

Creating Reports in SAS® Visual Analytics Designer that Dynamically Substitute Graph Roles on the Fly Using Parameterized Expressions

Kenny Lui, SAS Institute Inc.

ABSTRACT

With the expansive new features in SAS® Visual Analytics 7.1, you can now take control of the graph data while viewing a report. Using parameterized expressions, calculated items, custom categories, and prompt controls, you can now change the measures or categories on a graph from a mobile device or web viewer. View your data from different perspectives while using the same graph. This paper demonstrates how you can use these features in SAS® Visual Analytics Designer to create reports in which graph roles can be dynamically changed with the click of a button.

INTRODUCTION

The ability to manipulate your data in order to view exactly what you need is one feature available to report designers in SAS® Visual Analytics. SAS® Visual Analytics Designer and SAS® Visual Analytics Explorer provide you with full access to the data so that you can drag measures and categories onto the graph to see how the data has changed.

In order to get that same information to your users, you can create reports with multiple graphs and different combinations of data roles that display the most relevant information. Your users can then view the report on SAS® Visual Analytics Viewer or on a tablet with the SAS® Mobile BI application.

However, it is much easier if the user could select exactly what data they want to see. Instead of sifting through multiple sections that are crowded with various graphs to find what they need, they can use a prompt control to change the X-Axis and Y-Axis value of a single bar chart. For example, this bar chart might show expenses by month and then get transformed to show revenue by year with a few clicks of a button.

Parameters, available in SAS® Visual Analytics Designer 7.1, allow you to create reports with a single graph that can swap the roles on the fly. You are able to control what categories and measures they have access to and you no longer need to create multiple graphs to display the various combinations of roles.

AN INTRODUCTION TO PARAMETERS

In previous versions of SAS® Visual Analytics, interactions were based directly on the associated category or measure. If you filtered (or brushed) on a category like month, it would filter directly on the month that was selected. If you filtered on a measure like quantity, it would filter all values that had that quantity.

With parameters, you can still filter on the associated category or measure. However, the parameter enables you more flexibility on how to respond to the selected value. For example, if the month May is selected in a prompt control, instead of filtering on May, you can choose to set the filter to apply from January to May.

Parameters also enable you to set dynamic display rules and ranks. For ranks, you can move a slider to 5 and see the top 5 ranked values in your list table. If you moved that slider to 10, the list table would update to show the top 10 ranked values. The same can be done with display rules. You can highlight certain values based on a dynamic value that the user controls. Parameters have opened up a lot of options to create more interactive reports. There are many other applications to parameters, but this paper covers using parameters to change graph roles.

The best way to explain how this is done is by going through an example. The report is created in SAS® Visual Analytics Designer 7.1. This is a basic example that allows you to create a button bar that swaps between 3 measures. The measures used here are *Profit*, *Revenue*, and *Expenses*. The Y-Axis measure of the bar chart correctly changes to what is selected in the button bar.

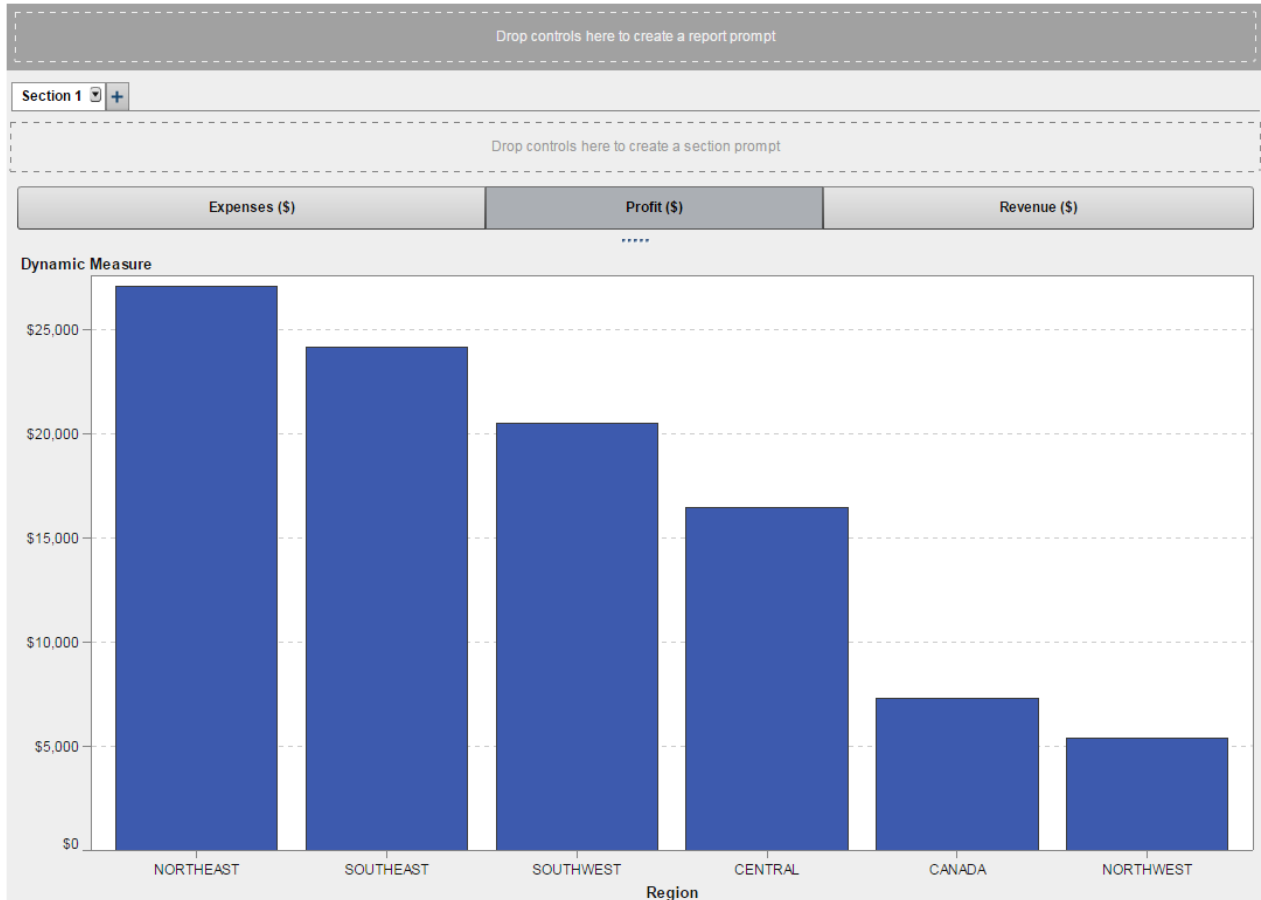


Figure 1: The Completed Example

CREATING A CUSTOM CATEGORY

A custom category is needed to control which measure or category you want to use in your visualization. You will create labels that enable your user to know which category or measure the prompt control will switch to. The actual category that is used to create your custom category is not that important. However, the custom category should be based on a category that has a minimum cardinality equal to the number of categories or measures that you want to switch between. For this example, you switch between *Profit*, *Expenses*, and *Revenue*. You need to base the custom category on a category that contains 3 or more unique variables.

Start by connecting to an instance of SAS® Visual Analytics Designer 7.1 and creating a new report. From the **Data** tab, load in the data source that you want to use.

1. To create the custom category, right-click on the category you want to use and select **New Custom Category**. You should see the list of categories in your **Data** tab.
2. In the **Name:** field, enter *MeasureSelection*.
3. Add two more labels to the Label Heading field by clicking the **New label** button twice. You should now have 3 labels. Enter the name of your measures or categories in the Label heading field by double-clicking a label name. For this example, name your three labels "*Profit (\$)*", "*Expenses (\$)*", and "*Revenue (\$)*".
4. After that is done, drag some categories into each label. Because this custom category is being used only for the labels, the exact categories that you drag into each label is irrelevant. Ensure that all of the category values are used so that you do not end up with the *Other* option on your control. Do not leave any label without a value. Click **OK**. The custom category is now complete.

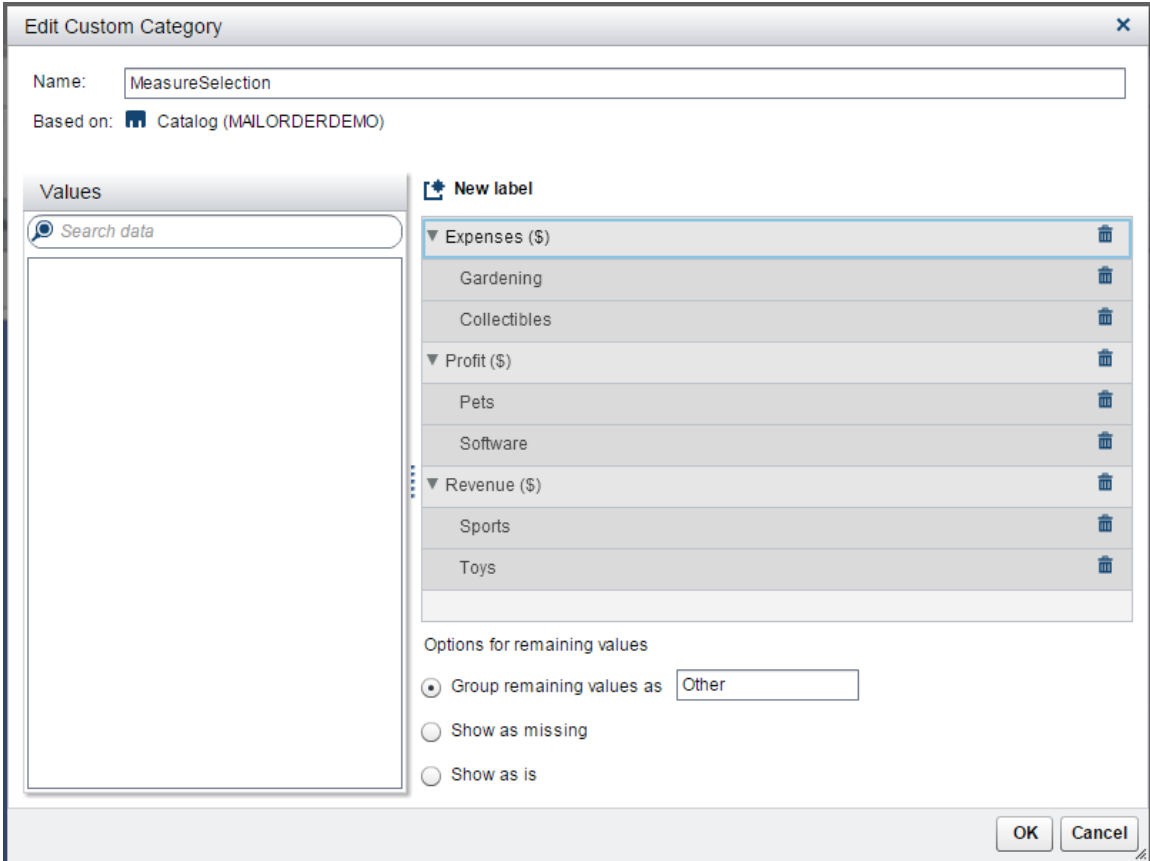


Figure 2: The Custom Category Containing 6 Values

CREATING A CONTROL

With the custom category created, you can now use this category in a prompt control. A drop-down list, list, or button bar are good controls to use. If you choose a list, ensure that only a single selection can be made. You should make the control **Required** so that a user cannot get into a situation where they have no measure or category selected.

1. For this example, drag a button bar onto the report. The button bar is found in the **Objects** tab. Ensure that this is placed in the object level and not the section level or report level. Doing this incorrectly will cause the values that you dragged into each label of your custom category to apply a filter that you do not intend to use.
2. Drag the custom category onto the button bar so that your labels now appear on the button bar.

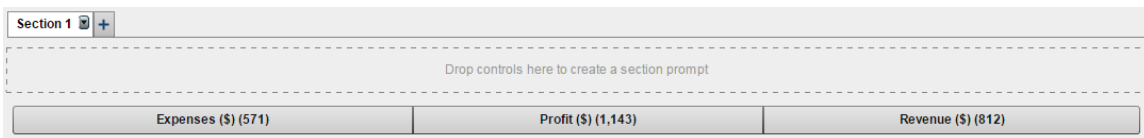


Figure 3: The Button Bar Now Displays Your Labels, Sizing Can Vary

3. You should remove Frequency from the prompt control as the frequency numbers are irrelevant. They are associated with the values that you dragged into each label. Right-click on the prompt, place your mouse pointer over **Remove**, and select **Frequency**.
4. Select the Button Bar if it is not already selected. On the **Properties** tab, check the **Required** option. This makes it clear to the user what measure is selected and displayed on your graph. It also prevents the situation where no measure is displayed on your graph.

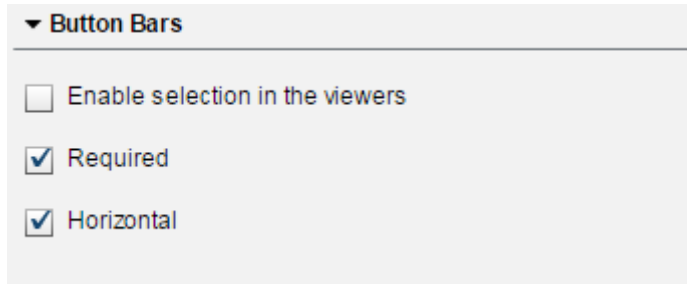


Figure 4: This Step Makes It Impossible to Have a Button Bar with No Selection

CREATING A PARAMETER

Now that the control is present, define a parameter so that you can use it in the calculated item. This enables the calculated item to know what value a user selects.

1. Right-click on the button bar and select **Create Parameter**.
2. Enter an appropriate **Name** for the parameter. The default parameter name (*MeasureSelection Parameter*) is fine for this example.
3. The **Type** should be **Character**. This parameter helps the calculated item determine what is selected. It does not matter if you are switching between a category or a measure, **Type** will always be **Character**. You can leave **Current value** empty because the control was set to **Required** and will always have a value. When completed, click **OK**.

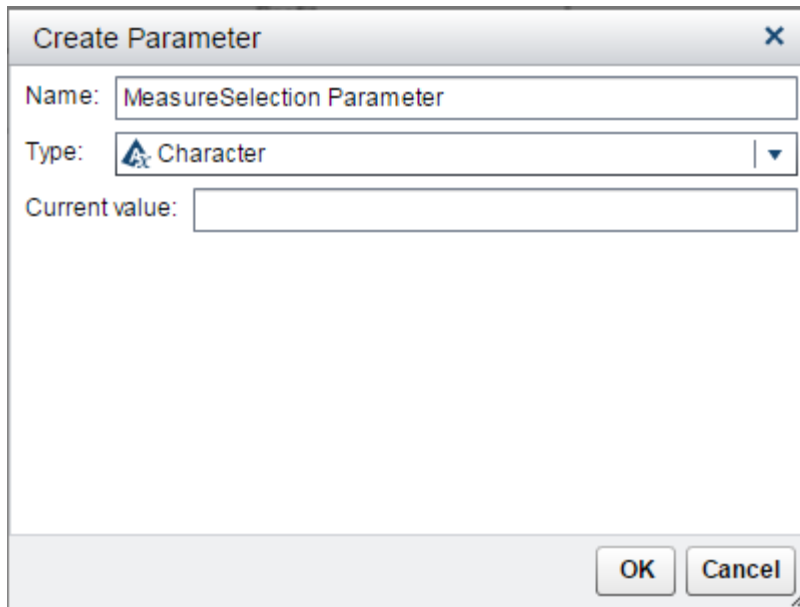


Figure 5: The Create Parameter Window

CREATING A CALCULATED ITEM

Most of the major setup for this example is done in the calculated item. This is the measure or category that you want to use on the graph or visualization. You need to define an expression that returns the correct variable based on what is selected in the prompt control and passed via the parameter.

For this example, the general logic of the expression checks if the value in the button bar contains "Expenses (\$)". If it does, it returns the *Expenses* measure. If it does not, the expression then checks if

the value in the button bar contains “Profit (\$)”. If it does, it will return the *Profit* measure. Otherwise, the *Revenue* measure is returned. Defaulting to *Revenue* ensures that a measure is always displayed in the graph.

1. Click the Options drop-down menu () in the **Data** tab and select **New Calculated Item**.
2. Provide an appropriate name for the calculated item. This is the name that displays on your graph or visualization. In this example, this name displays on the Y-Axis and does not change when the parameter is changed. Name this calculated item *Dynamic Measure*.
3. If you are swapping between measures, the **Result Type**: should be **Numeric**. If you are trying to change categories on the fly, use **Character**. It is also possible to switch between date values. If you do this, select the appropriate date type.
4. The expression part is when it gets a little tricky. You need an **IF...ELSE** to allow you to switch between measures. Depending on how many options you are switching between, you might need several nested **IF...ELSE** operators.

In this example, use two **IF...ELSE** operators. These operators will check to see if *Expenses* or *Profit* are selected. If neither are selected, the expression defaults to *Revenue*. Because of this, only two **IF...ELSE** operators are needed to switch between three options (*Expenses*, *Profit*, and *Revenue*).

From the **Operators** pane, expand **Boolean**, and drag an **IF...ELSE** onto the **Visual** pane. Drag a second **IF...ELSE** and place it inside the ELSE block of the first **IF...ELSE** on the **Visual** tab.

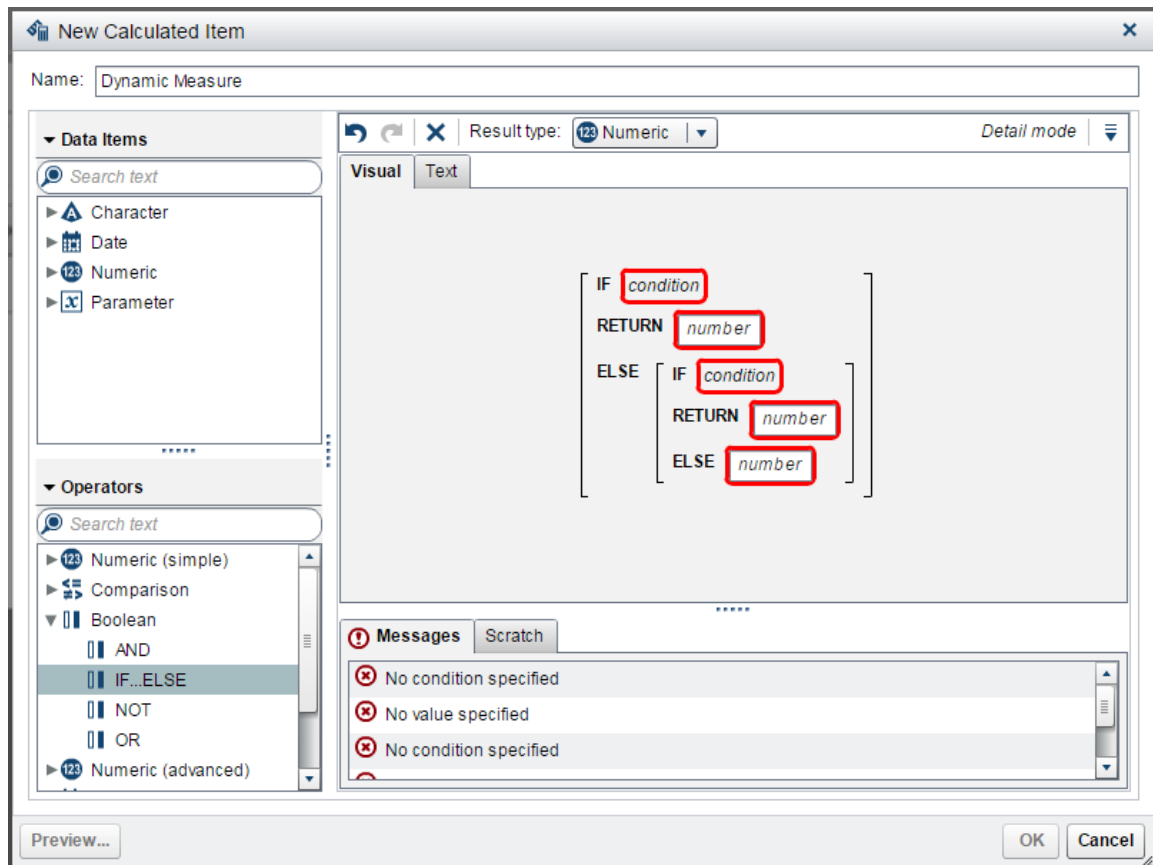


Figure 6: If Your Operators Are Placed Correctly, the Visual Pane Should Resemble This

5. The condition lets the calculated item know what is selected in the prompt control. There are several ways to identify what has been selected in the prompt control. For this example, use the **Contains** operator. The **Contains** operator checks that the parameter value contains the value that you set in the custom category label.

Under the **Operators** pane, expand **Text (simple)**. Drag a **Contains** operator and place it in the **IF** condition block. Repeat this for each **IF** statement.

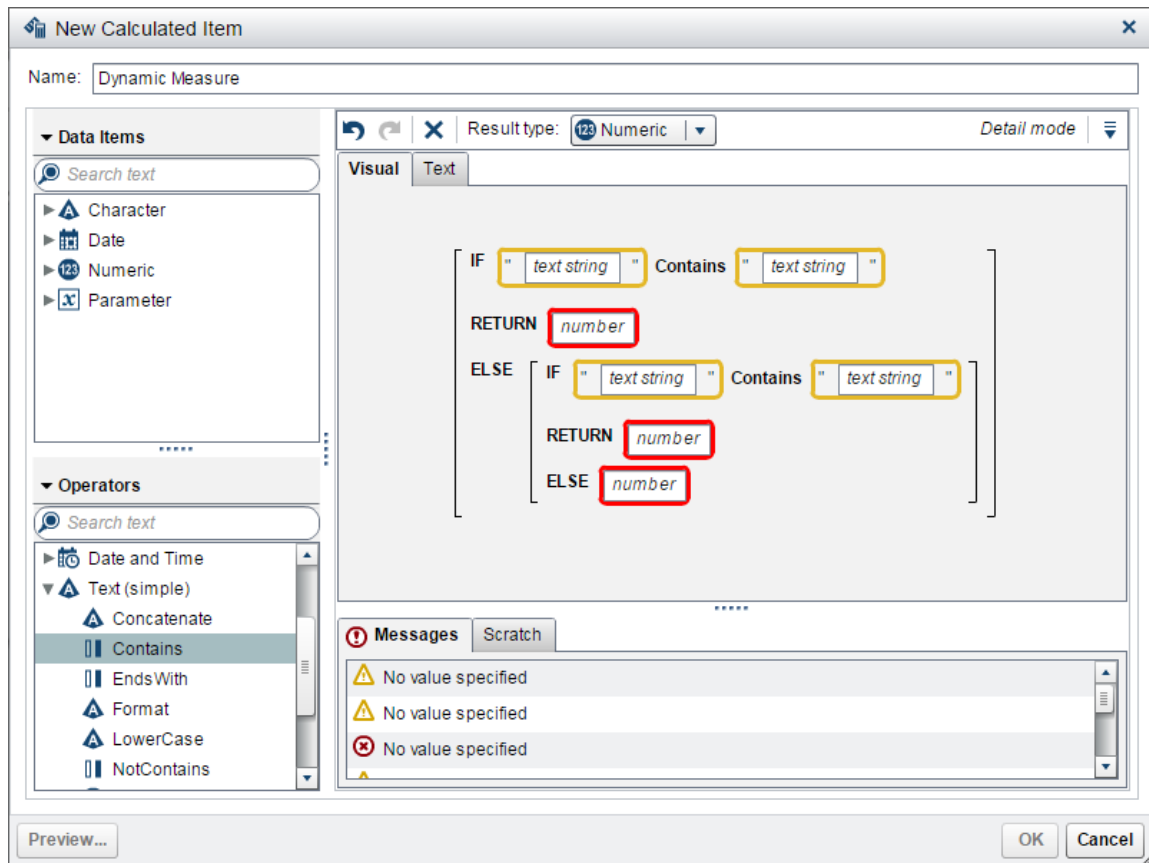


Figure 7: The Contains Operators Are Now Placed inside of the IF Operators

6. Now, define the **Contains** conditions. Drag the parameter that you created to the first part of the **Contains** text string area.

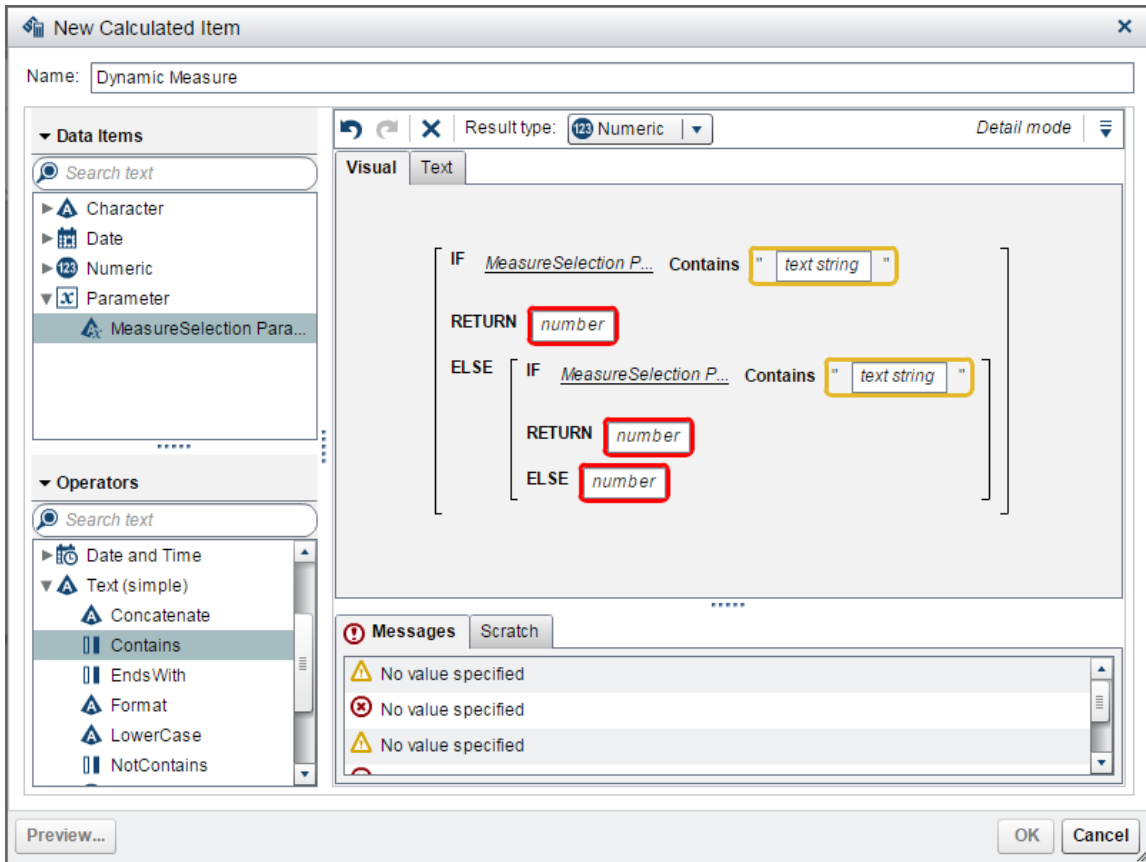


Figure 8: The Parameter Has Been Added to the Front of the Contains Operators

- For the second part of the **Contains** condition, enter the label values that you entered for the custom category. The **Contains** operator is straightforward and works with partial matches. You might not need to enter the complete label value, but it is safer to do so because your custom category labels will not appear in the suggestions drop-down list. The drop-down suggestions might attempt to match variable names. Do not select these.

Repeat this for each label that you created in the custom category. Because this example defaults to *Revenue* if neither *Expenses* nor *Profit* are not selected, you do not need to enter a label for “Revenue (\$)”.

Note: It is important that you do not use a variable for the **Contains** operator.

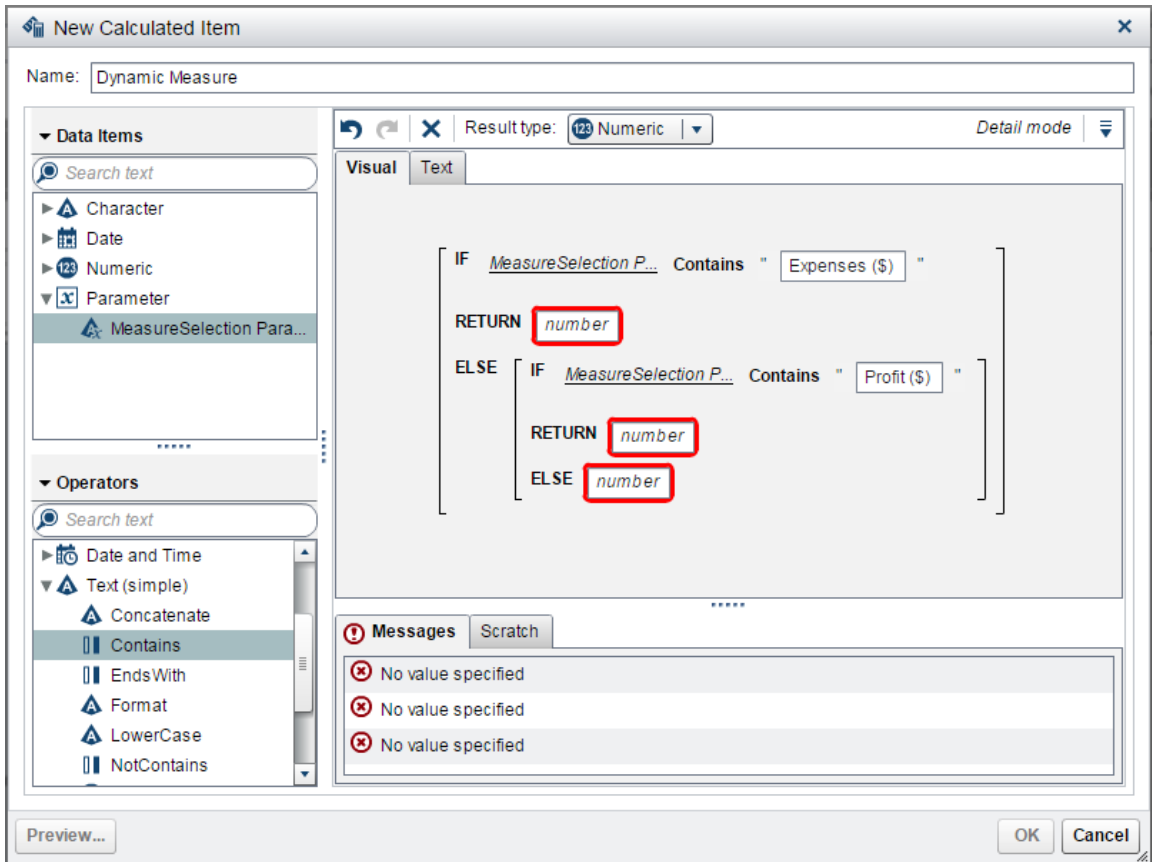


Figure 9: The Contains Operators Are Now Complete

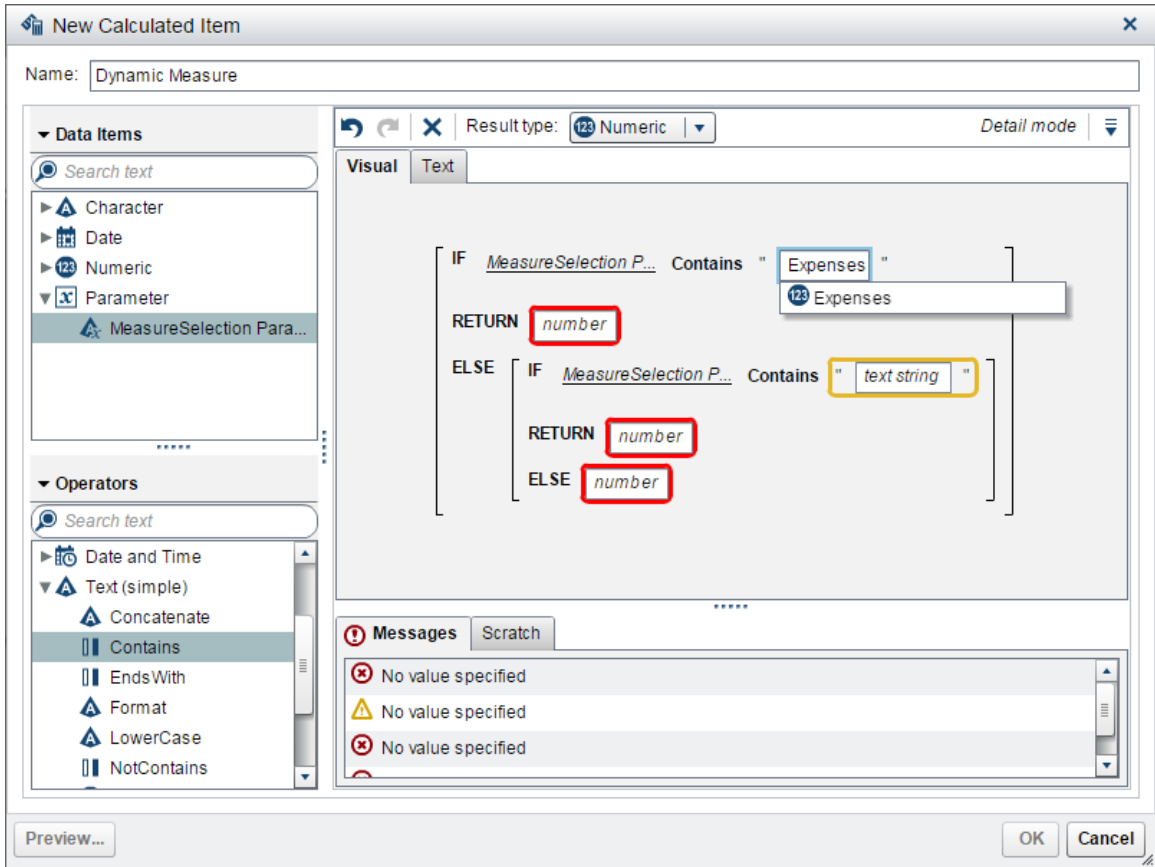


Figure 10: Do NOT Select That Variable from the Suggested Drop-down List

8. Now that the conditions are set, you need to specify what the calculated item will do when the condition is met. The value that is returned should correspond with your IF statement. Because this example returns numerics, you should enter the correct numeric measure. If this was a category that you were switching between, a category should be returned.

From the **Data Items** field, drag *Expenses* into the **RETURN** block that corresponds to the first **IF** condition. Drag *Profit* into the **RETURN** block that corresponds to the second **IF** condition. For *Revenue*, place it in the **ELSE** block so that it returns *Revenue* in the case where *Expenses* and *Profit* are not selected.

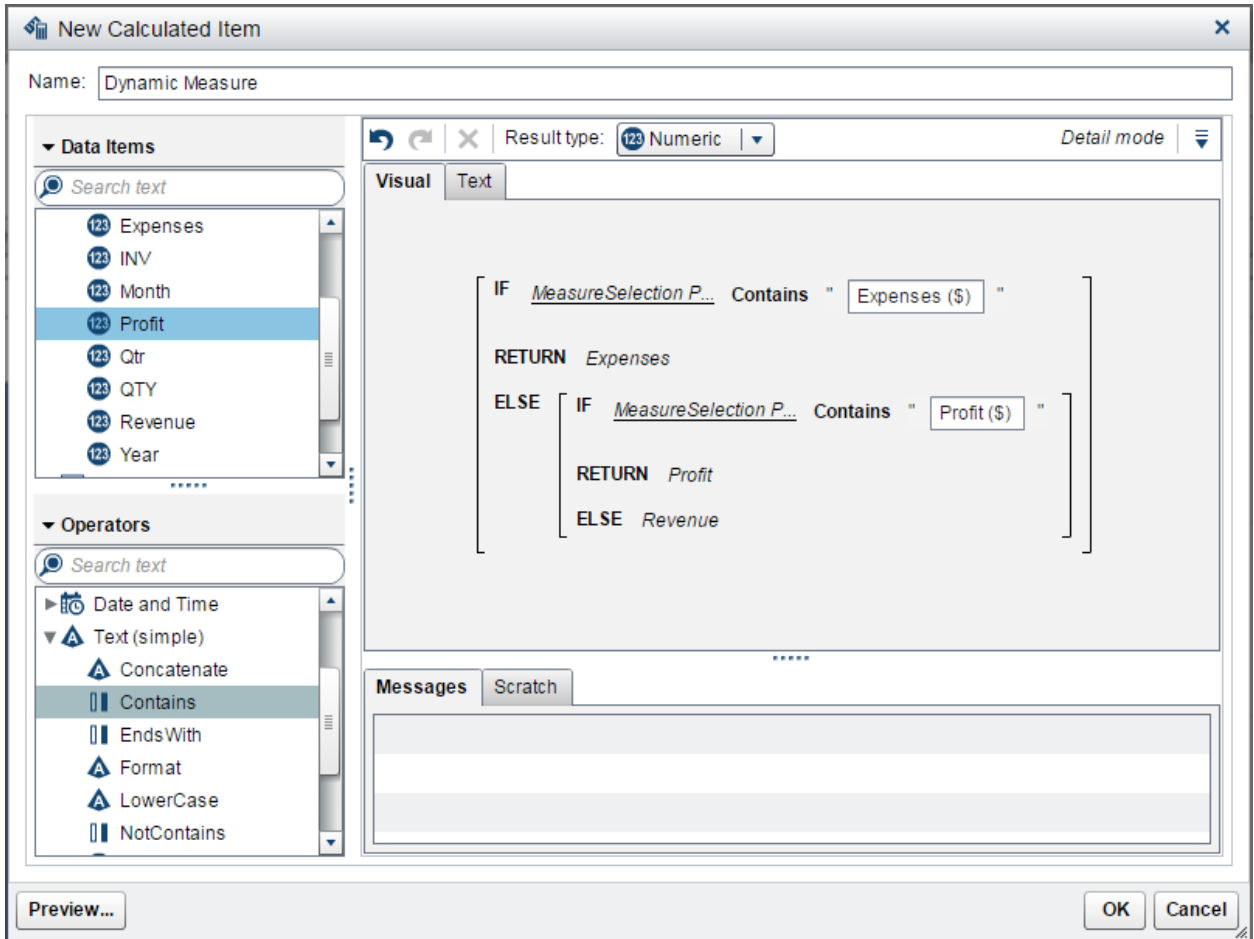


Figure 11: The Completed Calculated Item

9. With the calculated item complete, click **OK** to save this item.

PUTTING IT ALL TOGETHER

Everything is now correctly configured and you can use your dynamic measure. You are able to use this calculated item in a variety of ways, but this example uses a bar chart.

1. Drag a bar chart from the **Objects** tab and place it below the button bar.
2. From the **Data** tab, drag any category onto the bar chart.
3. Drag the calculated item you created (*Dynamic Measure*) from the **Data** tab onto your bar chart.

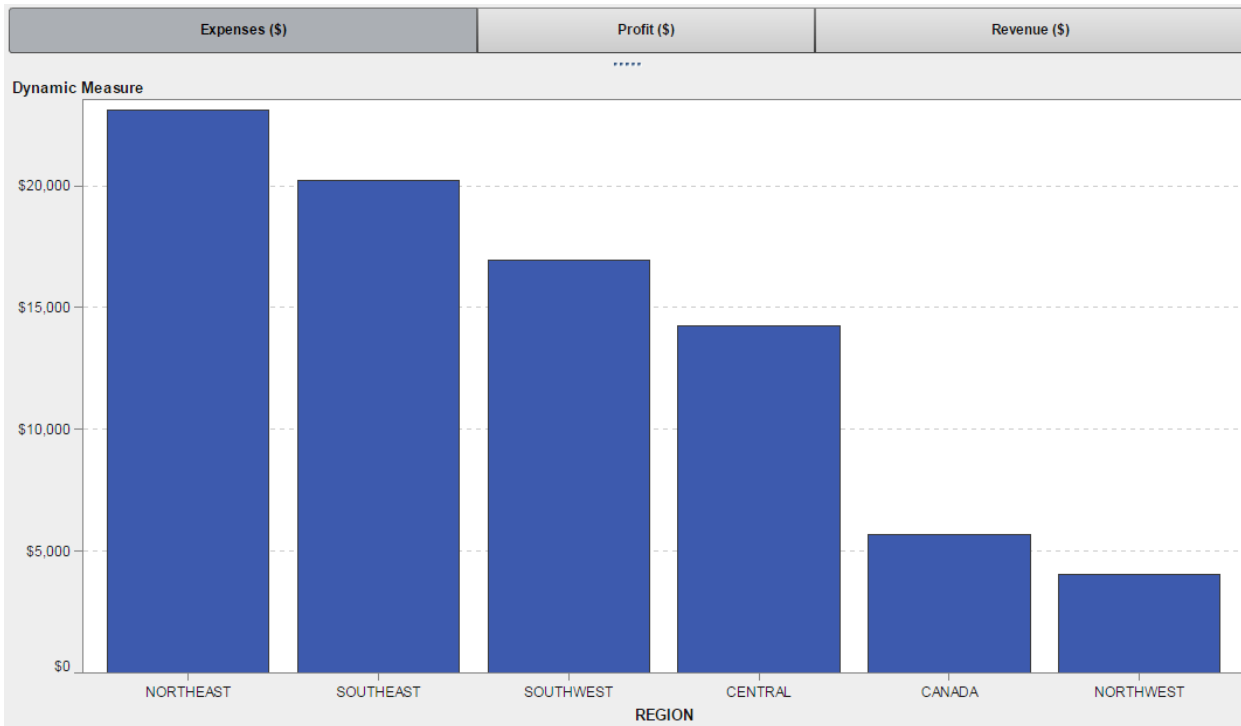


Figure 12: Your Bar Chart Should Resemble This Image

That is it, you are done. The calculated item does all of the work when the prompt is changed. Because “Expenses (\$)” is selected in the button bar, the value of “MeasureSelection Parameter” will be “Expenses (\$)”. In the calculated item, it will match that condition and return *Expenses* as the measure.

As shown in the screenshot, you can change the format of the calculated item. Because all of the values are in dollars, you can change the calculated item’s format to **Dollar**. The Y-Axis label always reads “Dynamic Measure” no matter what option is selected because that is the name of the calculated item.

Parameters are global, so this calculated item can be used in multiple graphs and on different sections. However, the parameter value can be controlled only by this single prompt. You are not able to create another prompt that uses the same parameter.

Parameters are separate from interactions, so they are not displayed in the **Interactions** tab. If you want to see how the parameter is being used, place your mouse pointer over the parameter in the **Data** tab. A tooltip should appear that provides this information.

```

Name: MeasureSelection Parameter
Value: Expenses ($)
Value From: Button Bar 1
Consumers
- Bar Chart 1

```

Figure 13: An Example Tooltip

CONCLUSION

Parameters are an amazing feature that allow a user to dynamically change variables on the fly. This is just a single example of what a parameter can do and there are many other concepts that can be implemented using parameters.

The example touches only on changing the measure of a graph, but the same concept can be applied to change the subgroup or category of a graph. All of this can be done simultaneously with multiple prompts. Users no longer need to browse through multiple graphs to find the data they need. You are now able to create reports that can directly address what data your user is looking for.

ACKNOWLEDGMENTS

As this was my first paper, I want to extend a big thank you to everyone that helped make this possible. This would not have been possible with the support I had from the Mobile team. Special thanks to my manager LA Lennon for supporting me through this and Yvonne Waters for giving me the much needed push to get this done.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Kenny Lui
SAS Institute, Inc.
919-531-1381
Kenny.Lui@sas.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.