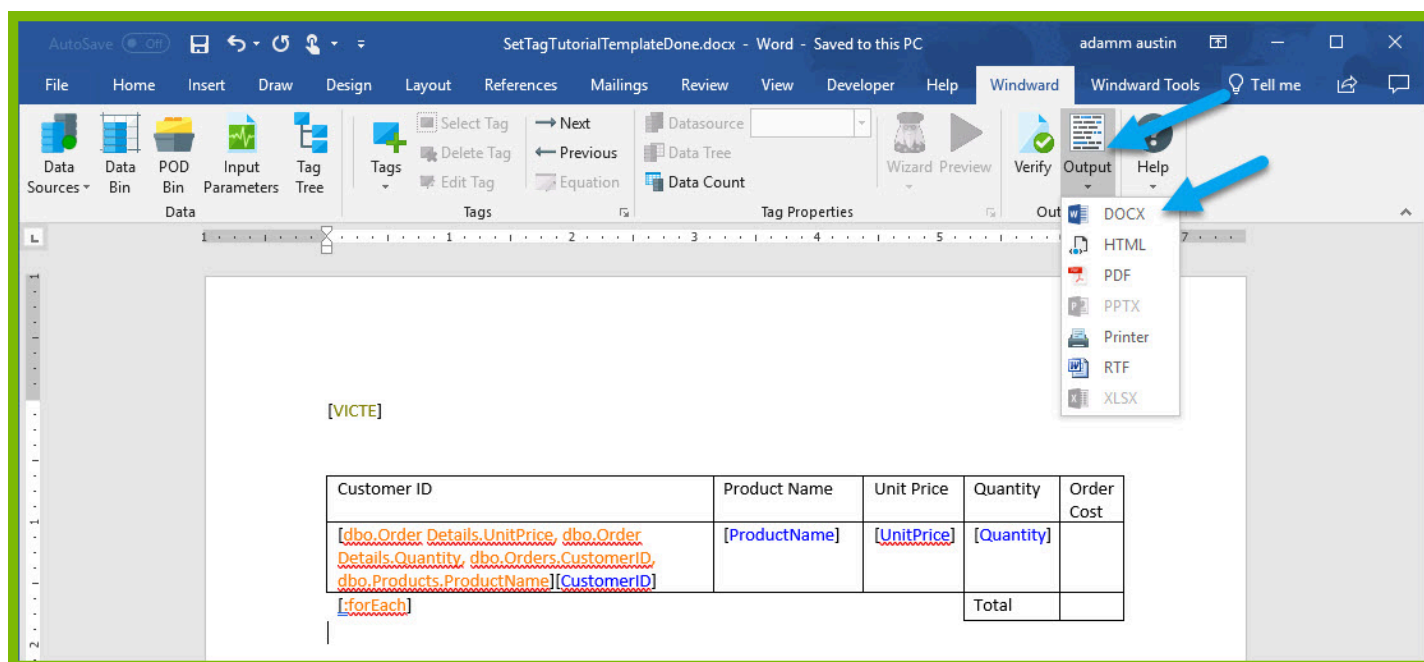

20. Place Out tags in columns Product Name, Unit Price, and Quantity, and repeat the steps above to give your Out tags selects on the nodes ProductName, UnitPrice, and Quantity respectively.

21. Before you proceed, add an EndForEach tag and try outputting your template to make sure you are on track. Place your cursor in the second second row of the table under the Customer ID column. (The first three columns of this row have no borders to improve the aesthetic design).
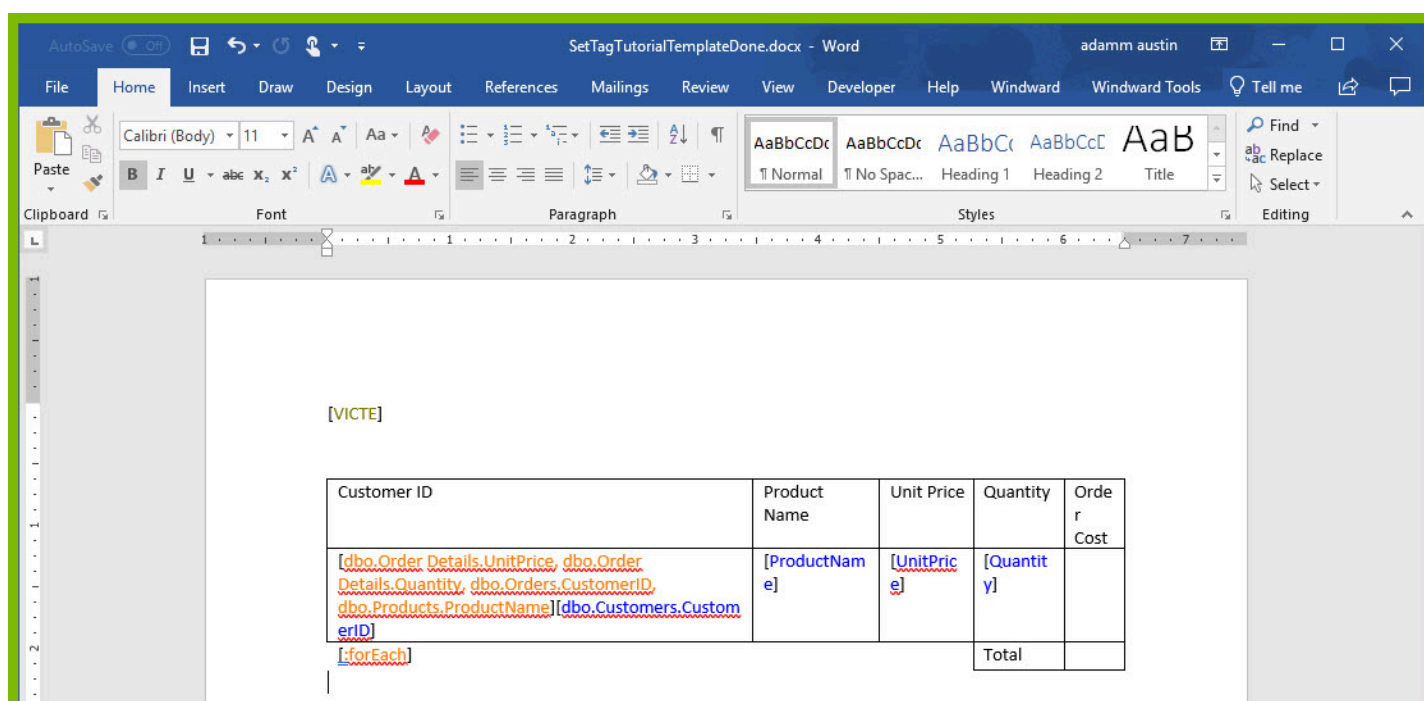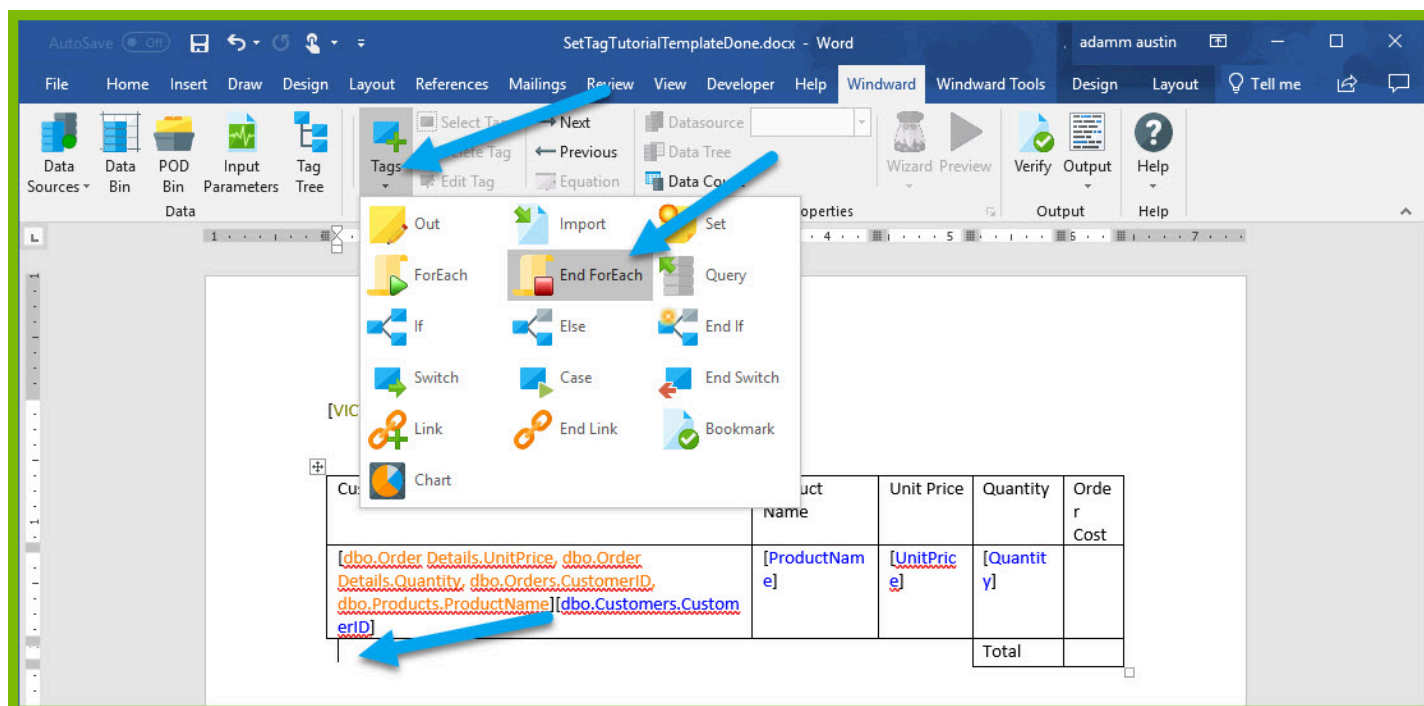
22. Click Output → DOCX to output to a DOCX format and you will get an output report of all of the orders from the Customer ID "VICTE" as well as the Product Name, Unit Price, and Quantity in the order.

23. Now our goal is to output the Order Cost for any one Order. (This will be the product of Unit Price and Quantity). Go back to your template and place a new OutTag in the first row of the Order Cost column and open the Tag Editor for the Out tag by clicking the "Edit Tag" button.
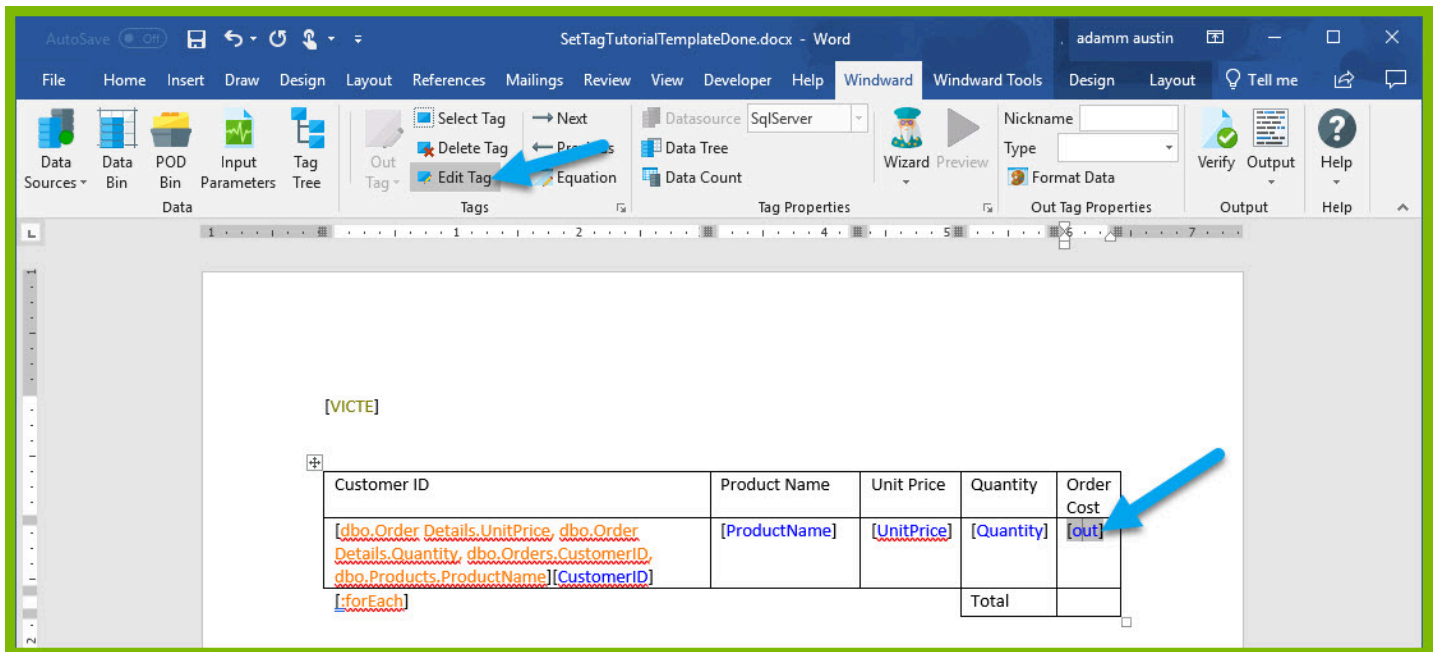
24. In the Tag Editor, you can take the product of our two variables using an equation. Click the Equation button in the Tag Editor.
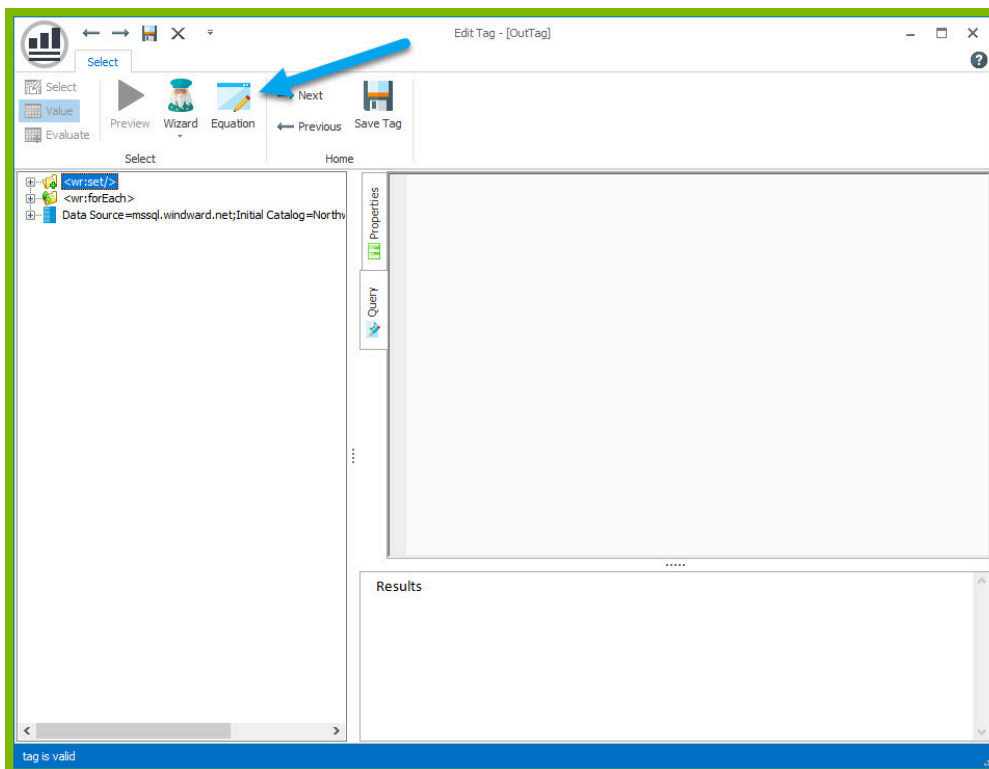

25. In the Equation dialog, find the PRODUCT equation. This will allow you to take the product of your two template variables. Select the PRODUCT equation and click OK.
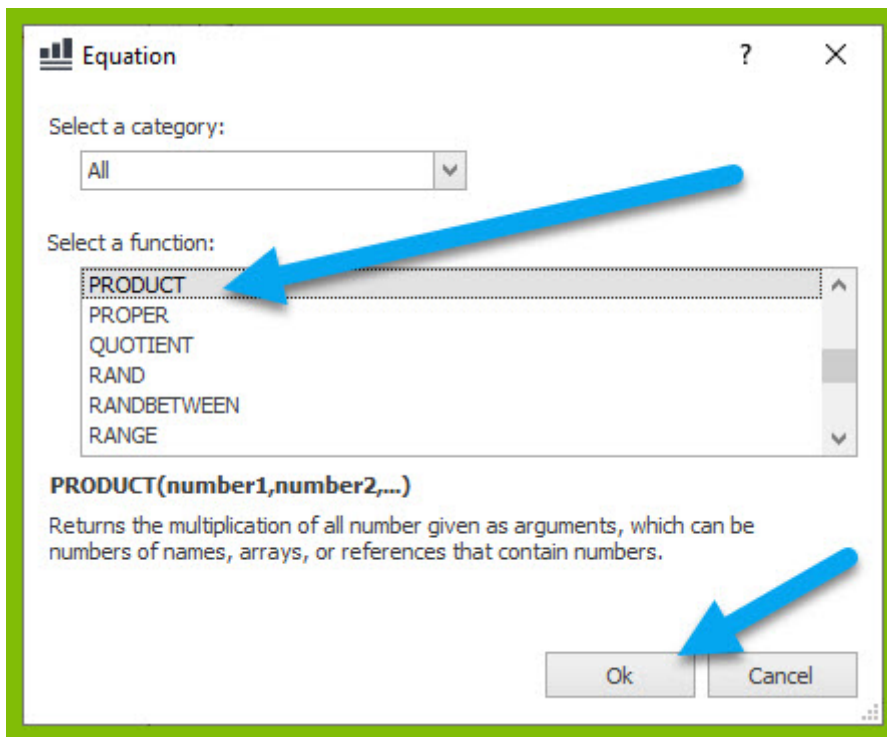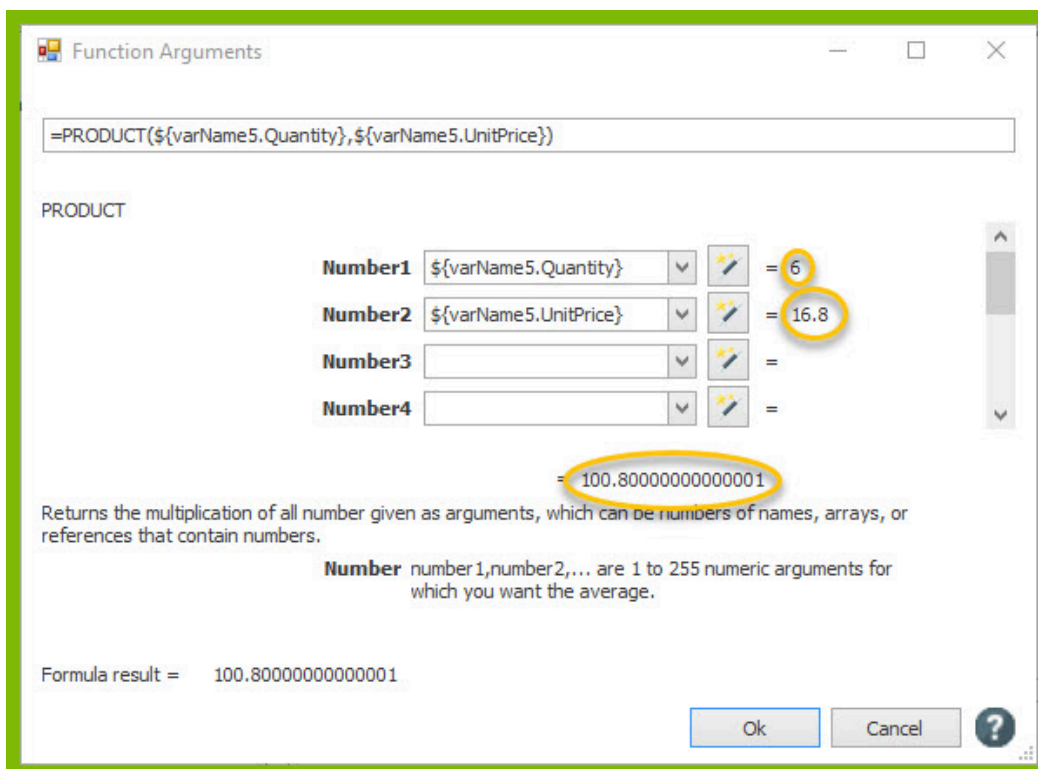
26. In the Function Arguments dialog, you can define the numbers you want to multiply. Use the down arrow to select from your Template Variables. Expand the drop down for Number1, and double click "Quantity."
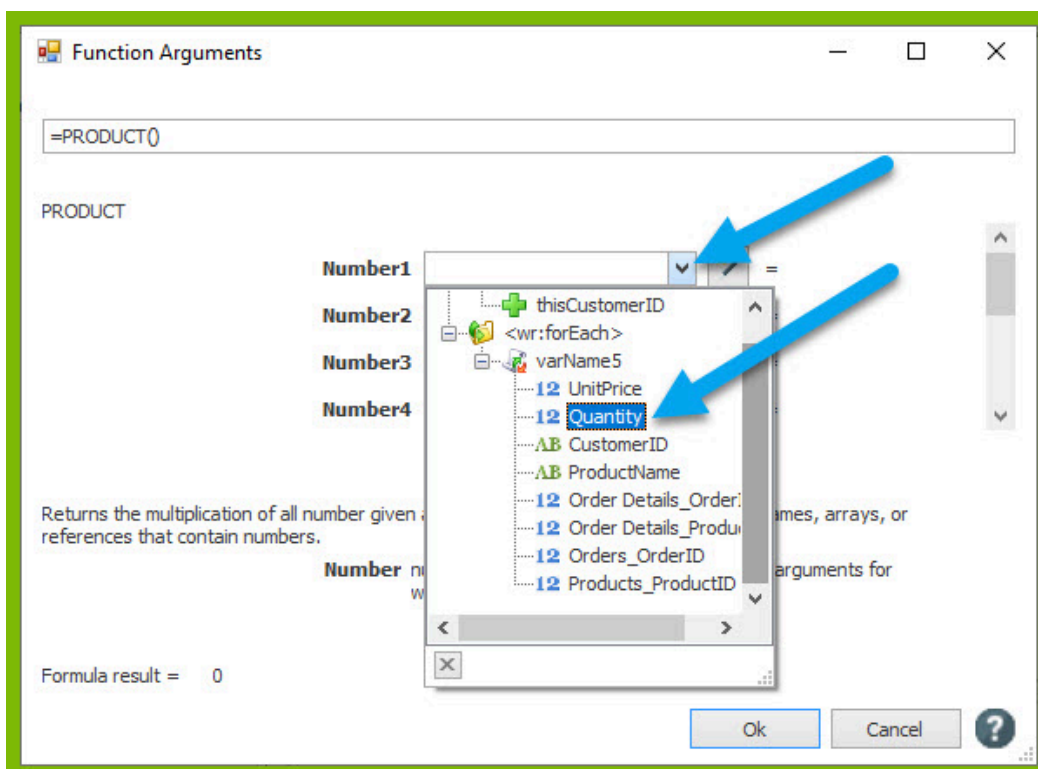

27. Expand the drop down for Number2, and double click "UnitPrice."
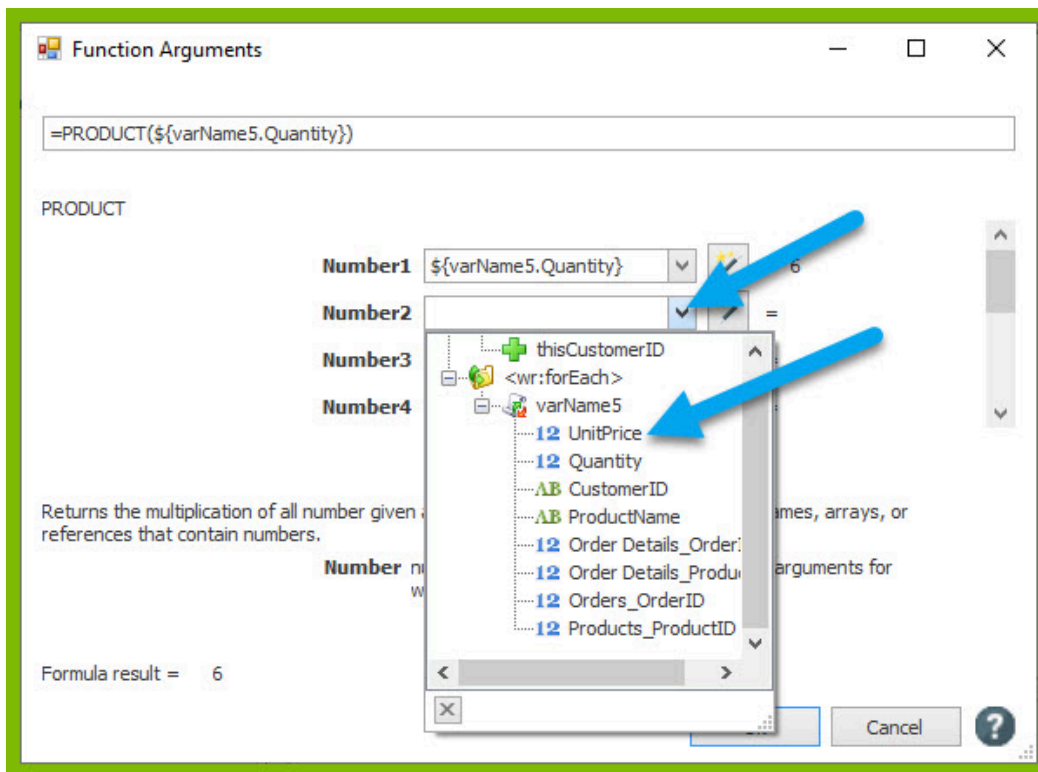
28. You can preview the result of your product in the Function Arguments dialog to make sure it looks as you expect.

29. Click OK to close the Function Arguments dialog. You will see the equation you created appear in the query pane of the Tag Editor, and previewing will show the result of the equation you made.
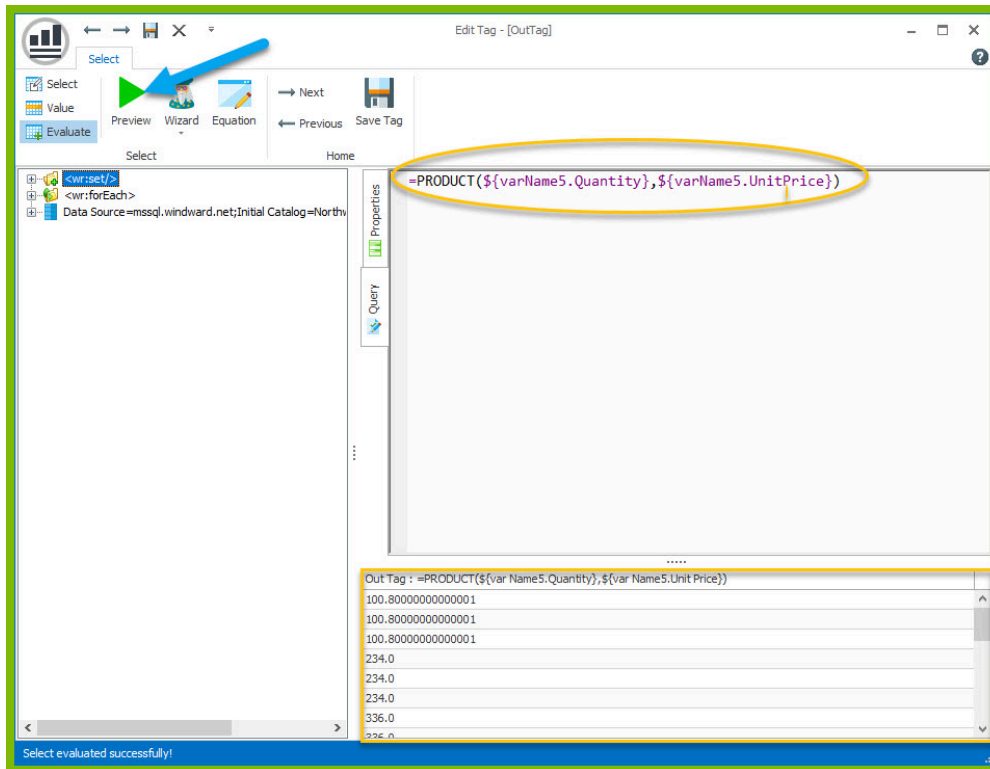
30. Save your tag.

31. Now your challenge is to display the total cost of all orders by the customer in the "Total" row of your table. You can achieve this by declaring a template variable outside of the table, and then add to this template variable for each Order Cost you calculate in the table. The first step is to declare our template variable outside of the table using a Set tag. Place a Set tag on the second line of the template.
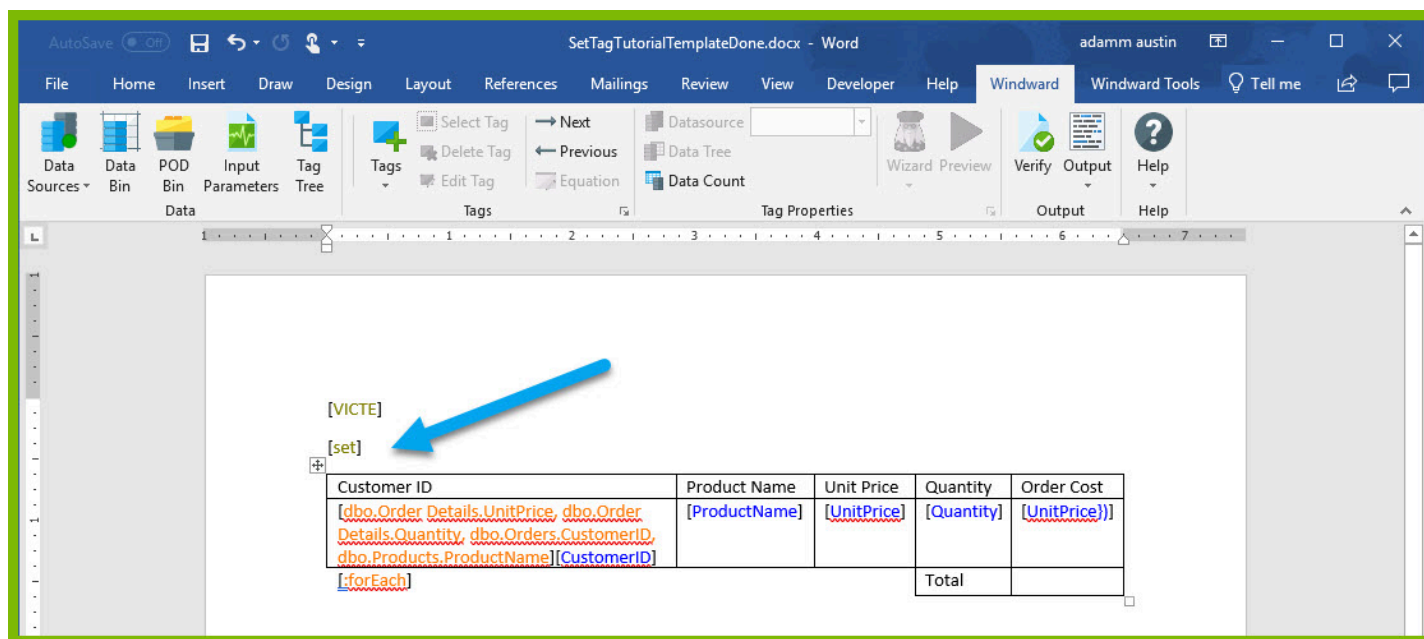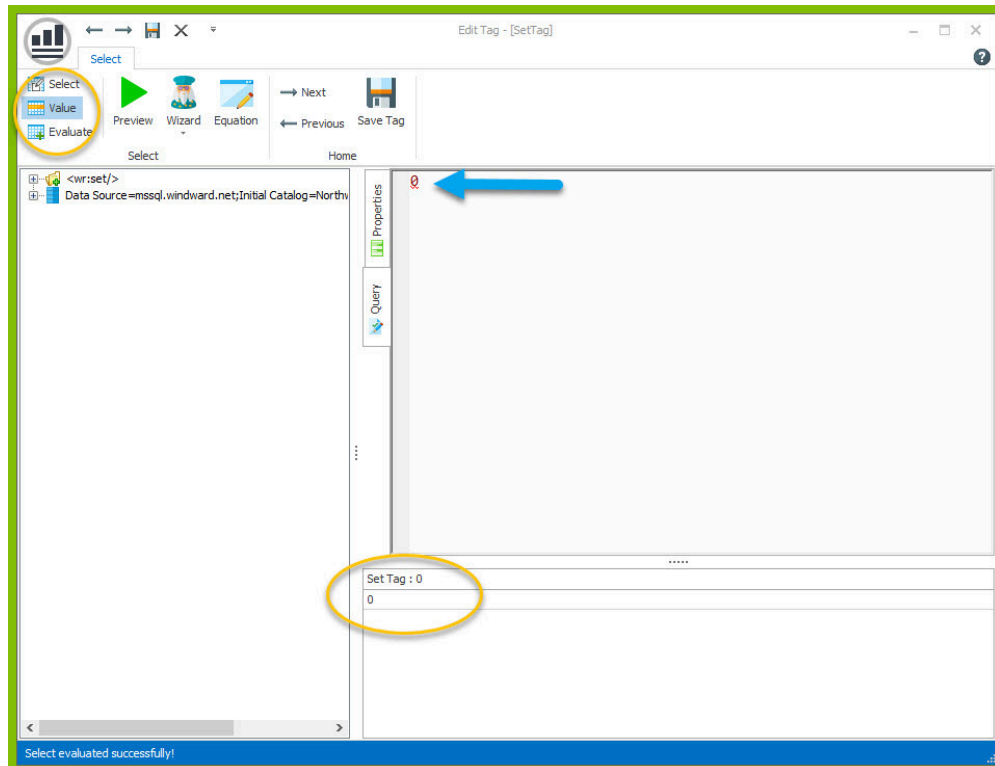
32. Place your cursor on the Set tag, and open the Tag Editor by clicking "Edit Tag"

33. Specify the initial value of the Set tag by entering 0 into the query pane.

34. Preview the tag to be sure it contains the value 0.

To set a variable name for the tag, select the Properties pane in the Tag Editor, and change the var value to "totalOrdersCost". Click the Save Tag button.

36. Now that you have a template variable to contain the total, we need to add the order cost calculated for each order to the "totalOrdersCost" variable. We can re-use the value output by our Order Cost Out tag by assigning a variable name to the Out tag, and then using the Out variable in our sum. Open the Out tag in the Order Cost column in the Tag Editor.

37. In the "Update tag references" dialog, click Yes to update all references to the variable name in the document.

38. To set a variable name for the tag, select the Properties pane in the Tag Editor, and change the var value to "currentOrderCost". Click the Save Tag button.

39. In the "Update tag references" dialog, click Yes to update all references to the variable name in the document.

40. Place a Set tag in the same cell after the Out tag in the Order Cost column and open the Tag Editor for the Set tag.

41. To update the value of the template variable, we will need to give this Set tag the same variable name as our template variable to hold the total orders cost. To set a variable name for the tag, select the Properties pane in the Tag Editor, and change the var value to "totalOrdersCost".

42. Now you must give the tag a select that will sum the existing currentOrderCost with the totalOrdersCost. You can do this with an equation like you calculated the product earlier. Click the Equation button in the Tag Editor.
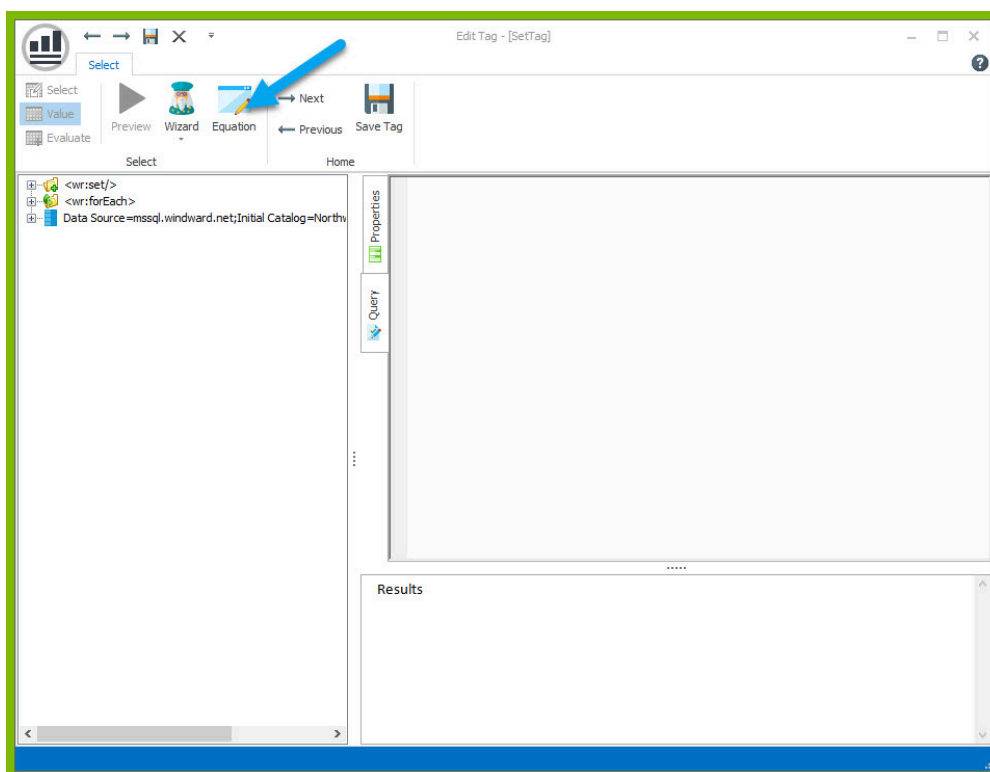

43. In the Equation dialog, find the SUM equation. This will allow you to add up the total of your two template variables. Select the SUM equation and click OK.
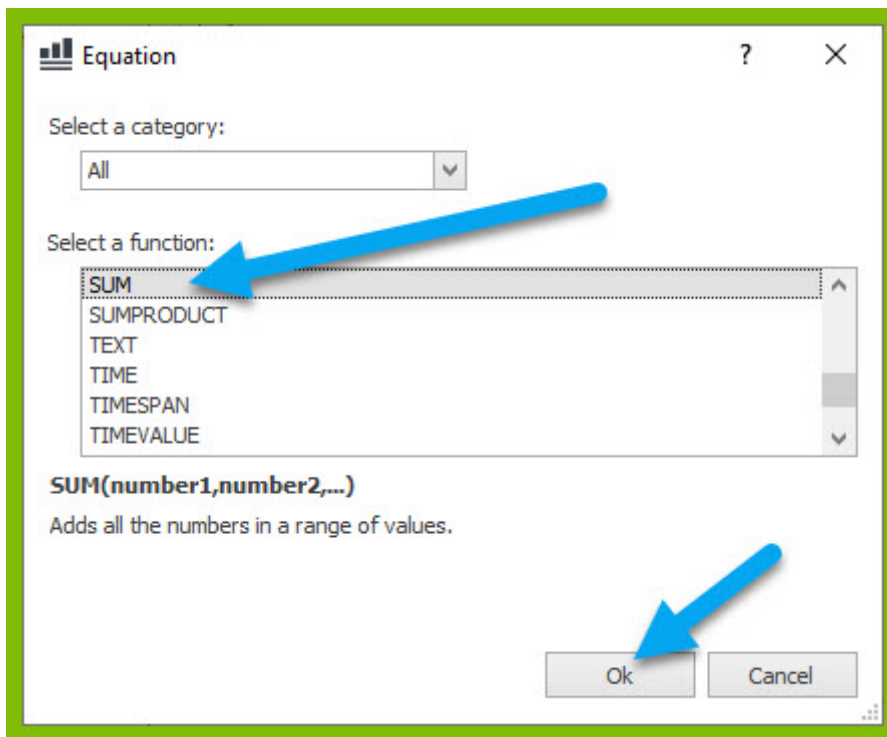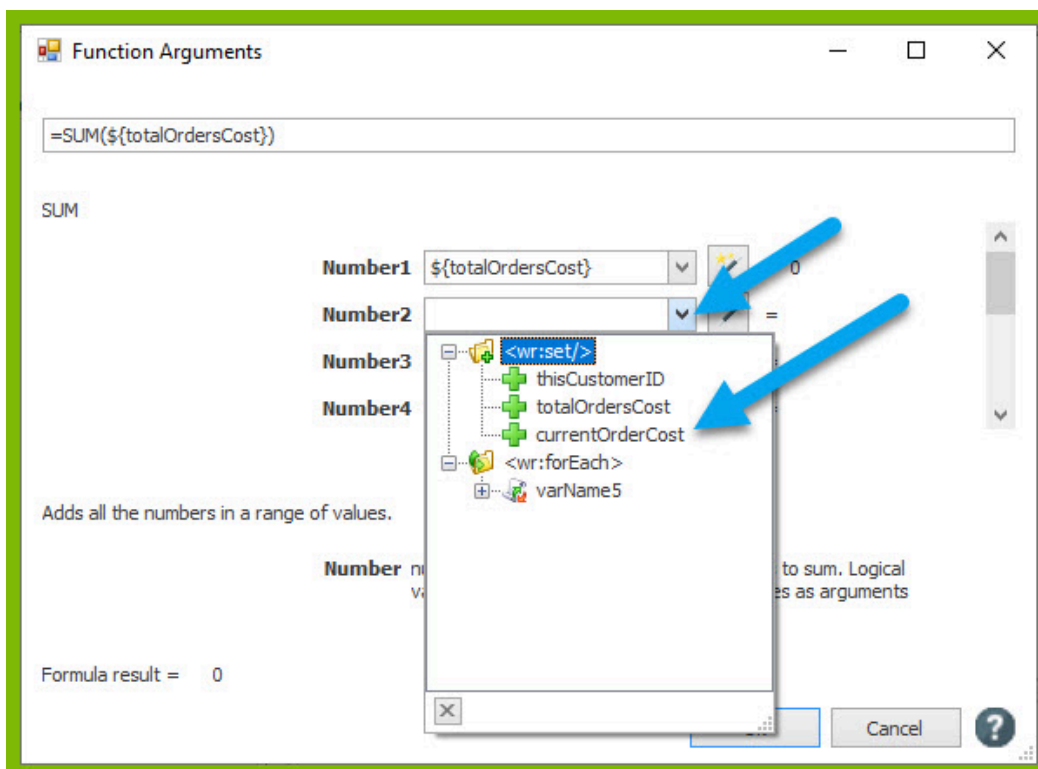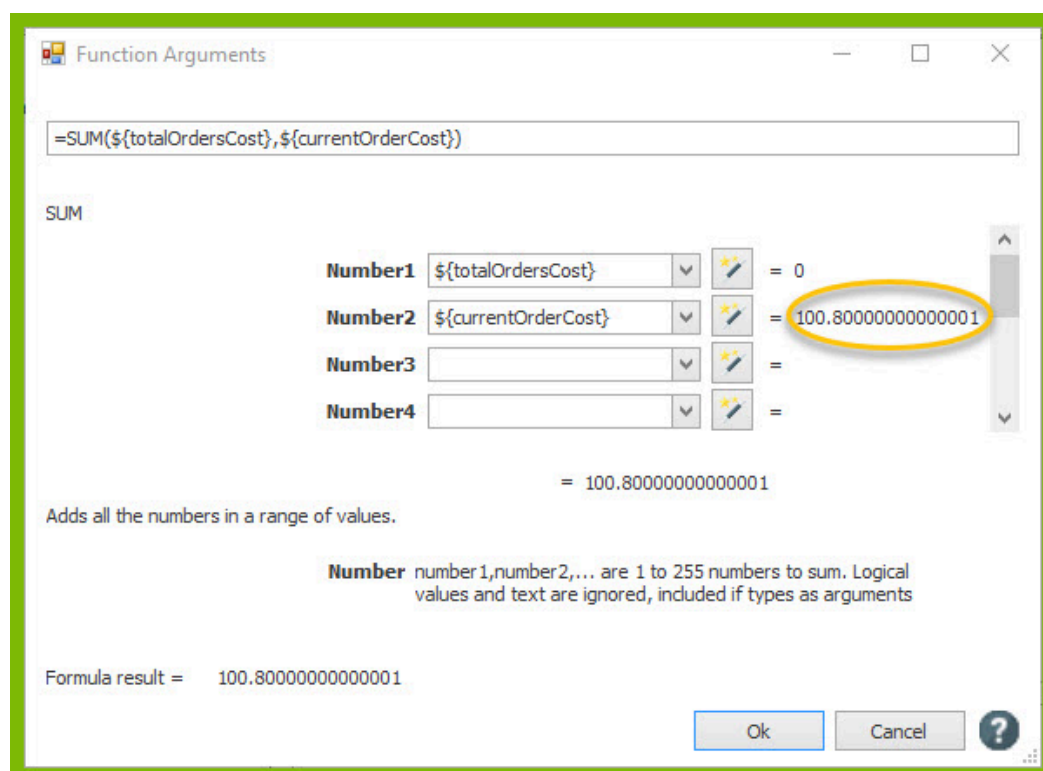
44. In the Function Arguments dialog, you can define the numbers you want to add. Use the down arrow to select from your Template Variables. Expand the drop down for Number1, and double click "totalOrdersCost". This is the Set tag variable that will hold the total across all orders. Notice after you select it, that the previewed value is 0. This is the value the variable was initialized to in the Set tag.
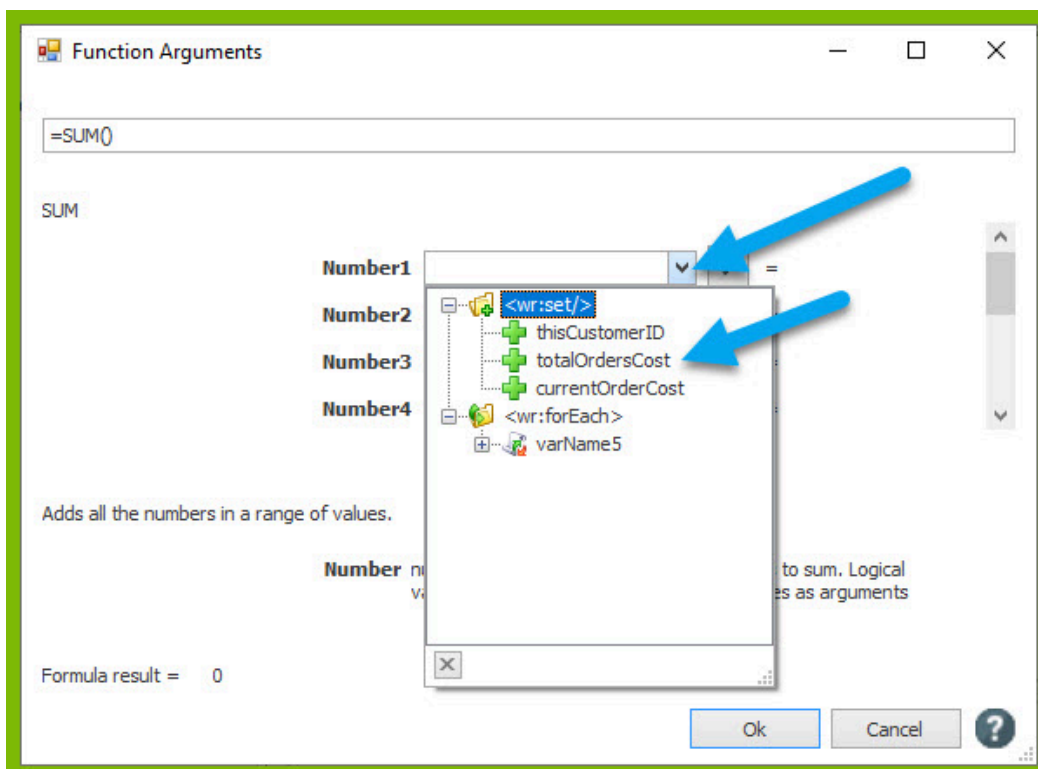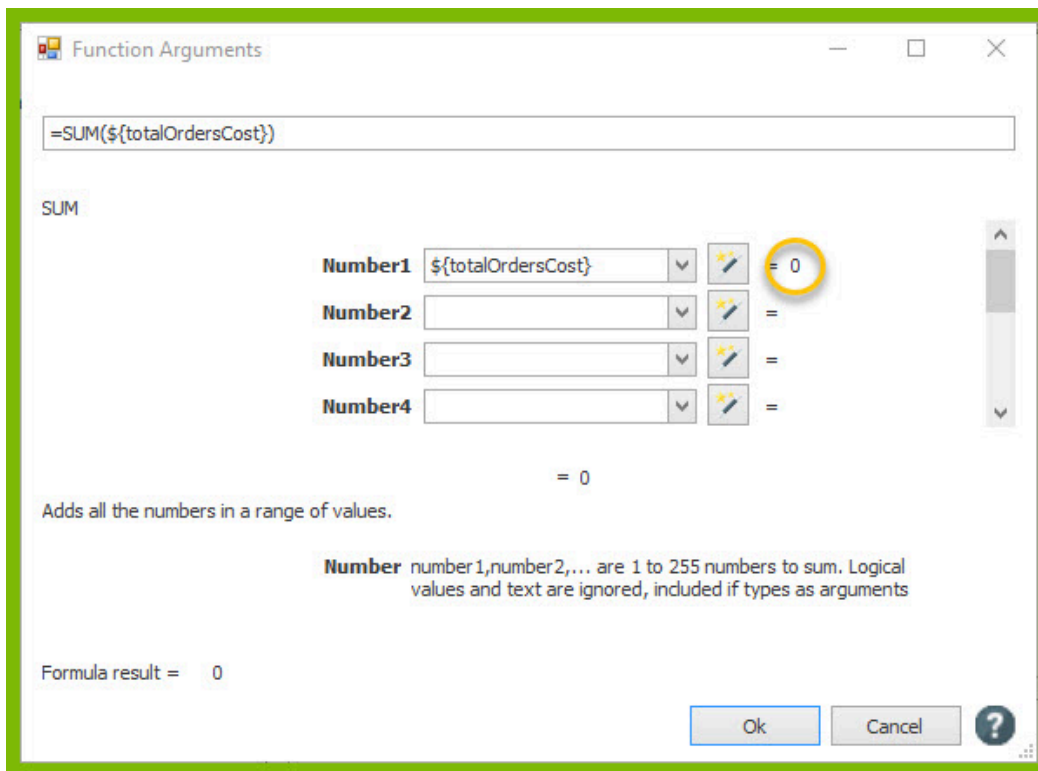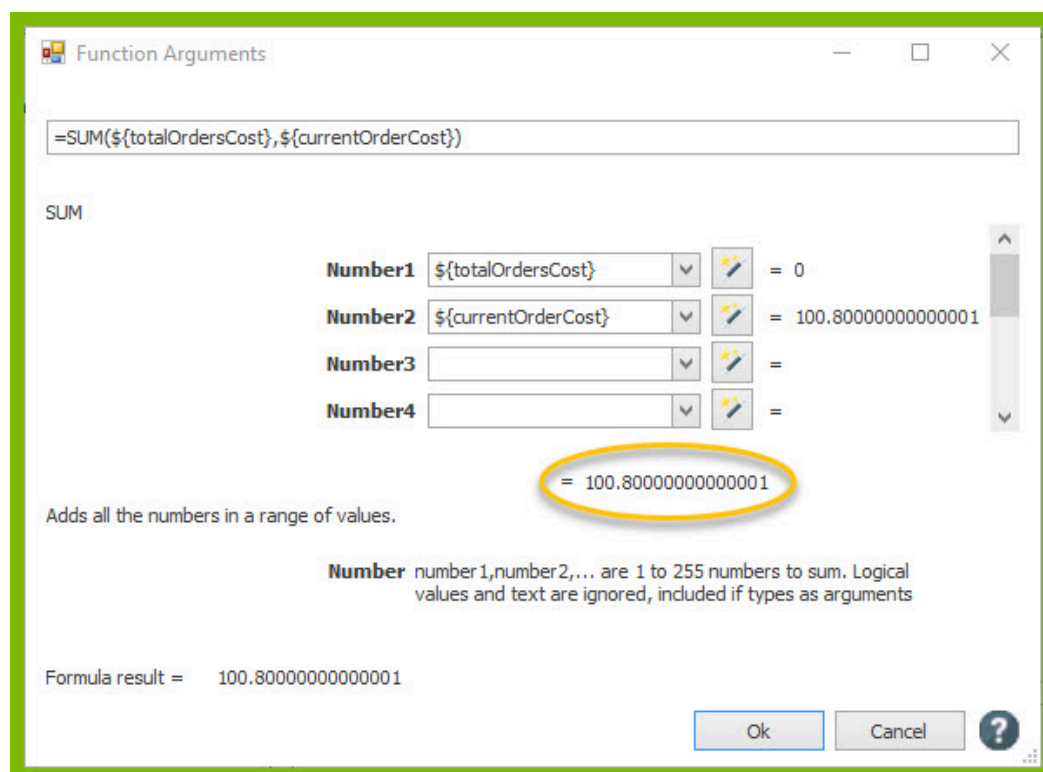
45. Expand the drop down for Number2, and double click "currentOrderCost". Notice after you select it, that the previewed value is 100.80. This is the first value contained by our Out tag variable after calculating the first product.

46. You can preview the result of your product in the Function Arguments dialog to make sure it looks as you expect.
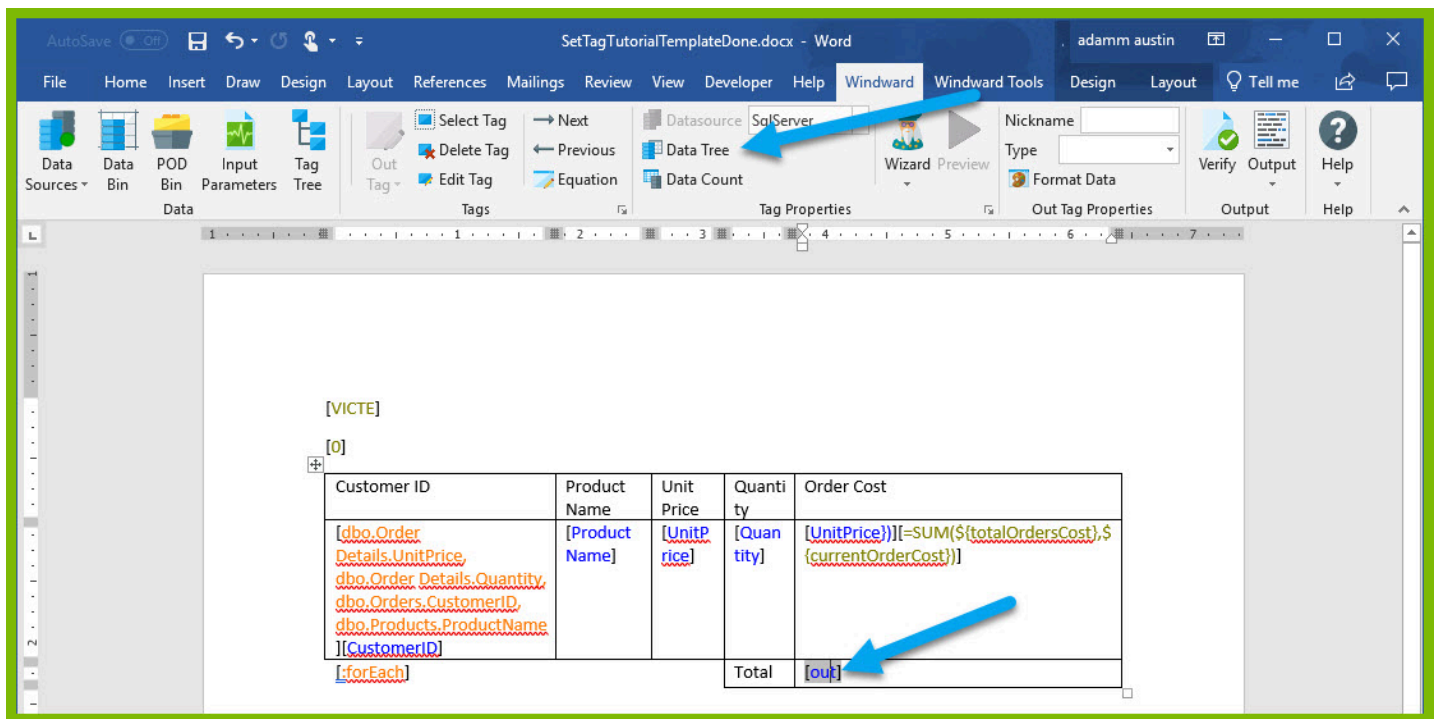
47. Click OK to close the Function Arguments dialog. You will see the equation you created appear in the query pane of the Tag Editor, and previewing will show the result of the equation you made.

48. Click Save Tag to save your tag.

49. In the "Update tag references" dialog, click Yes to update all references to the variable name in the document.

50. Your last step is to add an out tag to output the totalOrdersCost variable after adding the cost of each individual order to the variable. Place an Out tag in the cell to the right of the "Total" cell.

51. Place your cursor on the Out tag, and open the Data Tree from the Tag Properties section.

52. From the Data Tree, select the variable totalOrdersCost under your Set variables.
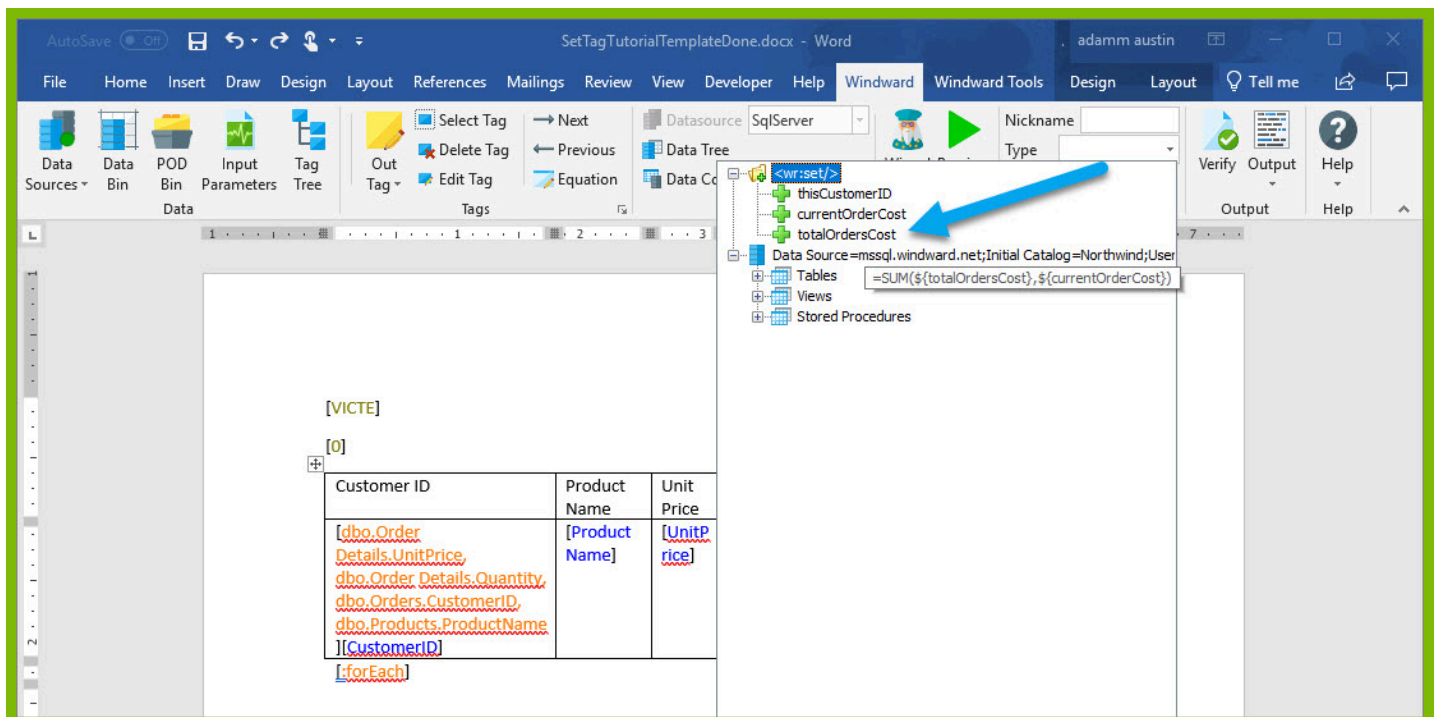
53. Now you can output your finished template! Click Output → DOCX and view your output template. Your output contains a list of orders, the order quantity, unit price, order cost, and a total cost of all their orders!

Now that you have finished this introduction tutorial, here are some resources we recommend you visit next:

1. [How to connect to XML datasources](#)

2. [How to use the XPath Wizard](#)

3. [The Windward Tutor: A collection of short video tutorials](#)