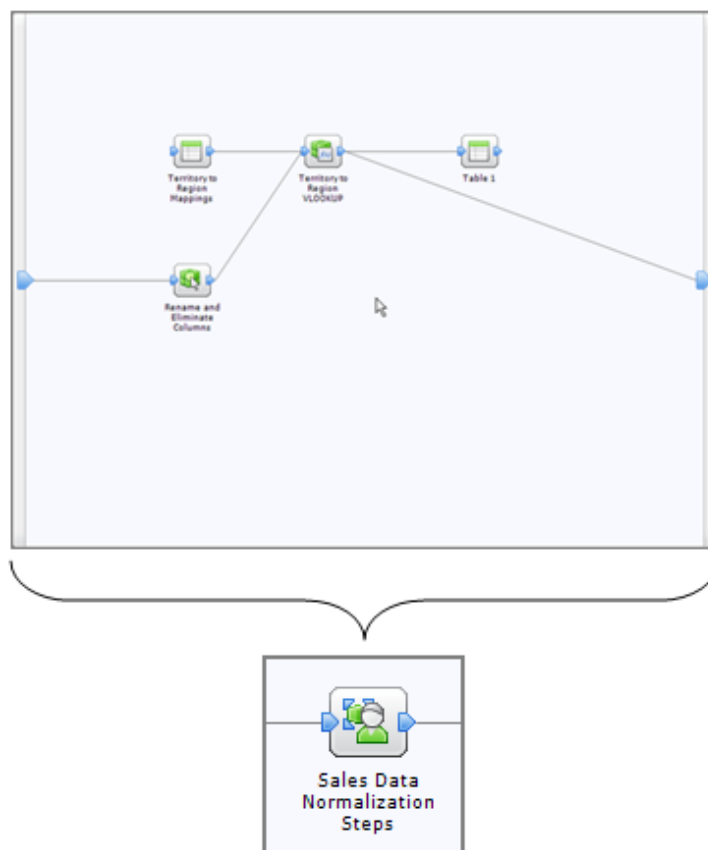


Building Components

Lesson 2



In Lesson 2, you will create Components out of primitive Proto modules as you build a revenue analysis dashboard to explore sample sales data.

This tutorial should take 40-60 minutes to complete.

Lesson Overview

In Lesson 2, you will learn to use primitive Proto modules to build data normalization and analysis Components. We will cover the following functionality and concepts:

- Use the Attribute Editor, Table, Entryset Formula, Simple Report, Pie Graph, Entryset Module and Listbox primitive modules to process data and create user interfaces.
- Group a selection of modules together to create a new Component.
- Create a new empty Component to work with.
- Save Components out of your dashboard to a folder to reuse them.
- Arrange, resize and combine Components and primitive controls in the Viewer.

The dashboard used in the Lesson 2 includes sample sales data to work with. First you will normalize the data and create a new column that maps existing sales territories to new regions. Then you will group the modules that perform the “normalization” processes into a new Component to illustrate how to build, save and reuse functionality in Proto. Next you will create a dynamic report that summarizes total revenue for each of the normalized sales regions. And finally, you will group the analysis functionality into another Component.

Before you start this lesson, you will need to have installed Proto Financial Version 1.9 and downloaded the Lesson 2 files from the Proto website. Lesson files can be downloaded in a zipped format from Proto’s website at <http://www.protosw.com/devcenter>

Before jumping into the tutorial, you might want to take a quick moment to explore and understand the Proto dashboard that you will be creating. To do so,

- 1 Locate the folder named “Lesson 02” in the folder titled “Proto Getting Started Lessons.”
- 2 Open the file named “Lesson 02 (Completed) – Sales Data Analysis.proto.”
- 3 Click “OK” if prompted that the file has VBA macros.

To begin building the dashboard, first open the lesson file and save a copy as “Sales Data Analysis.proto.”

- 1 Locate the folder named “Lesson 02” in the folder titled “Proto Getting Started Lessons.”
- 2 Open the file named “Lesson 02 (Begin) – Sales Data Analysis.proto” by double-clicking on the file.
- 3 Save this Proto dashboard with a new name in the “Lesson 02” folder. To do so, select File > Save As from the application menu bar. Then browse to the appropriate location and save the file as “Sales Data Analysis.proto.”

Normalize the Raw Sales Data Export

In the Viewer, take a moment to inspect the sample sales data. This data could have come from a sales database, a CSV file, a hosted sales application like Salesforce.com or any other source. The sample data is preloaded into Proto so that the screenshots and descriptions you see in this lesson are consistent with what you see on your screen. However, in a real analysis dashboard, the initial steps might require a user to pick a new CSV file, log into a sales account, refresh a database query or get data in a variety of other ways.

Unlike the sample data in Lesson 1, this data needs to be processed before it is ready for analysis. The imperfections are minor, but they illustrate the frequent need to clean and normalize data coming from disparate systems. In Proto, it is a common practice to develop “normalization” Components that can be reused to pre-process standard data sources.

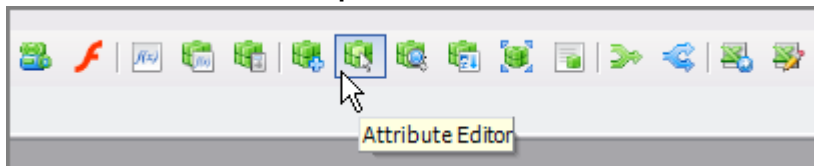
To normalize the sales data we will:

- I. Add an Attribute Editor module to change column names and eliminate some unnecessary columns of data.
- II. Insert a Table control and create an empty Entryset to type mappings from territories to regions directly into the dashboard.
- III. Use an Entryset Formula to add a calculated column to the data, which looks up the territory from the mappings table and fills in the appropriate region (like a VLOOKUP function in Excel).
- IV. Preview the normalized dataset while you work, using a Table module.

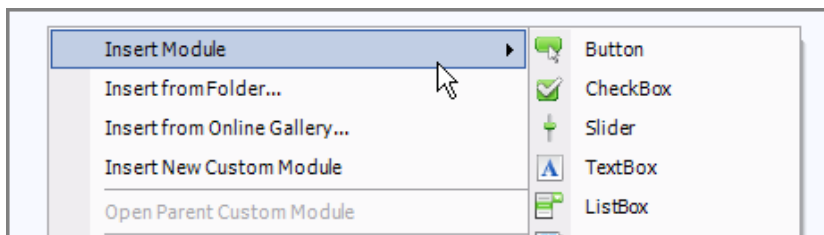
Add the Attribute Editor and Table modules to the Builder.

You can add primitive modules to the Builder in two ways:

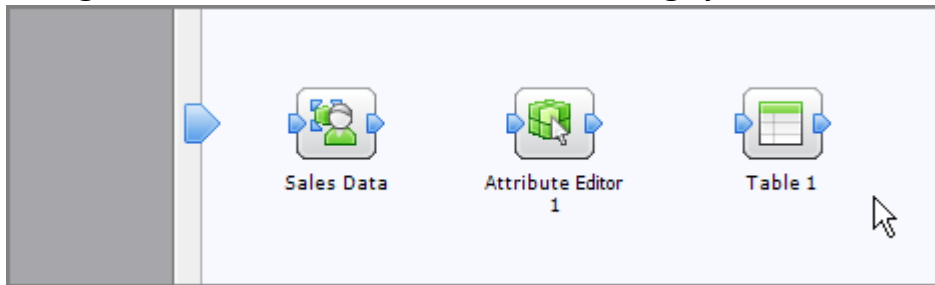
- 1 Use the toolbar at the top of the Builder window



- 2 Or right-click anywhere on the background of the Builder and select the primitive module from the “Insert Module” menu:



- 3 Try adding the Attribute Editor and Table modules to the Builder.
- 4 Configure the modules in the Builder to look roughly like the screenshot below.



Configure the Attribute Editor

Many primitive modules must have a Connection present to be configured. This is apparent in the Attribute Editor, which allows you to selectively include, create, merge and rename columns from Entrysets. To configure the Attribute Editor, an input Entryset must already be connected to the “Set Entryset” input. If you double-click on the Attribute Editor before making a Connection to it, you will notice that the configuration section of the Property Panel is empty.

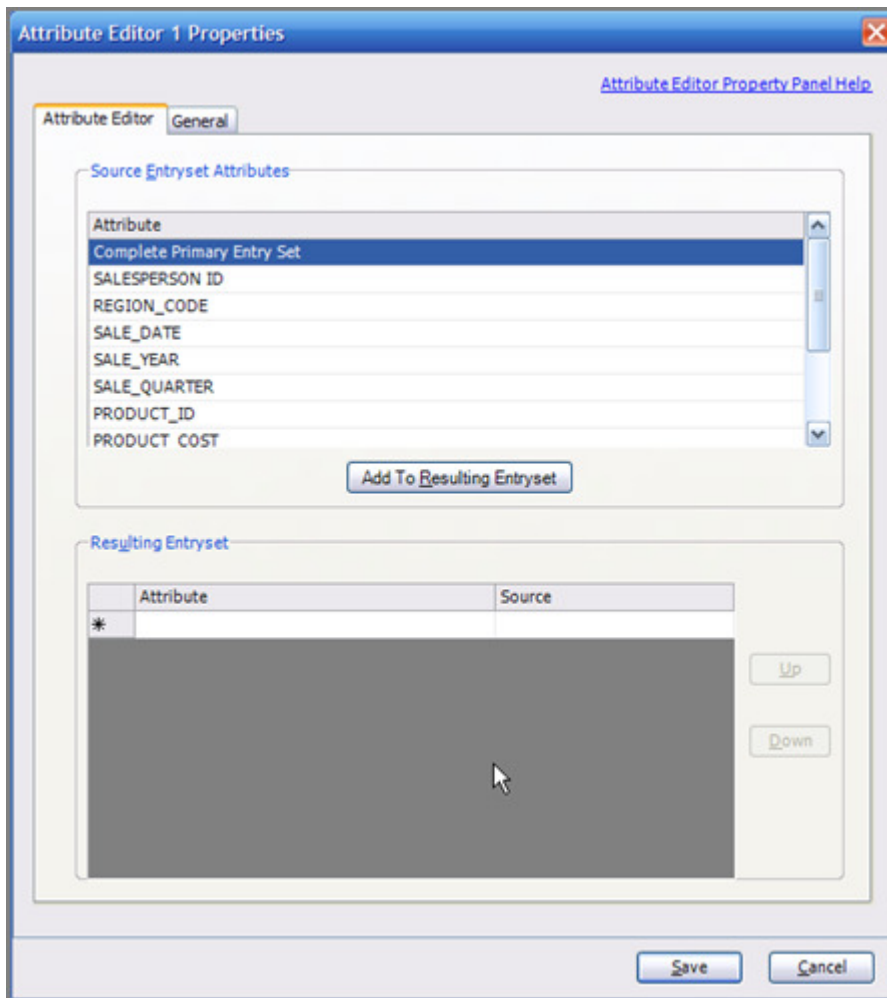
To configure the Attribute Editor, make a Connection to it from the sample data and then open its Property Panel.

- 1 First make the following connection from the sample sales data to the Attribute Editor.

From:	Sales Data
Output:	Sample Sales Data
To:	Attribute Editor 1
Input:	Attribute Editor Connectors > Set Primary Entryset

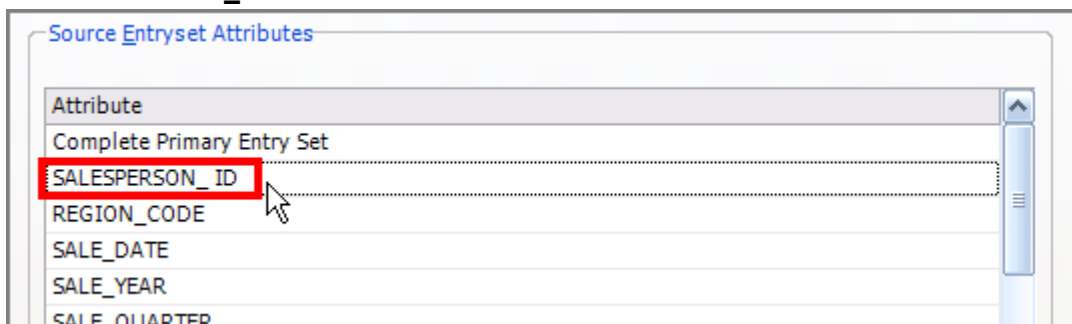
Note: Modules are appended with a number when you insert them, beginning with the number “1.” You may see instructions to make Connections to a module such as “Table 1” when in fact the module on your screen says “Table 2.” If there is any chance of ambiguity, you will be instructed to rename the modules prior to creating any Connections.

- 2 Now double-click on the Attribute Editor, or right-click and select “Properties” from the context menu to open the Attribute Editor’s Property Panel.



- 3 In the Property Panel you can see that the columns of the input Entryset are shown in the “Source Entryset Attributes” section. You will define how the input Entryset is translated to the “resulting” Entryset by adding rows to the configuration section in the “Resulting Entryset” group of the Property Panel.

Start by double-clicking on the row in the “Source Entryset Attributes” called “SALESPERSON_ID”

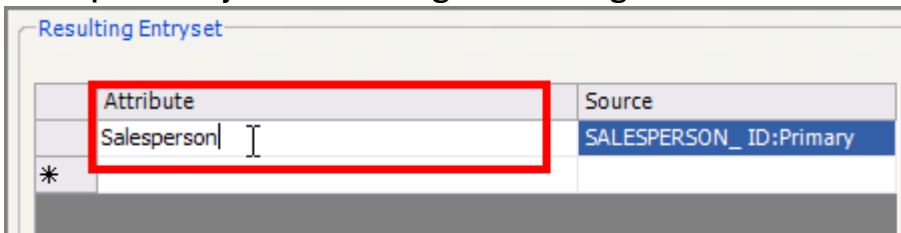


- 4 A row is added to the “Resulting Entryset” section. The “Source” column is equal to “SALESPERSON_ID:Primary” indicating that the first column in the resulting Entryset should get its source values from the “SALESPERSON_ID” Attribute of the

input Entryset. For the time being, don't worry about what the ":Primary" indication means. This is relevant only when there are two Entryset inputs to the Attribute Editor (*Primary and Secondary*).



- The "Attribute" column is also equal to "SALESPERSON_ID." This is the name that the column will be given in the Resulting Entryset. Change this to simply, "Salesperson" by double-clicking in the configuration table and changing the text.



- You have just configured the Attribute Editor to output a 1-column Entryset with a column named "Salesperson" sourced from the column "SALESPERSON_ID."

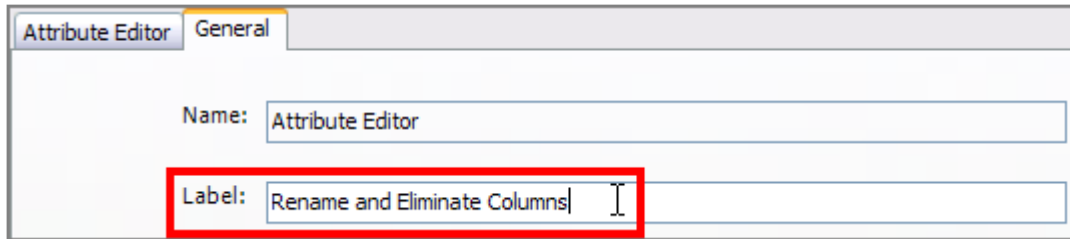
- Now make the following additional mappings to complete the configuration:

Attribute	Source
Salesperson	SALESPERSON_ID:Primary
Territory	REGION_CODE:Primary
Date	SALE_DATE:Primary
Year	SALE_YEAR:Primary
Quarter	SALE_QUARTER:Primary
Product Code	PRODUCT_ID:Primary
Revenue	REVENUE:Primary
Quantity	QUANTITY:Primary
Maintenance	MAINTENANCE:Primary
Total Revenue	TOTAL_REVENUE:Primary

Note: You can accidentally end up with an "empty" mapping if you double-click in the star (*) row and then click out. You will not be able to save the Attribute Editor settings in this state. To delete a row in the Resulting Entryset configuration section, click once on the row and hit the "Delete" key on your keyboard.



- Next, go to the “General” tab in the Property Panel, and change the label to “Rename and Eliminate Columns” to document what the Attribute Editor is doing for future reference.



- When you have finished configuring the Attribute Editor, click “Save” at the bottom of the Property Panel.

Connect the Attribute Editor output to the Table

To see what the Attribute Editor does, connect its output to the Table module’s input. After doing so, you can view the resulting Entryset in the Viewer.

- Make the following connection:

From:	Rename and Eliminate Columns
Output:	Attribute Editor Connectors > Get Entryset
To:	Table 1
Input:	Table Connectors > Set Entryset

- Go to the Viewer and enter Edit Mode by selecting “Tools > Edit Controls” from the application menu bar or hitting “Ctrl-E”.
- Move and resize the Table in the Viewer to examine the Entryset.

Note: The Table will appear **on top of** the sample data workflow step and may look somewhat confusing. We will address this visual clutter next.

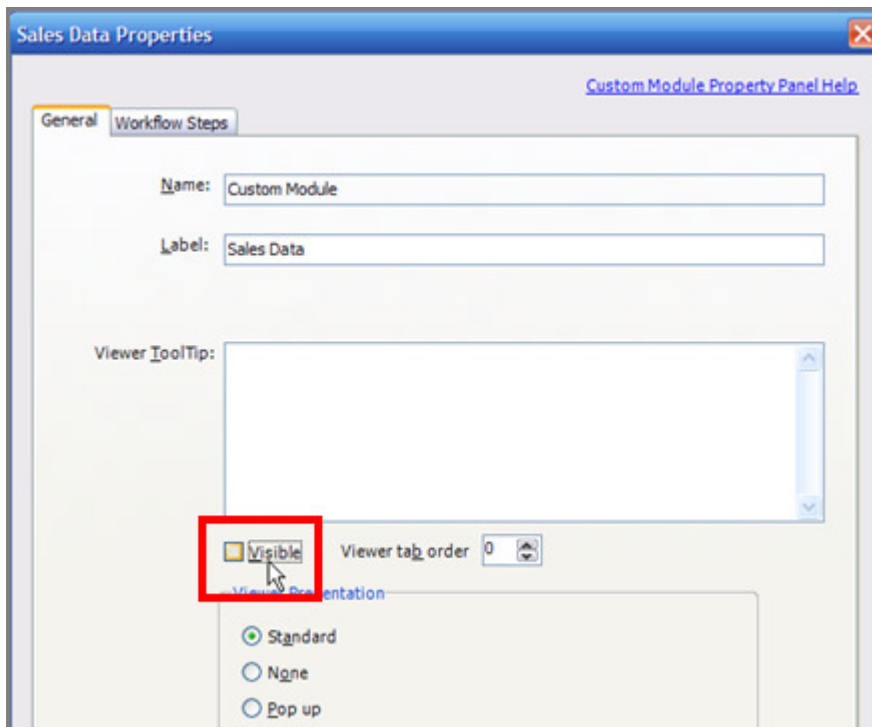
Hide the Workflow Steps to Temporarily Simplify the Viewer

To simplify the Viewer space while you add controls to build a Component, you will hide the workflow steps and make the sample data Component invisible. This will make it much easier to work within the Viewer.

- Select “View > Hide Workflow” from the application menu bar.
- Your Viewer window will now look something like the screenshot below, with a Table shown on top of the sample data Component.

	SALESPERSON_ID	REGION_CODE	SALE_DATE	SALE_YEAR	SALE_QUARTER	PRODUCT_ID	PRO
1	Salesperson 5	Central-East	5/13/2008	2008	Q2	B	
2	Salesperson 1	Salesperson	Territory	Date	Year	Qu	
3	1	Salesperson 5	Central-East	5/13/2008	2008	Q2	
4	2	Salesperson 5	Central-East	2/19/2008	2008	Q1	
5	3	Salesperson 5	Central-East	1/10/2008	2008	Q1	
6	4	Salesperson 5	Central-East	1/18/2008	2008	Q1	
7	5	Salesperson 2	FL-PA	7/29/2008	2008	Q3	
8	6	Salesperson 5	Central-East	11/29/2008	2008	Q4	
9	7	Salesperson 2	FL-PA	6/14/2008	2008	Q2	
10	8	Salesperson 2	FL-PA	11/14/2008	2008	Q4	
11	9	Salesperson 5	Central-East	1/30/2008	2008	Q1	
12	10	Salesperson 4	Central-West	10/23/2008	2008	Q4	
13	Table 1						
14							
15	Salesperson 2	FL-PA	8/8/2008	2008	Q3	D	
16	Salesperson 2	FL-PA	1/12/2008	2008	Q1	B	

- To set the sample data Component visibility to “False,” return to the Builder and double-click on the Sales Data Component. This will open the standard Component Property Panel.
- Click on the checkbox titled “Visible” in the “General” tab to uncheck it.



- Click “Save” at the bottom of the Property Panel.

6 Return to the Viewer and note that the Sales Data Component is no longer visible.

	Salesperson	Territory	Date	Year	Q
1	Salesperson 5	Central-East	5/13/2008	2008	Q2
2	Salesperson 5	Central-East	2/19/2008	2008	Q1
3	Salesperson 5	Central-East	1/10/2008	2008	Q1
4	Salesperson 5	Central-East	1/18/2008	2008	Q1
5	Salesperson 2	FL-PA	7/29/2008	2008	Q3
6	Salesperson 5	Central-East	11/29/2008	2008	Q4
7	Salesperson 2	FL-PA	6/14/2008	2008	Q2
8	Salesperson 2	FL-PA	11/14/2008	2008	Q4
9	Salesperson 5	Central-East	1/30/2008	2008	Q1
10	Salesperson 4	Central-West	10/23/2008	2008	Q4

Table 1

Create the Mappings from Territory to Region (like VLOOKUP)

Frequently, enterprise systems and other central data repositories fall out of synch with the categorizations that business users want to use in reports. In the next few steps, you will learn how to fill in a new column with data using a function that is very similar to the VLOOKUP function in Excel.

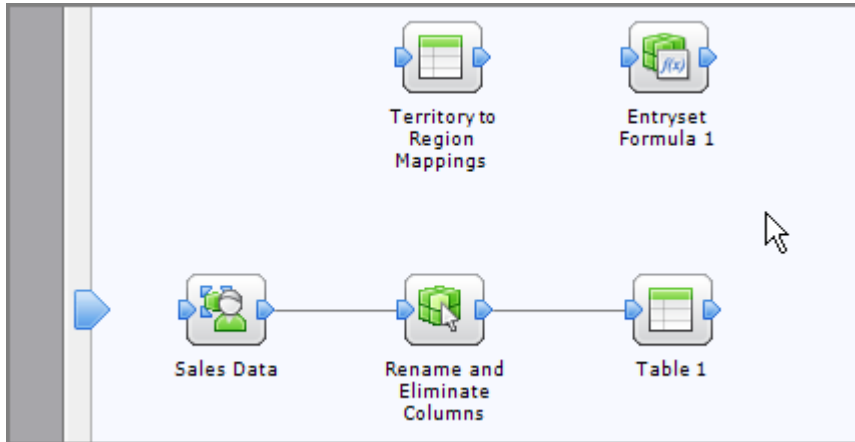
First you will create a new Table to type in a corresponding “Region” for each value in the “Territory” column. Then you will use an Entryset Formula module to apply a VLOOKUP function to a new column, for every row in the Entryset.

Note: In Excel, users frequently have to copy and paste formulas alongside tabular data when there is a change in the number of rows to process. In Proto, the Entryset Formula module automatically applies a formula to every row in an incoming Entryset.

Now go to the Builder and add a Table and an Entryset Formula

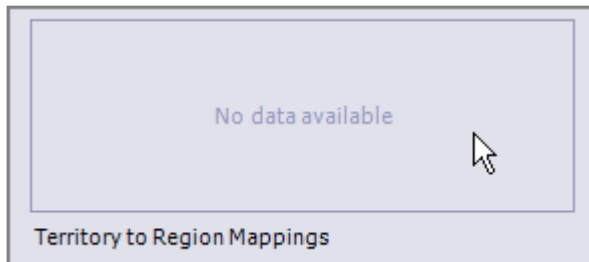
- 1 Insert a Table module in the Builder. Go to its Property Panel and change the Label to “Territory to Region Mappings” in the “General” Tab.
- 2 Insert an Entryset Formula module.

- 3 Your Builder window will look similar to the screenshot below.



Define an Entryset structure in the Table module.

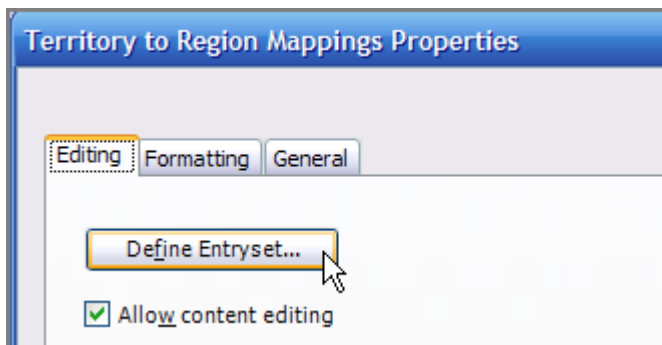
- 4 Go to the Viewer and note that the Table interface is empty.



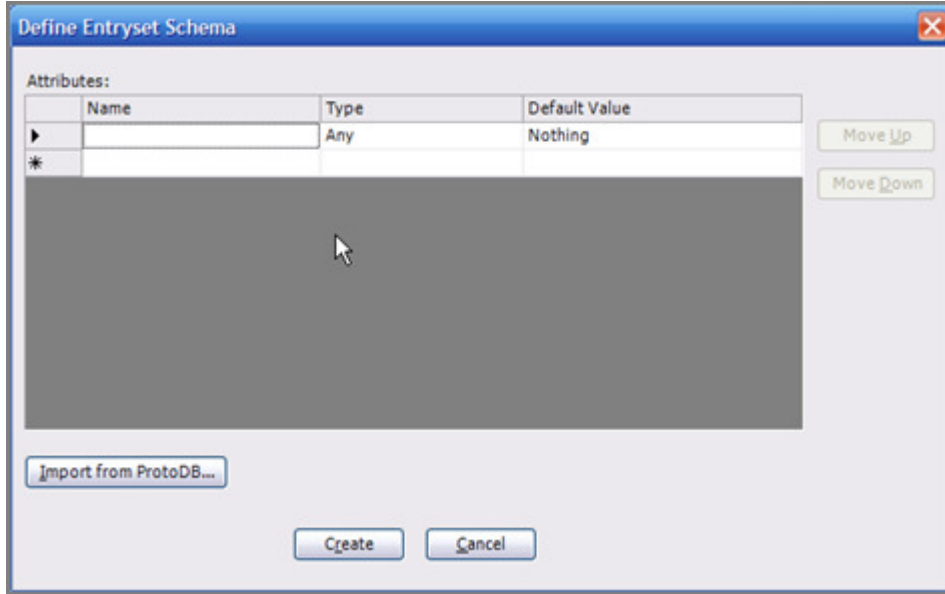
Note: The Table Module provides functionality to view, edit and create Entrysets. However Entrysets as a data structure exist independently from Table controls.

So far, you have only used Tables to display input Entrysets. Now you will define an Entryset structure in the Table module Property Panel, to create an empty Entryset that you can view, edit, etc. in the Table control.

- 5 Go to the Table module's Property Panel. You can do this either by returning to the Builder and interfacing with the Table module there, or you can right-click on the control in the Viewer and select "Properties..." from the context menu.
- 6 In the Property Panel, click on the button called "Define Entryset..." in the "Editing" tab.



- 7 A panel titled “Define Entryset Schema” will be displayed. This panel is also shown by other primitive Proto modules to edit the structure of an Entryset.

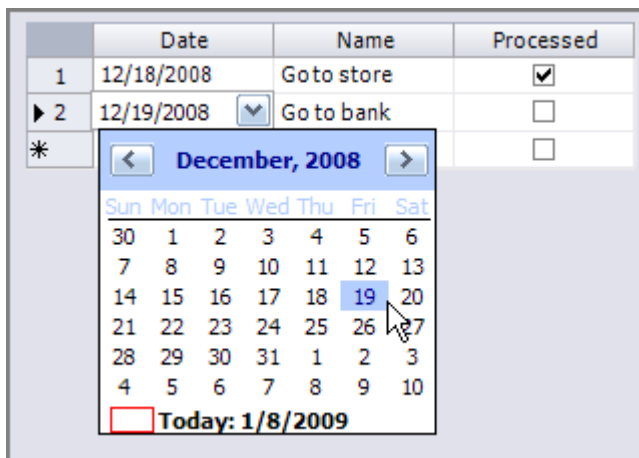


- 8 To define an Entryset, you will type in the Name, Type and Default Value for attributes. Configure the panel as shown below.

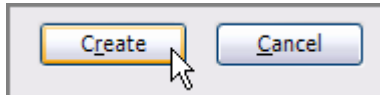
	Name	Type	Default Value
	Territory	Any	Nothing
▶	Region	Any	Nothing
*			

Note: Generally, Proto modules will treat data as whatever type “just works.”

When you import data from Excel, a database, a CSV, etc. you do not have to type the columns of data. However there are times when you may want to force a column to have a specific data type. The Table has settings to enforce data types that change how a user can edit values in the Table. For example, when a column is typed “Only Dates” a user can type in a value or select a date from a dropdown in the Table.

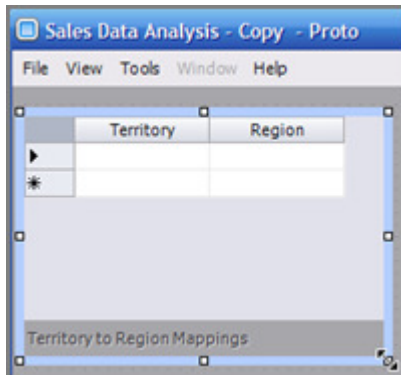


- 9 Click the “Create” button at the bottom of the “Define Entryset Schema” panel. This will close the panel and return you to the Table module’s Property Panel.



- 10 Then Click “Save” in the Table Module Property Panel.

- 11 Resize the Table control by entering Edit Mode to look something like the screenshot below.



- 12 Exit Edit Mode and type the following values in the Table.

	Territory	Region
1	FL-PA	East Coast
2	Central-East	Central
3	Central-West	Central
4	CA-WA	West Coast
▶ 5	NJ-NY-CT	East Coast
*		

To configure the Entryset Formula with a VLOOKUP function, return to the Builder.

The Entryset Formula module must be configured in particular sequence of steps.

- First an Entryset to process must be sent to the Entryset Formula. This is necessary because the structure of the Entryset is used to configure the Entryset Formula settings.
- Then go to the Entryset Formula’s Property Panel to configure the settings and define a formula expression that will be evaluated for each row of the input Entryset.
- Then save and close the property panel.
- And finally, make Connections from other controls and data sources to any variables defined in the Entryset Formula.

Note: Many other modules, such as the Filter and Formula, also require you to first create an “expression” or “condition” before you can make Connections to its variables. This can be a little confusing for some people who are used to the Excel

workflow of defining formulas **and** where to get the data for each variable at the same time.

For example, in Excel you can write $=2*(A1 + B4)$

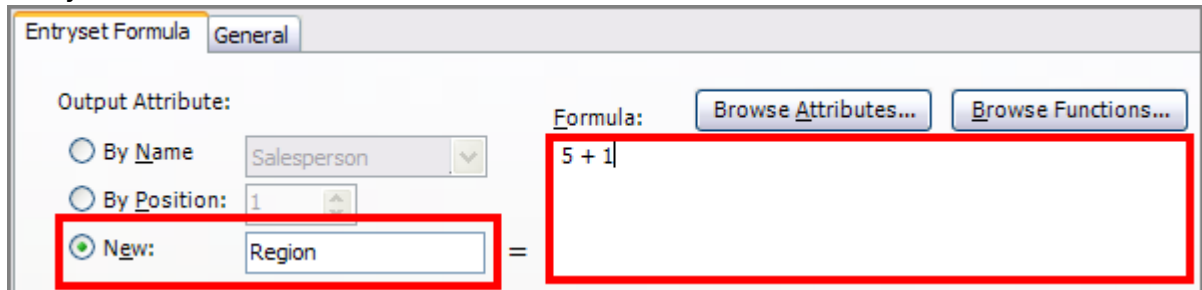
But in Proto you would have to write something like $2*(\text{variable1} + \text{variable2})$. Then you would make Connections to **variable1** and **variable2** from other sources such as a textbox, slider, other formula output, etc.

In Excel it makes sense to write the expression “ $2*(A1 + B4)$ ” because there is an inherent relationship between data and location in the spreadsheet. In Proto, all of the data relationships between modules are defined through Connections. In a Proto formula, you have to use variables in an expression like “commissions + salary,” and then make Connections to each variable afterwards in the Builder.

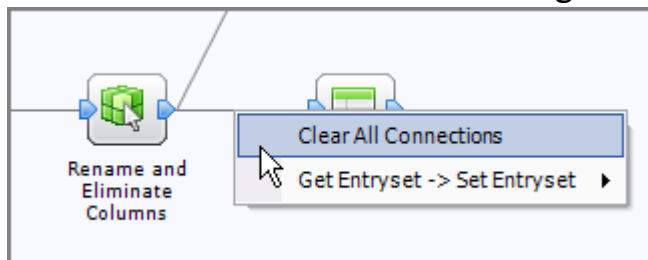
- 1 Make the following connection from the Attribute Editor to the Entryset Formula.

From:	Rename and Eliminate Columns
Output:	Attribute Editor Connectors > Get Entryset
To:	Entryset Formula 1
Input:	Entryset Formula Connectors > Set Entryset

- 2 Open the Entryset Formula’s Property Panel and configure the formula as shown below. To gain familiarity with the Entryset Formula, you will not make the VLOOKUP formula immediately. Instead you will first create a very simple formula to populate a new column named “Region” by evaluating the expression “5+1” at every row.



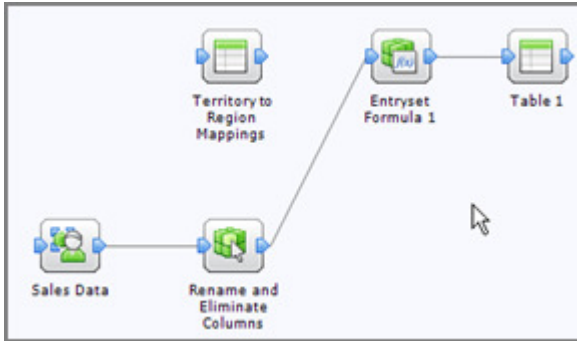
- 3 Click the Save button at the bottom of the Property Panel to save the Entryset Formula settings.
- 4 Delete the Connection between the Attribute Editor and the Table by right-clicking on the blue Connection line and selecting “Clear All Connections.”



- 5 Reposition the Table 1 module to the right of the Entryset Formula, and make the following Connection in the Builder.

From: [Entryset Formula 1](#)
Output: *Entryset Formula Connectors > Get Entryset*
To: [Table 1](#)
Input: *Table Connectors > Set Entryset*

- 6 Move Your Builder window should look similar to the screenshot below.

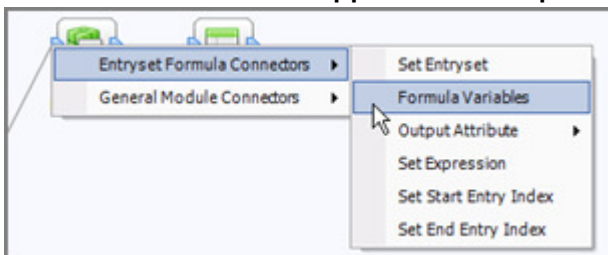


Note: We will frequently rely on using a Table control at the “end” of the chain of processing to quickly inspect and check results as we go.

- 7 Go to the Viewer and scroll in the Table control to inspect the new rightmost column named “Region” that should be filled in with the value “6.”

act Code	Revenue	Quantity	Maintenance	Total Revenue	Region
1	35000	1	7000	42000	6
2	140000	4	28000	168000	6
3	22000	11	0	22000	6
4	12000	6	0	12000	6
5	14652	148	0	14652	6
6	13175	17	0	13175	6
7	105000	3	21000	126000	6

- 8 Return to the Builder and inspect the input menu of the Entryset Formula module. Note that the menu item “Entryset Formula Connectors > Formula Variables” does not have any sub-menu items. When you add a variable to the formula expression, the variable name will appear at this input connector location.



- 9 Go back to the Entryset Formula Property Panel and change the formula to be:
VLOOKUP({Territory}, mappingsTable, 2)

The function of each input to the VLOOKUP function is summarized below. You can also click the button “Browse Functions...” to view function documentation.

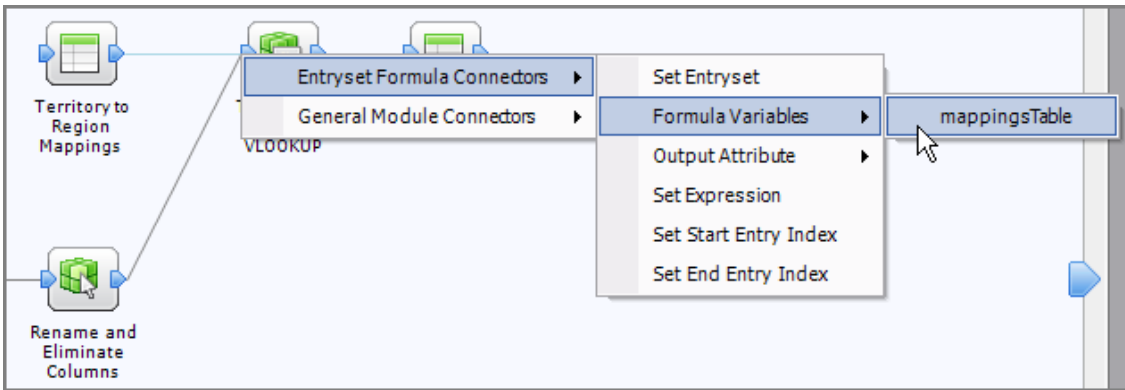
Variable	Function	Value
Lookup Value {Territory}	The value to look up in the first column of the lookup table.	The value in the “Territory” column, for the row that is currently being processed.
Lookup Table mappingsTable	An Entryset representing the lookup table.	This variable will be set through a Connection to the Entryset Formula. Its value will be the same for all rows.
Result Column 2	Specifies the column to retrieve a result value after finding a row in the Lookup Table with the Lookup Value in the first column.	The number “2” for all rows.

- 10** After entering the formula, make the “Type” of the “mappingsTable” variable equal to “Entryset.” You will do this in the “Test Values for Inputs” section of the Property Panel.



- 11** Change the label of the module to “Territory to Region VLOOKUP” in the General tab of the Property Panel.
- 12** Now click Save at the bottom of the Property Panel to save and close.
- 13** Make the following connection to the Entryset Formula.

From: [Territory to Region Mappings](#)
Output: [Table Connectors > Get Entryset](#)
To: [Territory to Region VLOOKUP](#)
Input: [Entryset Formula Connectors > Formula Variables > mappingTable](#)



Note: If you were to accidentally make the Connection to the “Entryset Formula Connectors > Set Entryset” input instead of the “mappingsTable” variable, you might be very confused by the results. The Entryset Formula would now process each row of the Entryset passed in from the “Territory to Region Mappings” Table, and not the Entryset from the Attribute Editor. **This will be true until the Entryset Formula module receives a different Entryset on its “Set Entryset” input connector. That means that even if you delete the accidental Connection and remake it correctly to the “mappingsTable” input connector, you will still not get the desired results.**

The fix is relatively straightforward. All you have to do is get the Attribute Editor to send its Entryset once again to the Entryset Formula. To do this, you can delete and remake the Connection from the Attribute Editor to the Entryset Formula, or you can simply open the Attribute Editor’s Property Panel and then click “Save” to close it. Though not relevant to this situation, triggering any event “upstream” from the Attribute Editor would suffice.

14 Now return to the Viewer and inspect the results in the Table.

	Act Code	Revenue	Quantity	Maintenance	Total Revenue	Region
14		105000	3	21000	126000	East Coast
15		16275	21	0	16275	East Coast
16		105000	3	21000	126000	East Coast
17		17825	23	0	17825	West Coast
18		105000	3	21000	126000	Central

15 If you’d like, you can experiment with the values in the Territory to Region Mappings Table to see what happens.

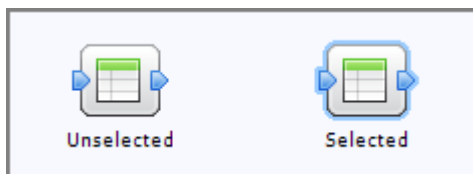
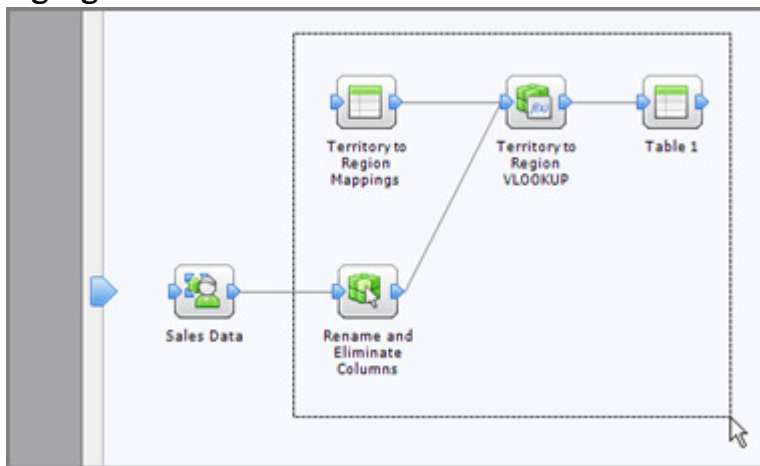
Congratulations. You have successfully used primitive modules to build some basic normalization and calculation steps with a simple user interface. In the next section, you will group the modules together to create a new Component and then export it to a folder as a Proto file.

Create a Component by Grouping Primitive Modules

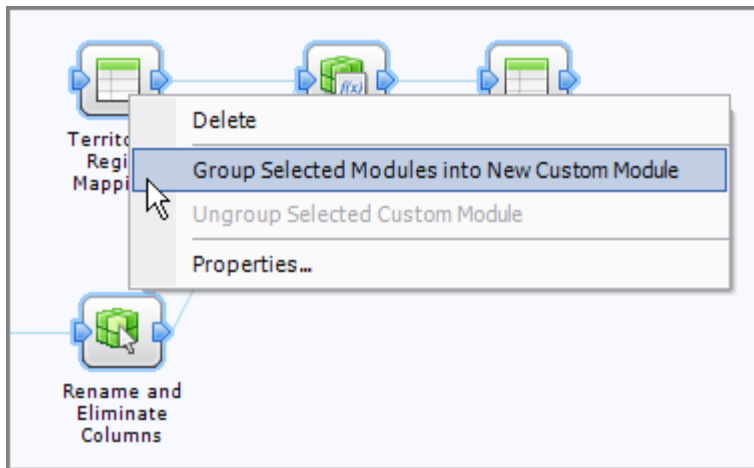
Components serve many purposes in Proto. They allow you to encapsulate, store and reuse functionality both personally and within a workgroup to avoid reinventing the wheel. And they are also required to use the workflow steps functionality in the Viewer. There are a few different ways to create components, but perhaps the most natural method is to group together related functionality “after the fact.”

In the next few steps, you will group the primitive modules on the Builder into a new Component.

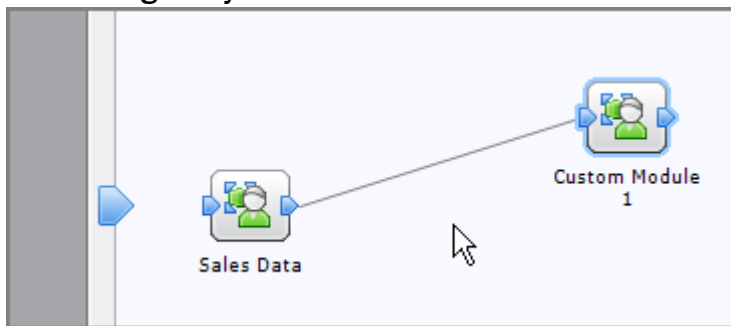
- 1 In the Builder, click and drag on the white background to select the four primitive modules. When you release the mouse, the selected modules will have a blue highlighted border.



- 2 Now right-click on any of the selected modules, being careful not to right-click on the background of the Builder instead, and select “Group Selected Modules into New Custom Module”

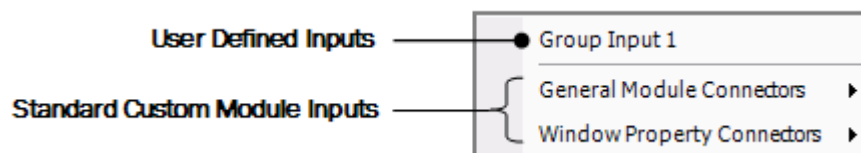


3 After doing this your Builder window will look similar to the following screenshot.



4 Rename the “Custom Module 1” as “Sales Data Normalization Steps” by double-clicking on the Component and typing in the Label field on the General tab in the Property Panel.

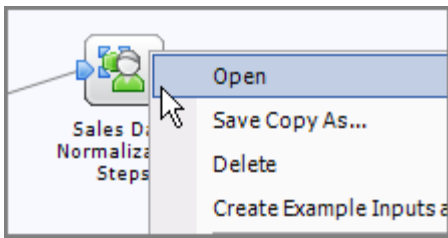
5 Inspect the inputs of the Component by clicking on the left blue arrow on the Component icon. There is one user defined input named “Group Input 1”, which was automatically created when you grouped the modules. When grouping modules into a Component, one input will be created for every Connection that must be routed into the Component.



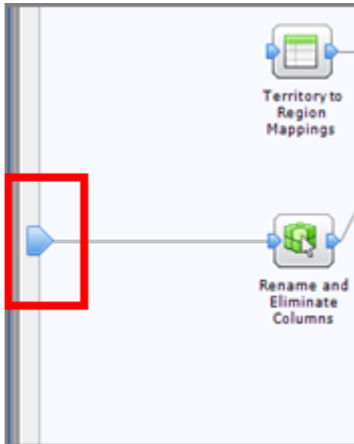
6 Now inspect the Component’s outputs by clicking on the right blue arrow on the Component icon. There are currently no user defined outputs defined for the Custom Module.



7 Now open the Sales Data Normalization Steps Component to inspect what is inside. To do this, right-click on the Sales Data Normalization Steps Component in the Builder and select “Open.”



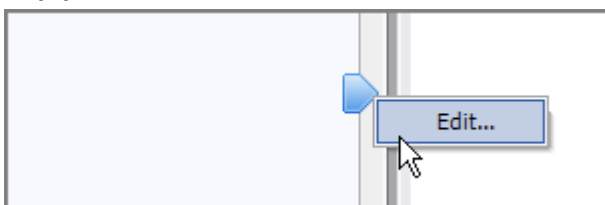
- 8 Another Builder window will be opened in the Builder area. Note that the primitive modules you assembled are inside.
- 9 Rename the user defined input “Group Input 1” by right-clicking on the blue arrow on the left hand side of the Builder area and selecting “Edit...”



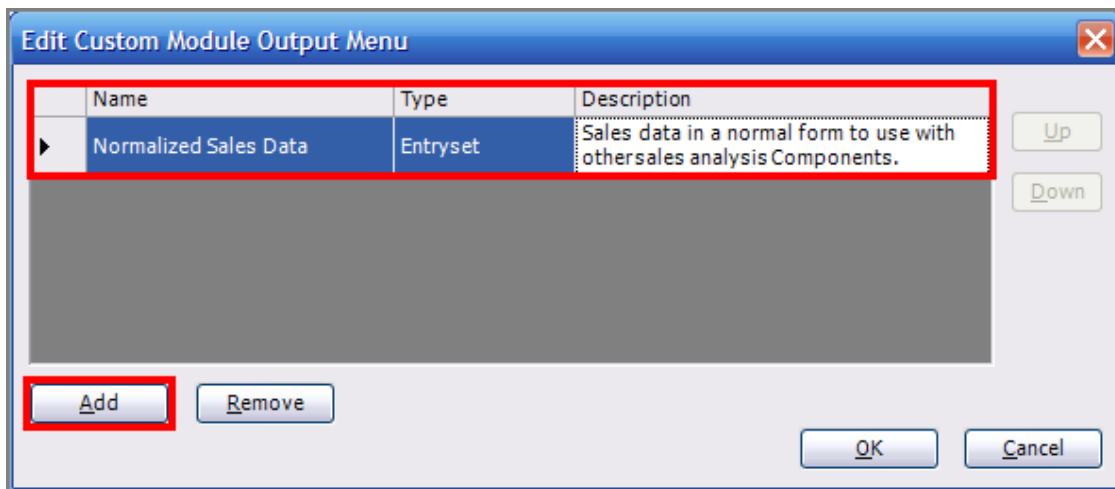
- 10 Configure the fields in the “Edit Custom Module Input Menu” panel to match the following screenshot. Then click the button titled “OK” to save the changes.

	Name	Type	Description
▶	Raw Sales Data	Entryset	Input raw sales data to be normalized.

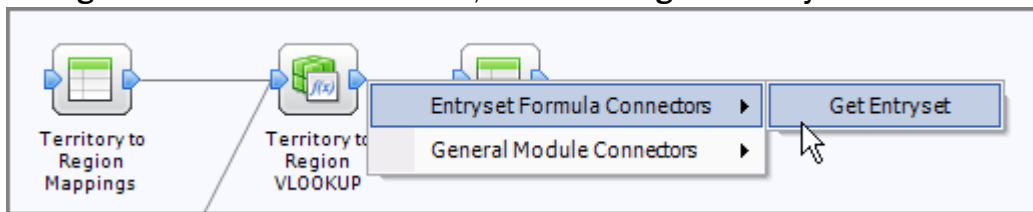
- 11 Next you will create an output for the Component to send out the processed Entryset from the Entryset Formula module. Create a user defined output by right-clicking on the blue arrow on the right hand side of the Builder area and selecting “Edit...”



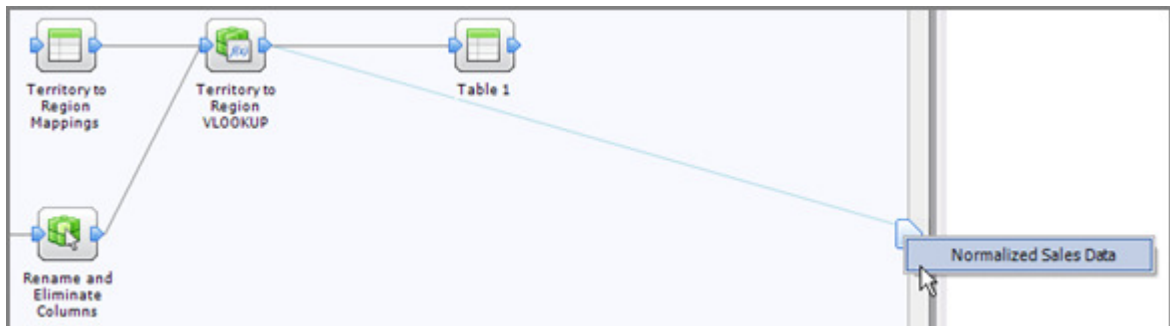
- 12 Click the “Add” button in the “Edit Custom Module Output Menu” panel, and enter the following fields. Click the “OK” button to save the changes.



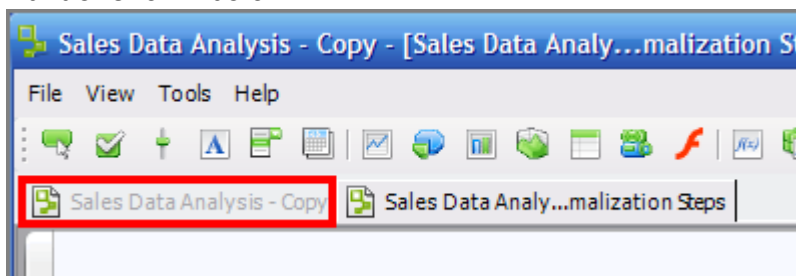
- 13** Now make a Connection from the Entryset Formula module to the Component’s output. Start the Connection in the same way you ordinarily would by clicking on the right blue arrow of the module, and selecting “Get Entryset.”



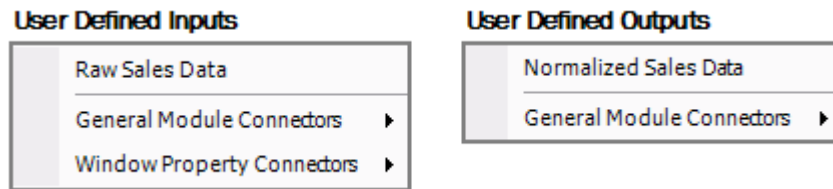
- 14** To complete the Connection, click on the blue arrow on the right side of the *Builder* window and select the menu item called “Normalized Sales Data.”



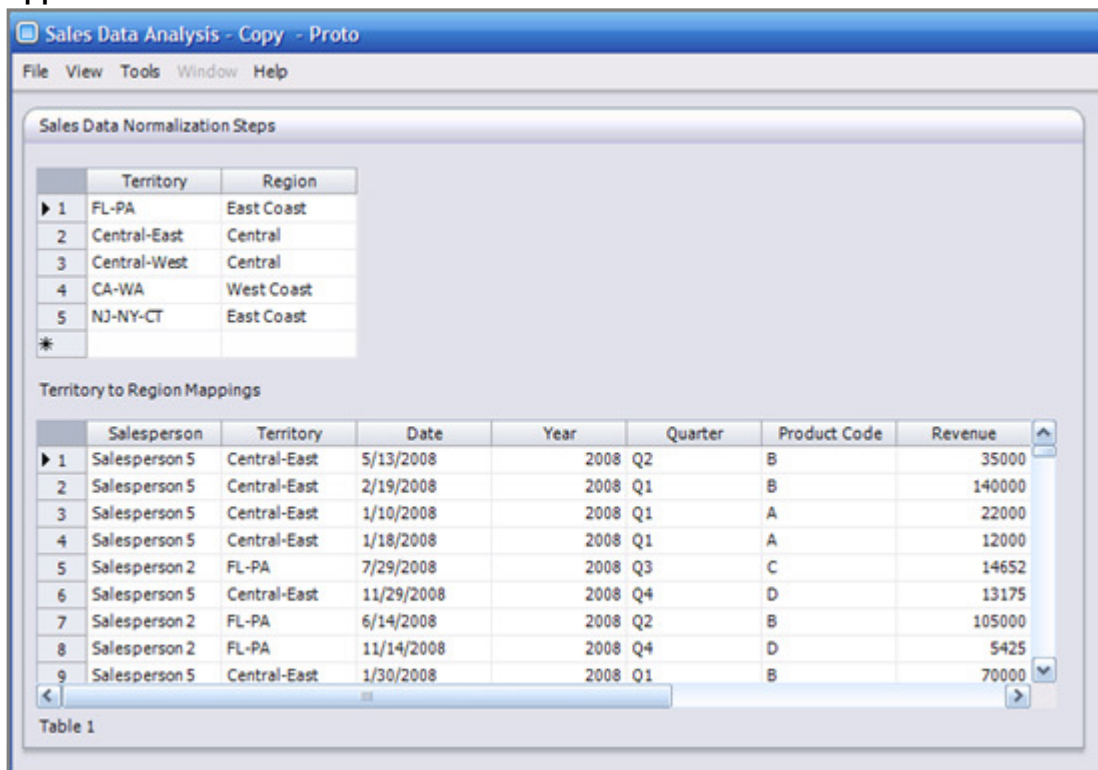
- 15** Return to the primary Builder window by clicking on the tab at the top of the Builder shown below.



- In the primary Builder window, inspect the inputs and outputs of the Sales Data Normalization Steps Component once again by clicking on the blue input and output arrows.



- Now go to the Viewer to resize and position the controls in the Component to approximate the screenshot below.



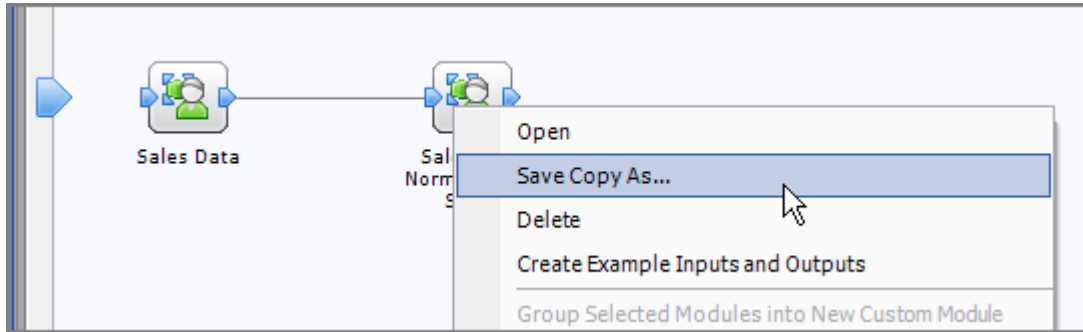
- Save your work by selecting File > Save from the application menu bar.

Export the Component to a Proto File

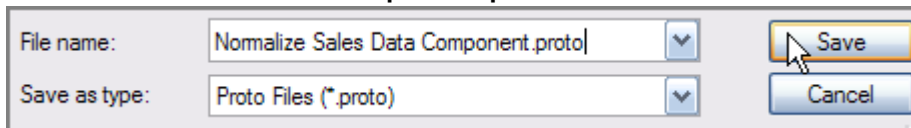
In Lesson 1, you used Components in the “Getting Started Components” directory to build a dashboard. Now you are going to export and save the Component you just created in a new Proto file.

- Go to the Builder and right-click on the Sales Data Normalization Steps Component.

- 2 Select “Save Copy As...” from the menu.



- 3 Browse to the Lesson 2 folder, and save this Component as “Normalize Sales Data Component.proto”



Note: There is no distinction in Proto files between Components and “dashboards”. You can open up any Component on its own, and you can insert any Proto file into a dashboard you are building. You can experiment with double-clicking on Components to open them up, or going to the Builder of a dashboard and inserting a “dashboard” like the Lesson 1 file you developed to see what happens.

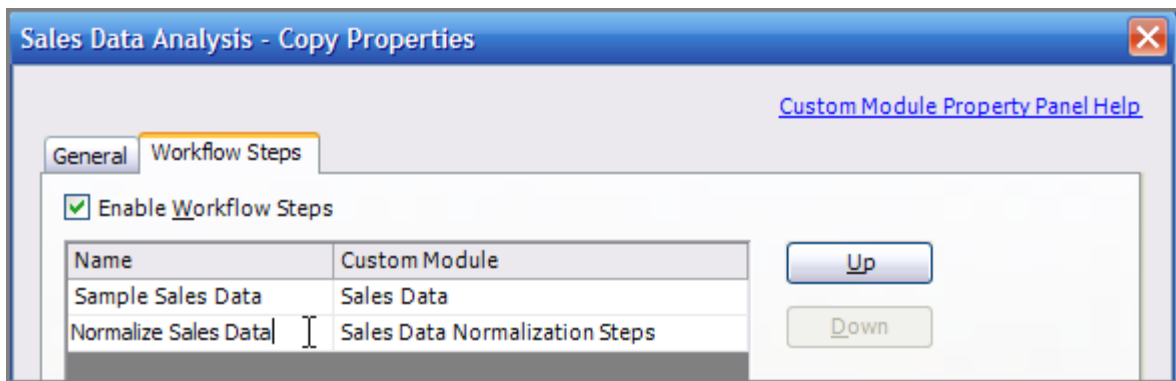
Complete the Sales Analysis Dashboard

Now that you have created a Component, you will make it into a workflow step using the techniques you learned in Lesson 1. Then you will create another empty Component to add another step to the dashboard.

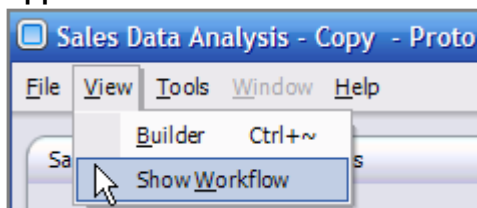
Make the Normalization Component into a Workflow Step

For a more thorough explanation of how to create and manage workflow steps, please reference Lesson 1.

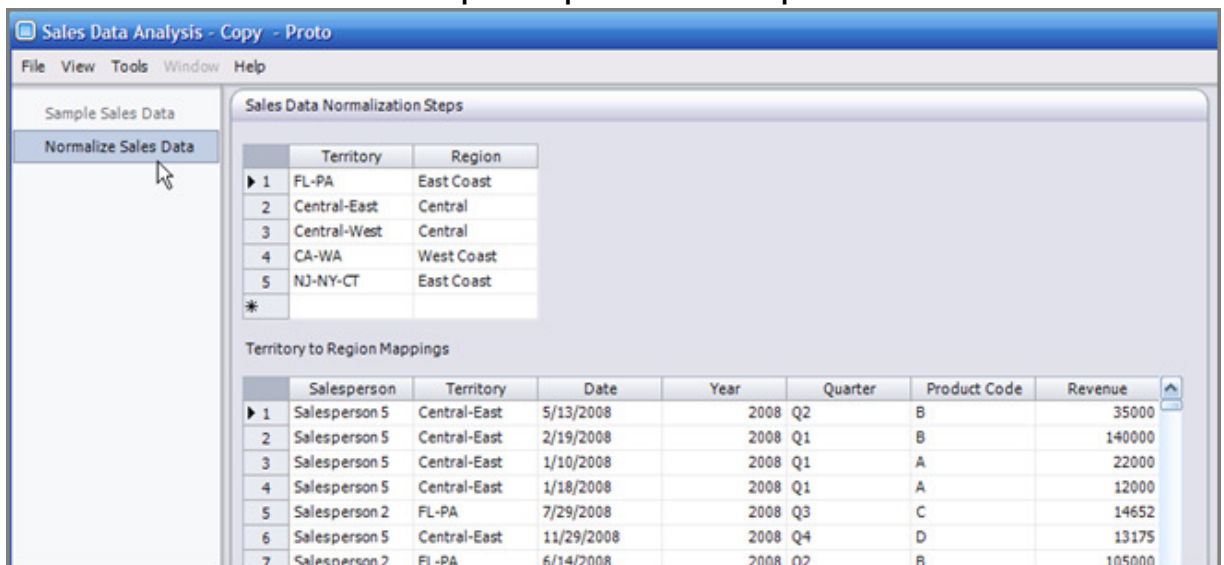
- 1 Go to the Builder and make the Sales Data Normalization Steps Component a workflow step. To do this, double-click on the background of the Builder and go to the Workflow tab in the Property Panel.
- 2 Click the “Add” button and then name the step “Normalize Sales Data” as shown below.



- Now go to the Viewer and select “Show Workflow” from the View menu in the application menu bar.



- There are now two steps in the workflow. Try clicking between them, and reposition the Sales Data Normalization Steps Component in the top left.

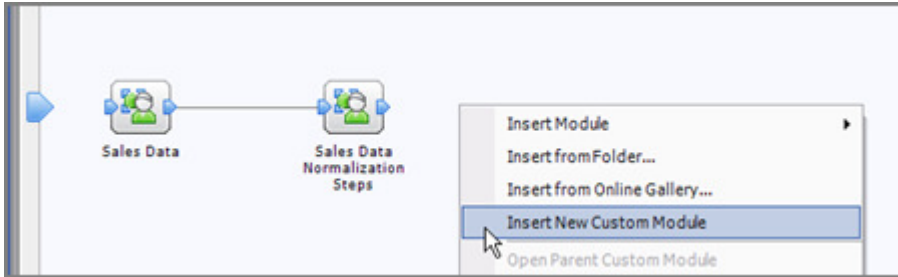


Create an Empty Component and Make it into a Workflow Step

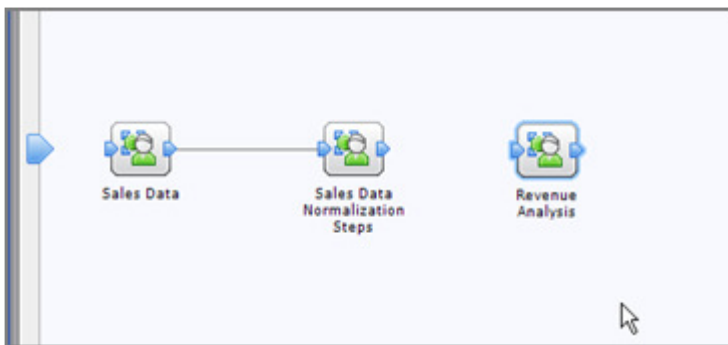
To build the Sales Data Normalization Steps Component, you assembled primitive modules and then grouped them together to form the Component. Now you will learn how to make an empty Component and immediately turn it into a workflow step. This will allow you to construct the Component without switching out of the workflow mode

in the Viewer. The steps to add and incorporate this new Component will review techniques from both Lesson 1 and earlier in Lesson 2.

- 5 Go to the Builder and right-click next to the Sales Data Normalization Steps Component as if you were going to insert a Component from a folder or primitive module. Instead, select “Insert New Custom Module” from the menu.



- 6 A new Component called “Custom Module 2” will be added to the Builder. Go to the Property Panel of the new Component by double-clicking on it, and change its label to “Revenue Analysis” under the General tab. Your Builder window will look similar to the screenshot below.



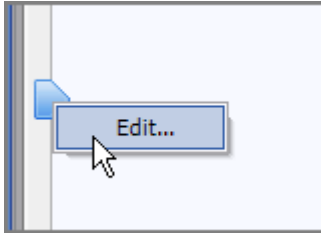
- 7 Make the Revenue Analysis Component into a workflow step, named “Revenue Analysis” as you have done in Lesson 1 and Lesson 2. To do this, double-click on the background of the primary builder window, go to the Workflow Steps tab, click the Add button and name the new workflow step.
- 8 Go to the Viewer to position and size the empty Revenue Analysis Component. Make it fairly large in the Viewer so you have room to work when you start to add controls.

Connect the Revenue Analysis Component

The Revenue Analysis Component appears as a workflow step, but it is empty and does not yet have the normalized sales data to analyze.

- 1 Return to the Builder to edit the inputs and outputs of the Component so that you can send in the normalized Entryset that was processed in the Sales Data Normalization Steps Component.

- 2 In the Builder, inspect the inputs and outputs of the Revenue Analysis Component. Note that two default inputs and outputs were created. While you can use these inputs and outputs directly, it is highly recommended that you create meaningful input and output names for Components.
- 3 Following similar steps to those on pages 18-20 in this lesson, open the Builder window for the Revenue Analysis Component, right-click on the blue input arrow on the left side of the Builder and select “Edit...”



- 4 Select “Input 2” and click the “Remove” button to delete this input.
- 5 Configure “Input 1” as follows.

	Name	Type	Description
▶	Normalized Sales Data	Entryset	Input normalized sales data to analyze.

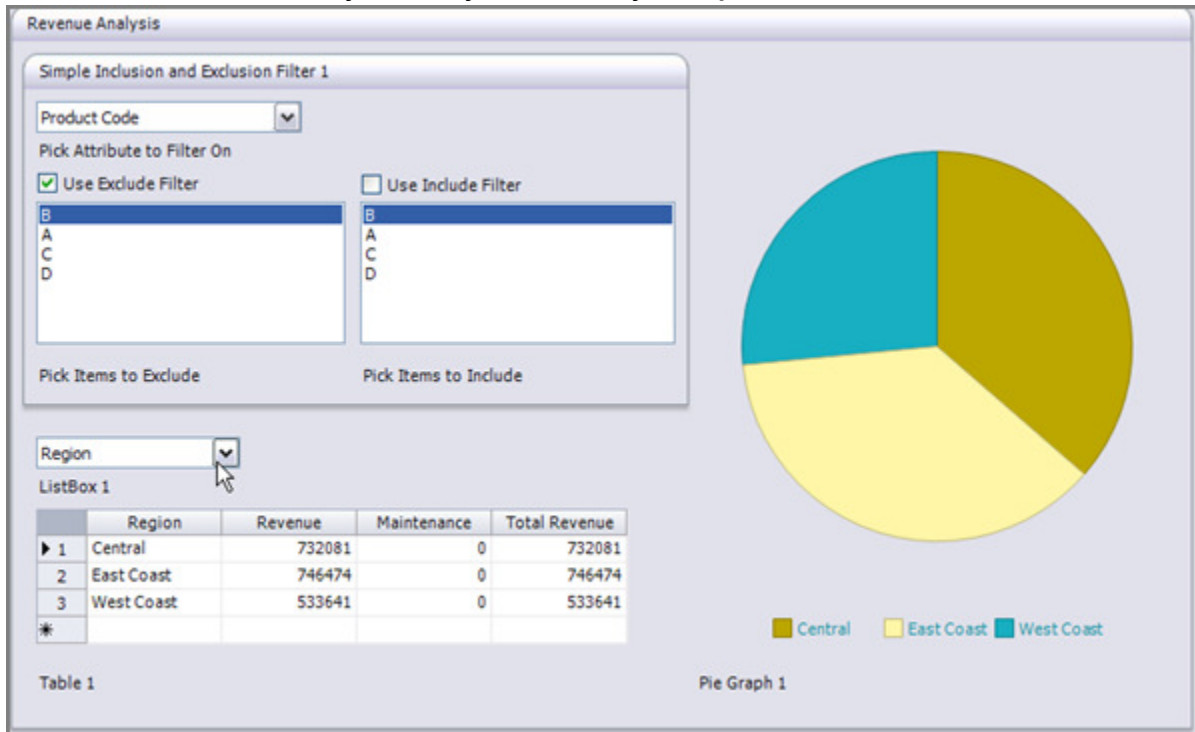
- 6 Click “OK” to save and close the Edit Custom Module Input Menu panel.
- 7 Now return to the primary Builder window by clicking on the tabs at the top of the Builder. Make the following Connection.

From: [Sales Data Normalization Steps](#)
Output: *Normalized Sales Data*
To: [Revenue Analysis](#)
Input: *Normalized Sales Data*

- 8 Now the Revenue Analysis Component is part of the workflow, and within its Builder window you can add primitive modules and other Components to assemble functionality.

Create a Simple Revenue Analysis Report

You will now create a very basic dynamic analysis report as shown below.



This report shows aggregate revenue, maintenance contracts and total revenue in a Table control, grouped by a category dynamically selected in the Viewer. It also allows for the selective inclusion and exclusion of sales data. With this simple arrangement, you can ask dozens of revenue-oriented questions such as, “How much revenue was booked per Region, excluding product B?”

- 1 Go to the Builder window of the Revenue Analysis Component, which is empty.
- 2 Insert the following Component, as you did in Lesson 1, by right-clicking on the background of the Builder window and selecting “Insert from Folder...” from the menu.

Location: ..\Getting Started Components\Do Stuff with Data\Sort and Filter Data

Component: Simple Inclusion and Exclusion Filter.proto

- 3 Now insert four more primitive modules into the Revenue Analysis Component you just created. You can use the toolbar at the top of the Builder or right-click on the background of the Builder and select “Insert Module > [Module Name]” as you did earlier in Lesson 2. Insert these modules:

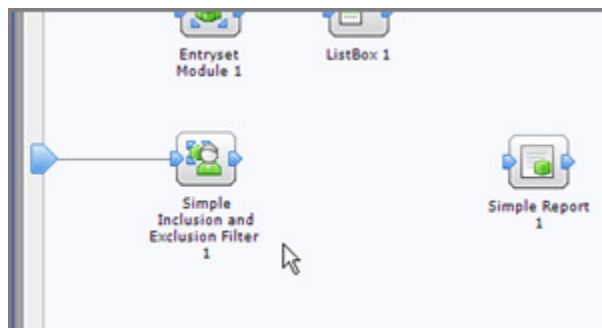
- Entryset Module
- Listbox
- Simple Report
- Table

- 4 Arrange the primitive modules and filtering Component on your Builder window as shown in the following screenshot.



- 5 First you will connect to the filtering Component.

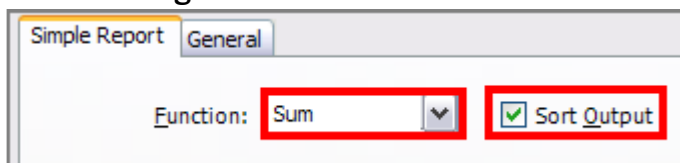
From: Builder Inputs
Output: Normalized Sales Data
To: Simple Inclusion and Exclusion Filter 1
Input: Set Entryset

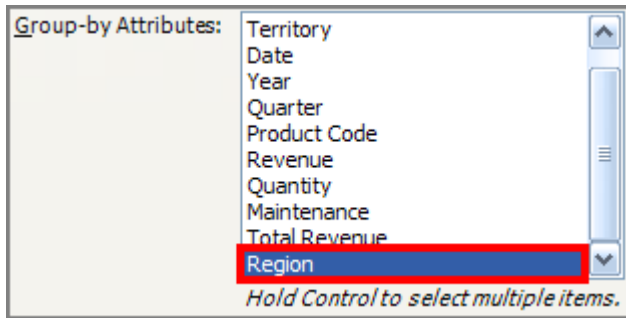
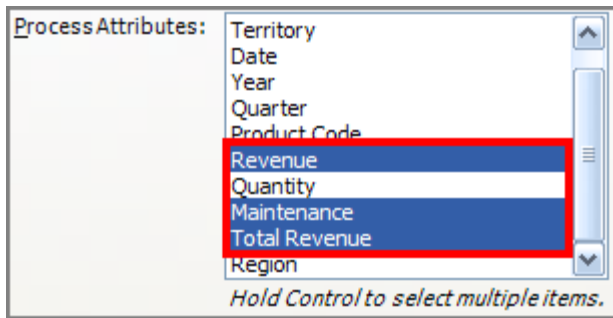


- 6 Now connect the filter Component to the Simple Report module. Note that this will cause the Simple Report to error.

From: Simple Inclusion and Exclusion Filter 1
Output: Get Filtered Entryset
To: Simple Report 1
Input: Simple Report Connectors > Set Entryset

- 7 To configure the Simple Report, go to its Property Panel and configure it as shown below. To select multiple items in the Listbox controls, hold down the Control key while clicking with the mouse.



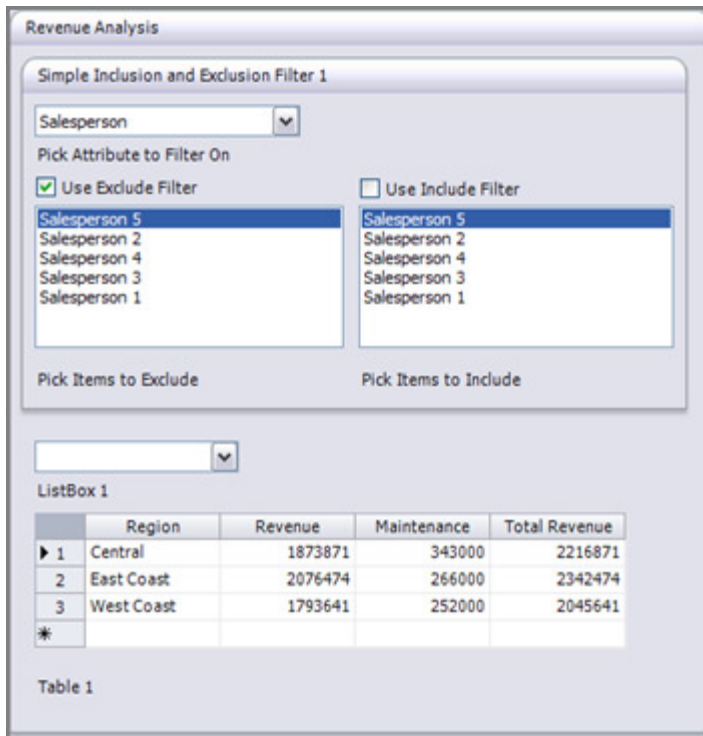


8 Click the “Save” button to close the Property Panel.

9 Make the following Connection

From:	Simple Report 1
Output:	Simple Report Connectors > Get Report Entryset
To:	Table 1
Input:	Table Connectors > Set Entryset

10 Go to the Viewer and arrange the controls as shown below. Experiment with the filter Component settings and note that the Table updates.



11 Return to the Builder and make three more Connections.

First you will connect the input sales data Entryset to the Entryset Module. One of the outputs of the Entryset Module is a new Entryset with one column, containing the Attribute names of the input Entryset. You will use the Attribute names Entryset as an input to the Listbox to set the selection choices. Finally the selection in the Listbox will set what Attribute to “Group By” in the Simple Report, thereby changing how the data is aggregated.

From: [Builder](#)
Output: *Normalized Sales Data*
To: [Entryset Module 1](#)
Input: *Entryset Module Connectors > Set Entryset*

From: [Entryset Module 1](#)
Output: *Entryset Module Connectors > Get Attribute List*
To: [ListBox 1](#)
Input: *ListBox Connectors > Set Entryset*

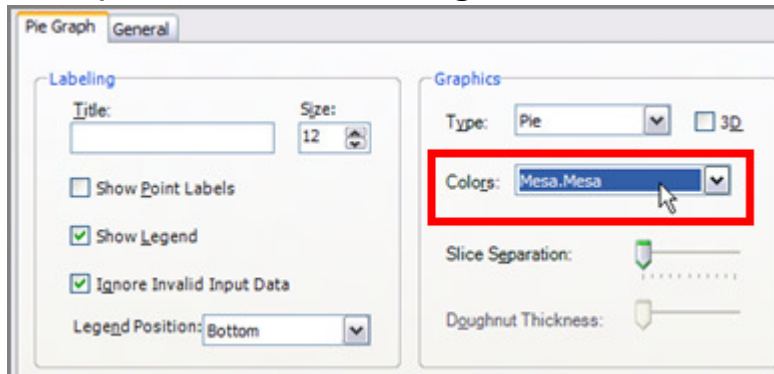
From: [ListBox 1](#)
Output: *ListBox Connectors > Get Selected Item*
To: [Simple Report 1](#)
Input: *Simple Report Connectors > Set Group-By Attribute*

12 Return to the Viewer to see that the Listbox is populated with the names of the sales data columns. Change the Listbox selection to update the revenue report.

- 13** As a final step, insert a Pie Graph primitive module and make the following Connection.

From: Simple Report 1
Output: Simple Report Connectors > Get Report Entryset
To: Pie Graph 1
Input: Pie Graph Connectors > Set Entryset

- 14** Go to the Pie Graph module's Property Panel to select a color scheme that you like and experiment with other settings, then click the Save button.



- 15** Return to the Viewer to resize and position controls as shown at the beginning of this section. Then experiment with the settings to answer the question, “How much revenue was booked per Region, excluding product B?”

- 16** Save your work by selecting File > Save from the application menu bar.

Thank You!

Thank you for your interest in Proto. If you have feedback regarding this tutorial or would like to see more training material covering specific topics or questions, please email us with suggestions at devcenter@protosw.com or visit the forums at <http://www.protosw.com/devcenter>.