

Paper RV-246

Building Interactive Microsoft Excel Worksheets with SAS® Office Analytics

Tim Beese, SAS Institute Inc., Cary, NC

ABSTRACT

Microsoft Office has over 1 billion users worldwide, making it one of the most successful pieces of software on the market today. Imagine combining the familiarity and functionality of Microsoft Office with the power of SAS® to include SAS content in a Microsoft Office document. By using SAS® Office Analytics, you can create Microsoft Excel worksheets that are not just static reports, but interactive documents. This paper shows how to create an interactive experience in Excel by leveraging Visual Basic for Applications using SAS data and stored processes. Finally, this paper demonstrates how to open SAS® Visual Analytics reports into Excel, so the interactive benefits of SAS Visual Analytics are combined with the familiar interface of an Excel worksheet. All these interactions with SAS content are possible without leaving Microsoft Excel.

INTRODUCTION

If you're like me, you use Microsoft Office every day. It is one of the most well-known and widely used pieces of software on the planet. Microsoft Office is easily recognizable and simple to use. SAS is the leader in business intelligence and can help you analyze your data and give you the insight required to make intelligent decisions. The SAS Add-In for Microsoft Office combines these two products, giving you the best of both worlds -- the ease of use that Office provides alongside the analytical power of SAS.

For years, data analysts have taken their data and run it through a SAS program to analyze it and create a report. You can copy and paste the results from SAS into a Microsoft Office document. Then in the Microsoft Office document, you add your own conclusions and create the final report. When the underlying data changed, you would have to run the program in SAS again, and then copy and paste the new results into the Microsoft Office document. With the SAS Add-In for Microsoft Office, reports can be updated directly in the Microsoft Office document, which saves you time and effort.

This paper goes even further. Using two examples, this paper shows how to combine Microsoft Excel and SAS to build interactive reports. The SAS Add-In for Microsoft Office lets you drive your analysis from Excel, so you can use these reports to interact with and analyze your data directly in Excel. The examples in this paper focus on using SAS Visual Analytics and SAS Stored Processes.

INTERACTING WITH SAS VISUAL ANALYTICS REPORTS

SAS Visual Analytics provides a complete platform for analytics visualization and interactive reporting capabilities. SAS Visual Analytics easily explores and analyzes huge volumes of data with SAS Visual Analytics Designer and SAS Visual Analytics Explorer. You can quickly create interactive reports that are easy to consume in web and native mobile tablet applications.

For more information about SAS Visual Analytics, see www.sas.com/software/visual-analytics.

Using SAS Add-In for Microsoft Office, you can open and interact with these SAS Visual Analytics reports. The following example analyzes the gross box office sales for all three movies in the Hobbit trilogy. The report has multiple sections. Each section shows different interactive aspects between the report and Microsoft Excel.

To open a SAS Visual Analytics report into Excel, go to the **SAS** tab on the ribbon and click **Reports**. The Reports dialog box appears. You can browse your SAS Folders and choose a report to open.

OPTIONS FOR OPENING THE REPORT

You have two options when opening a SAS Visual Analytics report. You can choose to first preview the report in a Microsoft Excel task pane, or you might choose to skip the preview and open the report directly into the Microsoft Excel worksheet. If you want to preview the report first, you must set the **Preview**

Visual Analytics reports before inserting into a document option. To set this option, go to the **SAS** tab on the Microsoft Excel ribbon and select **Tools > Options**. The SAS Add-In for Microsoft Office Options dialog box appears. In the navigation pane, click **Results**. Under the **Visual Analytics Reports** heading, find the **Preview Visual Analytics reports before inserting into a document** option.

If this option is deselected when you open a report, the entire report is automatically inserted into your Microsoft Excel workbook. Sometimes this is the desired effect. However, previewing the report provides a more interactive experience. Here are some advantages to previewing the report.

- You see the entire report before inserting it, allowing you to make sure that you have the report you want.
- You can perform any interactions on the report before inserting it. Previewing can result in better performance because time won't be wasted displaying content in the worksheet that will change when you interact with the report.
- You can select individual elements, letting you decide which elements to display, as well as where to display them. As a result, you interactively create your own layout in the Office document.

INTERACTING WITH THE REPORT

When you preview a report, it is opened inside the Report Controls pane. This is a custom task pane in Excel where SAS displays a viewer for the report. This viewer is fully interactive, allowing you to perform all of the report interactions, such as changing prompts; collapsing, expanding or drilling on a crosstab; selecting elements; and sorting tables.

Display 1 shows the **Data** section of the example report in the Report Controls pane. Notice the drop-down list at the top filters the table that appears on the **Data** tab. Also, the table is sortable, which enables you to customize how you view your data.

In addition to interacting with your report, you can refresh your report. A refresh will update the report and replace the old one in the Report Controls pane. Any prompt changes, crosstab interactions, or table sorts are retained and applied to the new report when it is downloaded. Any report elements that have already been inserted into your document are also updated.

You can also choose to reset your report, which restores your report to the default settings that are saved in SAS Visual Analytics.

Because screen real estate is limited inside of a custom task pane, you might want to view a single element more closely, instead of the entire report. There are two buttons that enable you to view your report in a larger area.

Date	Rank	Gross	The
Friday, December 14, 2012	1	\$37,125,547	
Saturday, December 15, 2012	1	\$27,739,815	
Sunday, December 16, 2012	1	\$19,751,941	
Monday, December 17, 2012	1	\$7,738,293	
Tuesday, December 18, 2012	1	\$7,855,090	
Wednesday, December 19, 2012	1	\$6,327,187	
Thursday, December 20, 2012	1	\$6,615,027	
Friday, December 21, 2012	1	\$10,078,827	
Saturday, December 22, 2012	1	\$14,160,166	
Sunday, December 23, 2012	1	\$12,649,372	
Monday, December 24, 2012	1	\$6,897,267	

Display 1. The Report Controls Pane



The maximize button extends the selected element to fill the entire space available for the viewer.



The full screen button temporarily displays this report across your entire screen. You can continue to interact with your report in full screen mode.

Combining the full screen mode with a maximized element provides a full-sized view of the individual element you want to examine.

INSERTING YOUR REPORT INTO THE WORKSHEET

When you are ready to insert your report into your Microsoft Excel worksheet, there are two ways to do this. You can insert one element at a time, or you can insert the entire report at once.

When inserting one element at a time, the element is inserted at the current selection in the worksheet. You can simply select a cell in the worksheet, select the element you want to insert, and then click the **Insert Selection** button.

If you choose to insert the entire report, you are prompted for a location. You can choose to insert the report at any cell location on an existing worksheet, or you can choose to create a new worksheet to display your report. There is even an option to put each section of your report on its own worksheet. This allows your Excel view of the report to mimic the view that you see in SAS Visual Analytics.

APPLYING STYLE TO YOUR REPORTS (OR NOT)

When inserting your results into the Excel worksheet, you can choose how to apply styles to your content. To set this option, open the SAS Add-In for Microsoft Office Options dialog box. In the navigation pane, click **Results**. Under the **Visual Analytics Reports** heading, find the option for **Apply style to Visual Analytics Reports**. There are three different choices here:

- **Never.** When the report is rendered in the Office document, no style is applied. This is good for when you want to apply your own styles in Excel. When you select this value, the styles you have in your worksheet are retained when you refresh the report.
- **When inserting content only.** The SAS style is applied the first time that a report is inserted into the worksheet. This gives you the look and feel of the SAS report. However, when the report is refreshed, the style is not applied again, so any changes you made to the style are preserved. The drawback to this option is that when results change in size, you need to manually apply the style to any new result areas. This is particularly true for tables that change size frequently when refreshed.
- **When inserting and refreshing content.** The SAS style is always applied. This overwrites any style settings that are manually set on the results when you refresh. However, using this setting provides a consistent look and feel for your entire report. Even if your tables change in size, the new results have the style applied.

Regardless of your style setting, any graphs that are generated by the report always have the style applied, because these graphs are static images and are not subject to style changes by the Excel user.

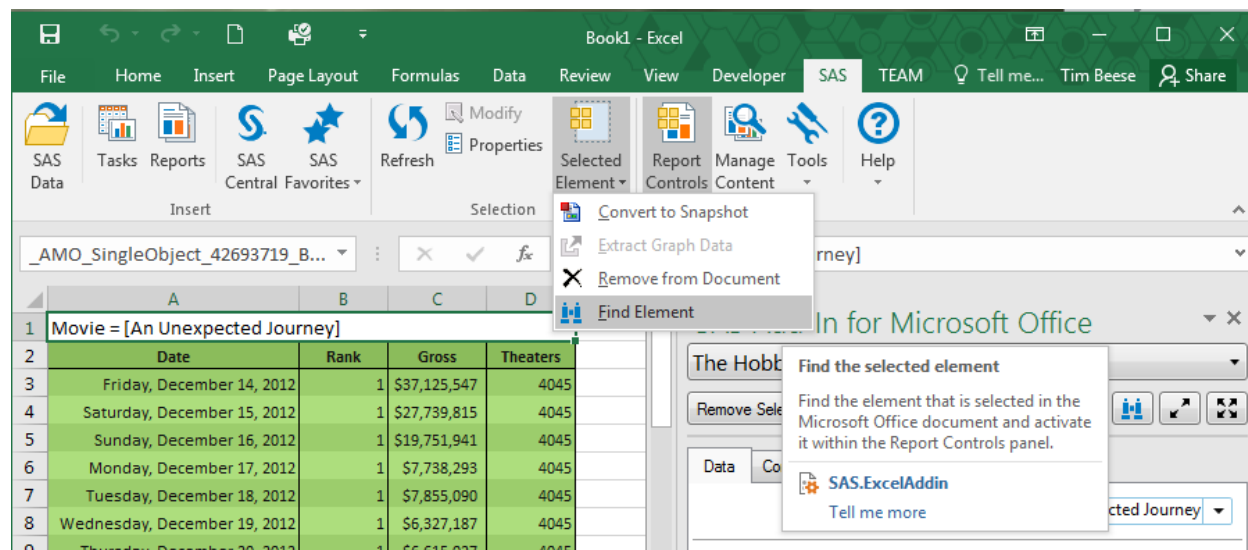
MANAGING YOUR REPORT ELEMENTS IN THE WORKSHEET

After you've inserted one or more elements into your worksheet, you can continue interacting with your report. If you make any changes in the Report Controls pane to an element that is already inserted into the worksheet, the report element in your worksheet is automatically updated to reflect the changes. For example, if you sort the table in the Report Controls pane, the table that has been inserted in the worksheet is also sorted.

If you have a large workbook (perhaps with several worksheets) and you're looking at a particular element in the Report Controls pane, you can easily locate the corresponding element that you inserted

into the workbook. Click  to find the element. SAS will activate the worksheet and select the cells that represent that element.

Likewise, if you are working with an element in your worksheet and you have several reports open in the Report Controls pane, you can easily find a specific report element by selecting **Selected Element > Find Element** in the **SAS** tab in the ribbon. This option activates the report in the Report Controls pane that contains the element that you selected in the worksheet and selects the element in the Report Controls pane as well. Display 2 shows the **Selected Element** menu on the **SAS** tab of the ribbon.



Display 2. Using the Selected Element Menu in the SAS Tab

Remember: if you're working with an element in the Report Controls pane, use the Find button in the Report Controls pane to find the same element in the worksheet. If you're working with an element in the worksheet, use **Selected Element > Find Element** in the **SAS** tab in the ribbon to find the element in the Report Controls pane.

If you no longer wish to view an element in the worksheet, there are two easy ways to remove it.

- If you're working in the Report Controls pane, select the element that you want to remove. The same button that you used to insert the selected element is now called **Remove Selection**, and clicking this button removes the selected element from the document.
- If you're in the worksheet with the element selected, go to the **SAS** tab in the ribbon and select **Selected Element > Remove from Document**. This selection removes the entire element from the document. You can insert it again at any time, so changing your layout in the Microsoft Office document is easy.

HANDLING LARGE REPORTS

Reports can be considerably large in size. They might have dozens of elements in a section or even several sections. It is possible that a report could have a large amount of data that is required to display and interact with the report. By default, when a report is opened into Excel, the full state of the report is also stored with the workbook, so when you save and reopen the workbook, the report can be re-created exactly as it was before. However, saving the full state of the report can introduce issues with memory and large document sizes.

To help control these issues, there is an option in the SAS Add-In for Microsoft Office Options dialog box. In the navigation pane, click **Results**. Under the **Visual Analytics Reports** heading, find the **Save the full report state within the Office document** option.

- When this option is selected, the report uses a lot more memory to save the state of the report and the size of your Excel document is considerably larger. However, this option gives you the best retention of the report.

- When this option is not selected, the impact on memory and document size is minimal. However, the full state of the report is not saved. The report retains the selection for any prompts, the drill or collapse/expand state of any crosstabs, and the sort state of the tables. Other state settings (such as selections within an element or filters between elements) are not retained. When you reopen your Excel workbook, you need to refresh the report to get the view synchronized between the workbook and the Report Controls pane.

SIZING YOUR GRAPHS

When you view a graph inside of the Report Controls pane, the pane resizes itself and updates the display of the graph based on the amount of space available. Graphs work differently when inserted into the Excel worksheet. In the worksheet, graphs have a scroll bar and a lot more room within the sheet. It doesn't make sense for them to change their size and display when inserted in the worksheet. With more real estate, the graph can show more information.

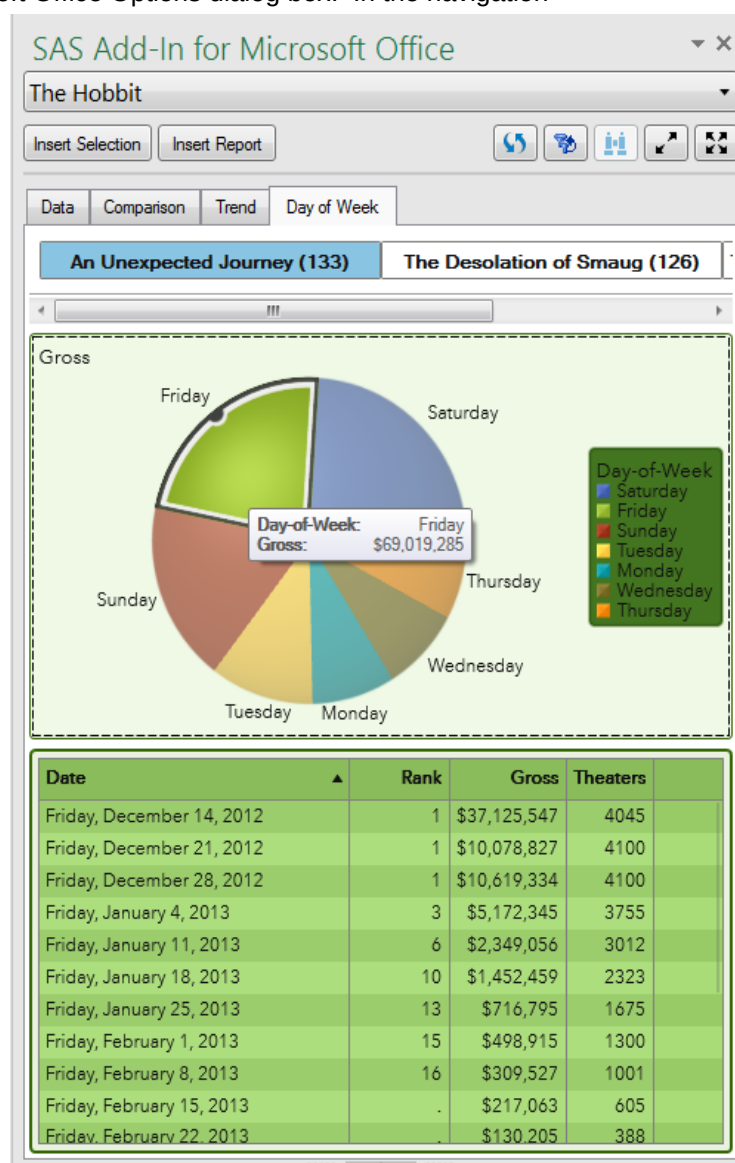
When any graph or KPI is inserted into the Office document, it uses a default size controlled by an option. To change this option, go to the SAS Add-In for Microsoft Office Options dialog box. In the navigation pane, click **Results**. Under the **Visual Analytics Reports** heading, find the option for **Graph Size**. Use this option to control the width and height of the image that is generated.

It is impossible to have one perfect size for all graphs, so the best method for getting your graphs exactly the size that you want is to insert the graph into your document and see how it looks. Then, you can resize the image in the Excel worksheet. After you find the right size for the image, refresh the report and a new image is generated with the optimal view for the size you have allocated. This allows you to custom fit each graph for your document.

ELEMENT TO ELEMENT INTERACTIONS

When building a SAS Visual Analytics report, one element can be set up to apply a filter or a brush for another element. For example, you could have a pie chart that filters a table. As each section of the pie is selected, the data that is used for that portion of the pie is displayed in the table. Display 3 shows a pie chart that represents each day of the week. When an individual slice is selected, the table below the pie chart is updated to show just the data for the selected day of the week.

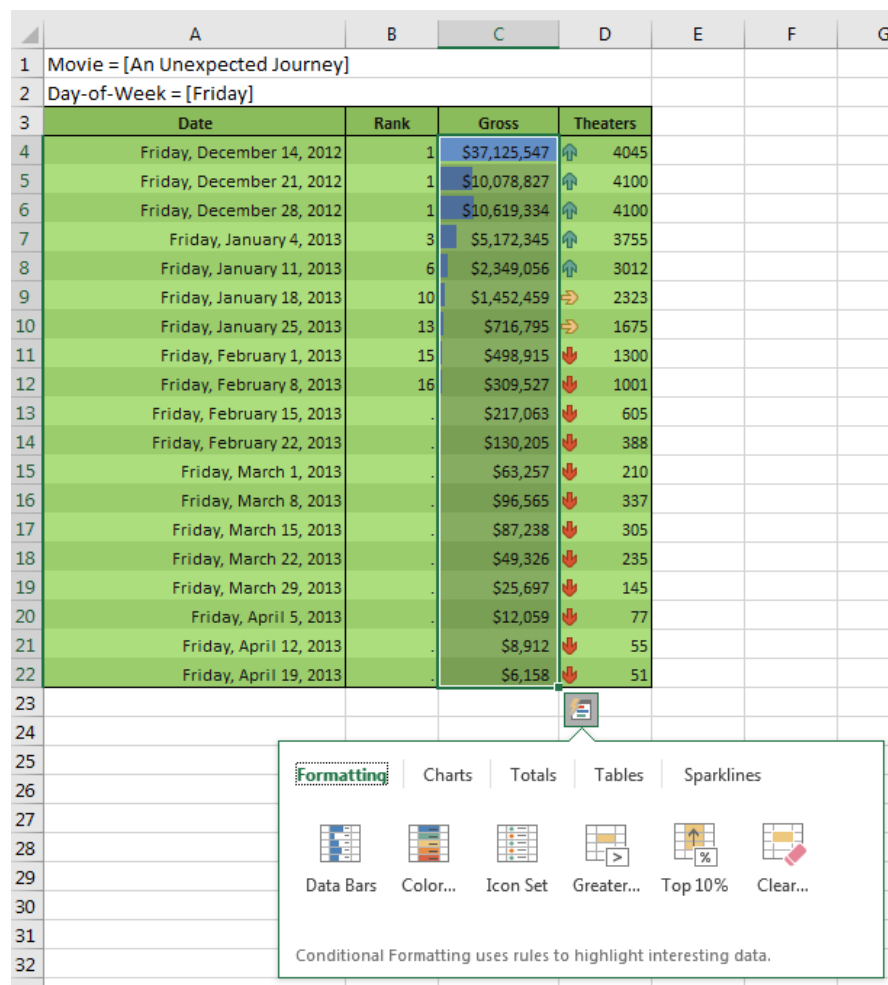
In this example, if you insert the table into your worksheet and select a day of the week, the data in the worksheet gets updated each time your selection changes. This is a great way to quickly analyze and work with your data in an interactive manner. Your worksheet isn't just a representation of static data. It is responding to your graph selection and giving you detailed information that pertains to your selection. Because the data is stored locally, these interactions are designed to be quick.



Display 3. A Pie Chart Filtering a List Table

USING THE QUICK ANALYSIS TOOLBAR IN EXCEL

Microsoft added the Quick Analysis toolbar in Excel 2013. This toolbar is a very useful way to expand on the data that you are viewing. To use it, select an entire column of data, such as the gross revenue for a particular day of the week. As you position your mouse pointer over the selection, a floating Excel button appears in the bottom right hand corner. Click the floating button to bring up the Quick Analysis toolbar. Display 4 shows the Quick Analysis toolbar, as well as a column formatted with data bars and another column formatted with an icon set.



Display 4. Quick Analysis Toolbar in Excel

The Quick Analysis toolbar lets you apply formatting to the selection, create charts and tables, or add totals or sparklines to your data. Click **Data Bars** to have Excel create custom data bars that show how all of the values in your selection relate to each other. In the example, the data bars quickly show you which days made the most money and which ones did not. Now, as you change your selection in the pie chart, the data in the table updates, and the data bars immediately update to represent the values from the new data. Using data bars enables you to inspect your data and understand how the values relate to each other without really even needing to see the numbers. You can do the same with color scales and icon sets.

If you choose to insert a chart, a native Excel chart that represents the data from the selection is added to the worksheet. This chart is purely an Excel chart and can be formatted the same as any other native Excel chart. As the data in your table is updated, the chart is automatically updated as well. This is great for when your SAS Visual Analytics report does not contain a chart, but you want to include one with your workbook.

CREATING SNAPSHOTS

In the example where the report is analyzing the data for a particular day of the week, you might want to compare the data for one day of the week to another. You can't easily do this comparison with the report that was created because there is only one table and it always represents the day that is selected in the pie chart. You would need a custom SAS Visual Analytics report to show this information.

However, since you can insert the data into Excel, you can easily accomplish this task without having to change anything about the SAS Visual Analytics report. First, select the day you want to start comparing, such as Friday. Insert the table into your Excel worksheet. Then, from the **SAS** tab on the ribbon, select **Selected Element > Convert to Snapshot**. This option converts the content into pure Excel content. Because there are no more links connecting this content to the Report Controls pane, clicking in the pie chart no longer updates the table in the worksheet.

Essentially, this element is no longer treated as if it exists in the document, which means you can insert it again. Choose another day of the week, such as Saturday, and insert the table into the worksheet next to your existing table. Display 5 shows two tables comparing different days of the week while the movie was shown in theaters.

	A	B	C	D	E	F	G	H	I
1	Movie = [An Unexpected Journey]					Movie = [An Unexpected Journey]			
2	Day-of-Week = [Friday]					Day-of-Week = [Saturday]			
3	Date	Rank	Gross	Theaters		Date	Rank	Gross	Theaters
4	Friday, December 14, 2012	1	\$37,125,547	4045		Saturday, December 15, 2012	1	\$27,739,815	4045
5	Friday, December 21, 2012	1	\$10,078,827	4100		Saturday, December 22, 2012	1	\$14,160,166	4100
6	Friday, December 28, 2012	1	\$10,619,334	4100		Saturday, December 29, 2012	1	\$11,588,637	4100
7	Friday, January 4, 2013	3	\$5,172,345	3755		Saturday, January 5, 2013	3	\$7,629,018	3755
8	Friday, January 11, 2013	6	\$2,349,056	3012		Saturday, January 12, 2013	6	\$4,190,664	3012
9	Friday, January 18, 2013	10	\$1,452,459	2323		Saturday, January 19, 2013	9	\$2,820,219	2323
10	Friday, January 25, 2013	13	\$716,795	1675		Saturday, January 26, 2013	12	\$1,647,767	1675
11	Friday, February 1, 2013	15	\$498,915	1300		Saturday, February 2, 2013	12	\$1,008,214	1300

Display 5. Using a Snapshot to Compare Data Side by Side

USING EXCEL FORMULAS

When SAS inserts data into the worksheet, it inserts the raw values and then applies an Excel format to those values to make the data match what is in the Report Controls pane as closely as possible. Because the raw values are being used, the values can be used in Excel formulas. Continuing with the previous example where you created a snapshot of Friday to compare to the active data of Saturday, you can create an Excel formula that compares the gross revenue for Friday versus Saturday for each week that the movie was in theaters.

As you change your selection in the pie chart and the data for Saturday is updated with the selected day of the week, the formulas are recalculated using the values from the updated cells. Display 6 shows how to use a formula to compare values from both tables and display the result in the Excel worksheet.

	A	C	E	F	H	J	K
1	Movie = [An Unexpected Journey]		Movie = [An Unexpected Journey]				
2	Day-of-Week = [Friday]		Day-of-Week = [Saturday]				
3	Date	Gross		Date	Gross		Comparison
4	Friday, December 14, 2012	\$37,125,547		Saturday, December 15, 2012	\$27,739,815		\$9,385,732
5	Friday, December 21, 2012	\$10,078,827		Saturday, December 22, 2012	\$14,160,166		(\$4,081,339)
6	Friday, December 28, 2012	\$10,619,334		Saturday, December 29, 2012	\$11,588,637		(\$969,303)
7	Friday, January 4, 2013	\$5,172,345		Saturday, January 5, 2013	\$7,629,018		(\$2,456,673)
8	Friday, January 11, 2013	\$2,349,056		Saturday, January 12, 2013	\$4,190,664		(\$1,841,608)

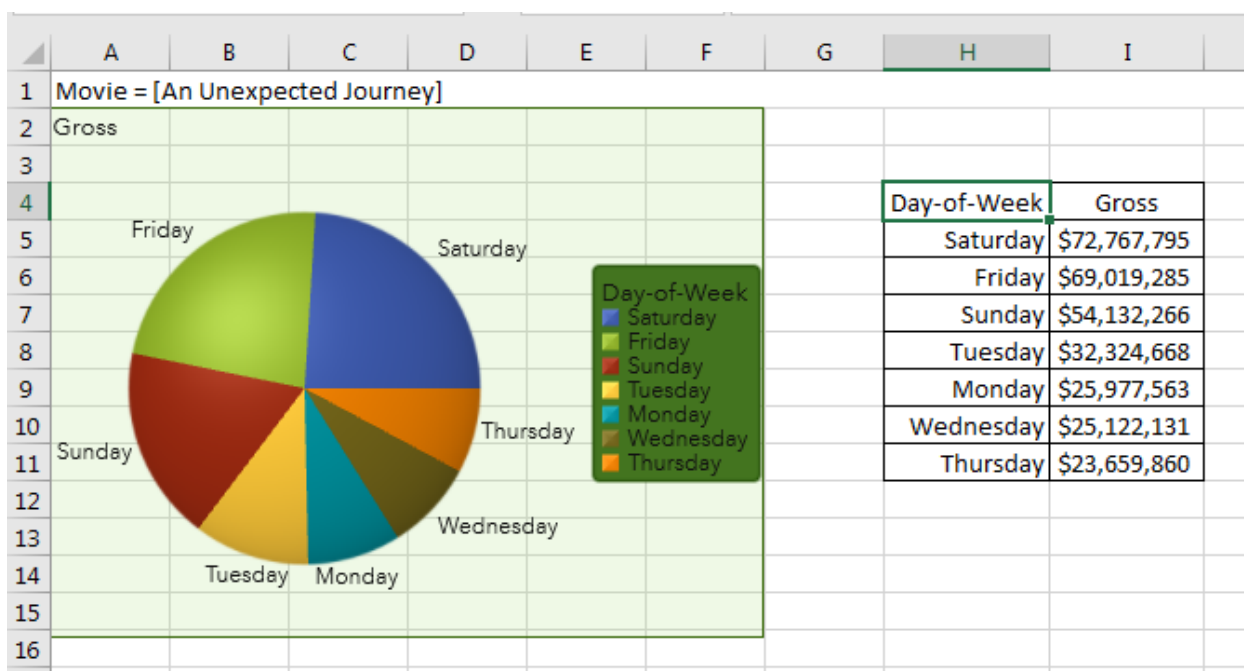
Display 6. Comparing Two Tables with an Excel Formula

This example demonstrates another way that you can use the power of SAS with the many features of Excel to understand your data directly in your Microsoft Excel worksheet.

EXTRACTING DATA FROM A GRAPH

You've seen how to take the data from a SAS Visual Analytics table and use Excel to create a graph with that data. You can also go the other direction. If you have a graph that was created by SAS Visual Analytics, you can use the SAS Add-In for Microsoft Office to extract the data behind the graph.

First, insert the graph into the Excel document. From the **SAS** tab on the ribbon, select **Selected Element > Extract Graph Data**. You can choose where to put the data that was used to create the graph. This is not the full set of data behind the graph. It is just the aggregated data that is used to build the graph. Display 7 shows a pie chart that has been inserted into the worksheet and the data extracted from it.



Display 7. Using Extract Data to Display the Data behind a Graph

After you've extracted the data, if you change a prompt or refresh the report and your graph changes, the extracted data is automatically updated as well. All of the Excel Quick Analysis functions are available on the extracted data, allowing this to become another interactive part of your Excel worksheet.

BUILDING AN INTERACTIVE STORED PROCESS

Another type of content that can be created by SAS is a stored process. A stored process enables you to share SAS code with others. There are several features centered around stored processes where the SAS Add-In for Microsoft Office can create an interactive experience.

- You can run a stored process from Excel and get the results back into your Excel worksheet. When you refresh the stored process, the results in your worksheet are updated.
- You can populate stored process prompts using cell values in an Excel worksheet.
- You can stream data from an Excel worksheet to use as the input data for the stored process.
- You can create output parameters that write directly to a specified cell in an Excel worksheet.
- You can write Visual Basic for Applications (VBA) code to automatically refresh the results when a user clicks a button or when certain cell values are changed in the Excel workbook.

The stored process example analyzes the box office revenue of the Star Wars prequels. The example starts with a workbook containing the gross revenue for each day these movies were in theaters, as well as some SAS code that analyzes the gross revenue by the day of the week.

CREATING THE STORED PROCESS

In my opinion, SAS Enterprise Guide is the easiest way to create a stored process, especially when the stored process includes more than just running SAS code. Adding prompts, input streams, and output parameters makes things more complex, but the Create New Stored Process Wizard in SAS Enterprise Guide makes those extra steps extremely easy to do.

There are several steps to creating the stored process:

- Adding code to the stored process
- Setting the execution options
- Defining the prompts
- Creating an output parameter
- Creating an input stream

To create the stored process, open SAS Enterprise Guide and select **File > New > Stored Process**. The Create New Stored Process wizard appears. In the first step, enter some basic information about the stored process such as what to call it and where to store it.

Adding Code to the Stored Process

In step 2 of the wizard, write the SAS code for your stored process. Every stored process must contain SAS code. Other parts of the stored process are optional, but not the SAS code. You can take some SAS code that you have already written from a program node and paste it in here, or you can write the code directly in this wizard.

Here is the example SAS code to analyze the data that exists in the Excel workbook. You can simply copy and paste it into step 2.

```
/* Use the dayTotal macro variable as an output parameter. It will be set
   below to total the revenue for a given movie on a given day. */
%LET dayTotal=0;

/* Define an input stream that will allow us to read the data from
   Excel and use it in the stored process. */
```

```

LIBNAME instream xml;
DATA EXCELDATA;
    SET instream.&_WEBIN_SASNAME;
RUN;

/* This will prep the data, taking the date variable and creating a new
   variable to track which day of the week it corresponds to. It also
   filters the data so that only the selected movie is included. */
PROC SQL;
    CREATE TABLE WORK.SW AS
    SELECT
        t1.date FORMAT=DATE9.,
        (strip(put(t1.date, WEEKDATE9.))) AS dayname,
        t1.date as dayindex FORMAT=WEEKDATE9.,
        t1.gross FORMAT=DOLLAR17.,
        t1.theaters
    FROM EXCELDATA t1
    WHERE T1.movie = "&movie";
QUIT;

/* This creates a pie chart breaking down the movie revenue based on the
   day of the week. */
TITLE "Day of the week breakdown for &movie";
FOOTNOTE;
PROC GCHART DATA=WORK.SW;
    PIE dayindex /
    SUMVAR=gross
    TYPE=SUM
    NOLEGEND
    SLICE=OUTSIDE
    PERCENT=NONE
    VALUE=OUTSIDE
    OTHER=4
    ANGLE=0
    OTHERLABEL="Other"
    COUTLINE=BLACK
    DISCRETE
    NOHEADING;
RUN; QUIT;

/* This creates a table displaying the revenue made for a particular day
   of the week, during its time in theaters. */
TITLE "Movie revenue for &movie on &dayofweek";
PROC PRINT DATA=WORK.SW noobs;
    WHERE dayname = "&dayofweek";
    VAR date gross theaters;
RUN;
QUIT;

/* This puts the total gross for all days that match the requested day of the
   week into a macro variable. This is the output parameter for the stored
   process that will be written back out. */
PROC SQL NOPRINT;
    SELECT sum(gross) FORMAT=DOLLAR17. INTO :dayTotal
    FROM WORK.SW
    WHERE dayname = "&dayofweek";
QUIT;

```

Setting the Execution Options

Step 3 of the wizard gives options for how to execute this stored process. You can choose to execute it on a stored process server or a workspace server. You can choose where to store the source code as well. Leaving the defaults for these options is generally the best practice. However, you may wish to choose to store the source code in metadata so that you don't have to specify a folder location. If you want your source code available from a network location, you need to provide the repository and a source file for where the SAS code will be stored.

The other thing that you need to set in this step is to have both the stream and package result capabilities. The results are returned to Excel in a package format, but in order for the input stream to be created, select **Stream** from the **Result capabilities** option. Display 8 shows how to set the execution options.

The screenshot shows the 'Create New Stored Process' wizard, Step 3 of 6: Execution Options. The window has a title bar with the text 'Create New Stored Process' and a close button. The SAS logo is in the top right corner. The main content area is divided into several sections:

- Application server:** A dropdown menu showing 'SASApp'.
- Server type:** Three radio button options:
 - ☒ **Default server**: Select this option to allow the client application to specify the server.
 - ☐ **Stored process server only**: Select this option if the stored process uses sessions or if it uses replay (for example, to produce graphics in streaming output).
 - ☐ **Workspace server only**: Select this option if the stored process must be run under the client identity.
- Source code location and execution:** Two radio button options:
 - ☐ **Allow execution on other application servers (store source code in metadata)**
 - ☒ **Allow execution on selected application server only**
 - ☒ **Store source code in metadata**
 - ☐ **Store source code on application server**
- Source code repository:** A dropdown menu showing '*** Select a repository ***' and a 'Manage...' button.
- Source file:** A checkbox labeled 'Source file:' is checked. Below it is a text field containing 'Stored Process.sas' and an unchecked checkbox labeled 'Overwrite existing file'.
- Result capabilities:** Two checked checkboxes: ☒ **Stream** and ☒ **Package**.

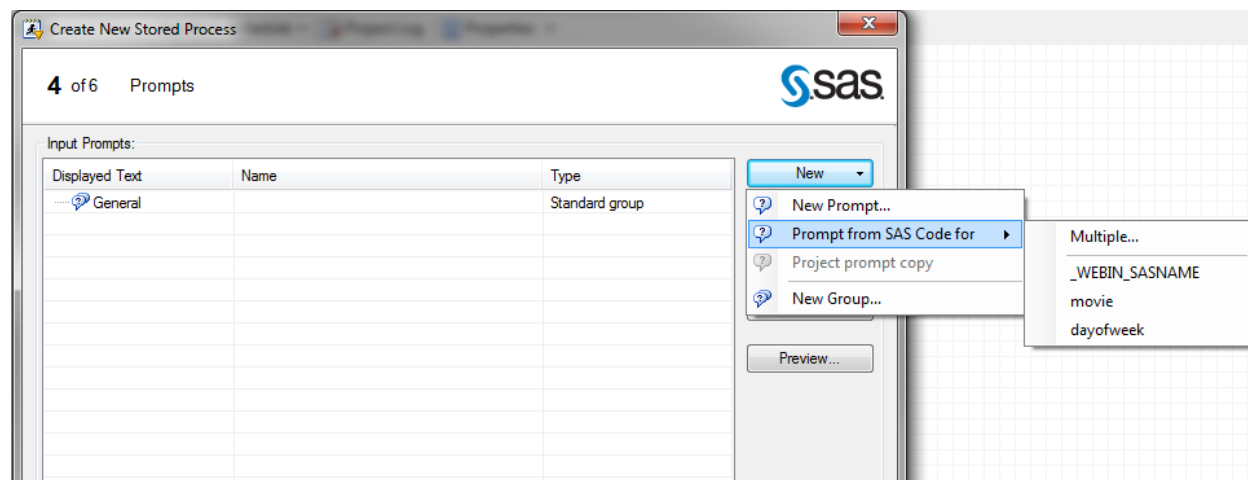
At the bottom of the window, there is a 'More (F1)...' link and a set of navigation buttons: '<Back', 'Next>', 'Finish', and 'Cancel'.

Display 8. Setting the Execution Options

Defining the Prompts

Step 4 of the wizard lets you choose your input prompts and your output parameters. For the input prompts, you can create them manually, defining each prompt. Or, you can choose to let them be automatically generated from the SAS code. Either way, you still need to define the type for each prompt.

From the **Input Prompts** section, select **New > Prompt from SAS Code for**. Every macro variable in your code is listed in the pop-up menu. You don't need to create a prompt for `_WEBIN_SASNAME`. The SAS Add-In for Microsoft Office automatically populates this reserved macro variable to define the data source that is being streamed in to the stored process. You need to create prompts only for the macro variables that you want the user to provide. In this case, you need prompts for `movie` and `dayofweek`. Display 9 shows where to add prompts for each macro variable defined in the stored process.



Display 9. Adding Prompts to a Stored Process

Choosing **movie** from the pop-up menu opens the Edit Prompt dialog box. From here, you can change the displayed text for the prompt, define whether this prompt requires a value, and set its default value. On the Prompt Type and Values page, you define the type for this prompt. The type determines the control that appears in the client for selecting the value of the prompt.

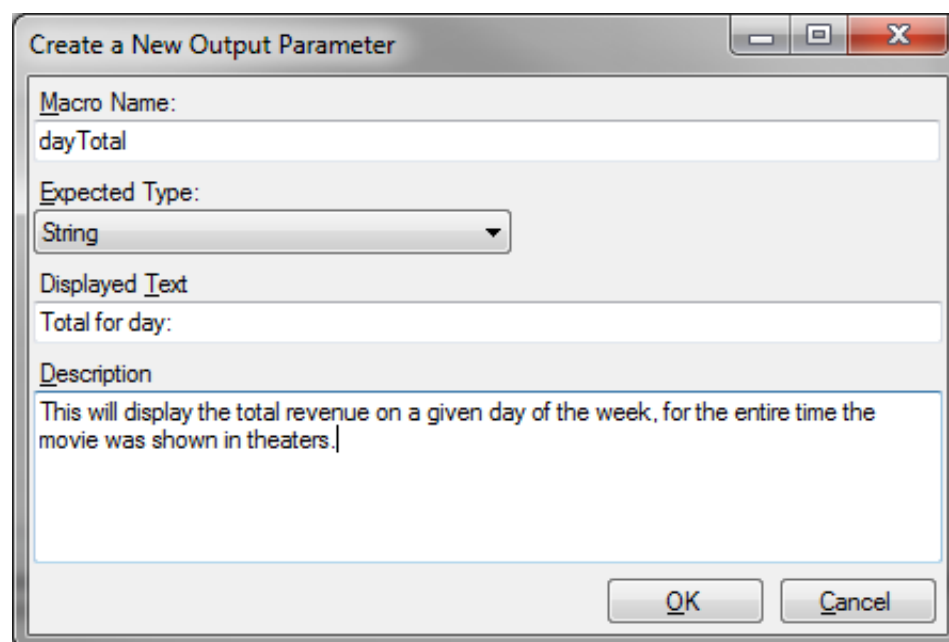
The SAS add-in can display all prompt types, but it supports only connecting Excel cell values for basic text, numeric, and date prompts. The SAS add-in does not link to the worksheet for multi-value prompts or range prompts. Those prompts need to be set in the dialog box that appears when the stored process runs.

The `movie` and `dayofweek` prompts are both text prompts, which enable the user to provide text strings for the movie name and day of the week to the stored process.

Creating an Output Parameter

Step 4 of the wizard also lets you define any output parameters. Output parameters are macro variables that are defined when you create the stored process. The client (in this case, the SAS add-in) can request the value of the macro variable to display within the Office document. For Excel, the output parameter can be linked to a cell in the worksheet. The code that you copied and pasted into step 2 defines a macro variable called `&dayTotal`.

In the **Output Parameters** section, click **New**, and then enter the name of the macro variable. You can assign the displayed text and choose a description as well. Whatever value is assigned to the macro variable in the code can be displayed in the Microsoft Excel worksheet. Display 10 shows how to create a new output parameter.



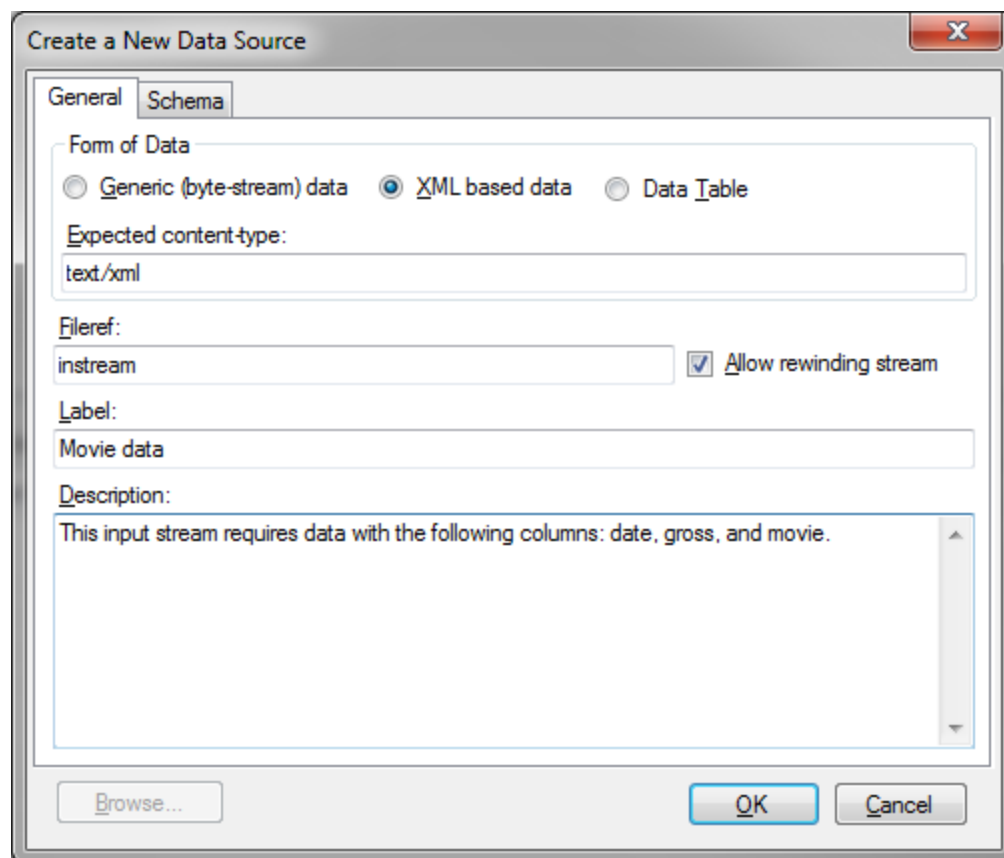
Display 10. Creating an Output Parameter

Creating an Input Stream

The input stream is the data that is going to be streamed to the stored process. Adding an input stream requires that the stored process be executed from an application that supports input streams. SAS Enterprise Guide does not support input streams, but the SAS Add-In for Microsoft Office does.

Step 5 of the Create New Stored Process wizard is where you create the input stream. See the Data Sources section at the top of the step. Click **New** to define a new data source. This displays a dialog box to create the data source. The SAS Add-In for Microsoft Office supports only streaming XML-based data, so select that as your form of data.

In the SAS code that you copied into step 2, the name of the input stream is **instream**. In this step, add **instream** as the fileref, and be sure to check the **Allow rewinding stream** option. This is needed so that the XML LIBNAME engine can make multiple passes through the stream in order to correctly set up the data source. Display 11 shows how to create an input stream for your stored process.



Display 11. Creating an Input Stream for Your Stored Process

It is also a good idea to provide a description of what type of input data this stored process requires. If certain variable names are required, you can state that in the description, so the user can make sure that their Excel column names are properly defined. The description is available as a tooltip when the user is prompted for the input data while running the stored process. This stored process requires columns named: date, gross, and movie.

A helpful trick for building and debugging a stored process with input streams is to create a sample data set with similar data, and write your stored process to analyze this data before you define the input stream. Using this trick enables you to run your stored process from SAS Enterprise Guide, analyze the results, fix any errors, and try again as many times as needed. After you have the code correct and are ready to run the stored process in Excel, create the input stream and update the SAS code to use the input stream as the data source instead of the sample SAS data set.

Stored Process Complete!

Finally, the stored process is ready for execution. Click **Finish** in the wizard, and SAS Enterprise Guide creates the stored process for you. Because the stored process requires an input stream, you won't be able to execute the stored process from SAS Enterprise Guide, but it is ready for use in Microsoft Excel.

RUNNING THE STORED PROCESS IN EXCEL

Now that you have created the stored process, you can execute it in Excel. You already have the data handy. You just need to stream it to the stored process. Go to the **SAS** tab on the ribbon and click **Reports**. From the Open dialog box, you can browse for your stored process.

Providing Prompt Values

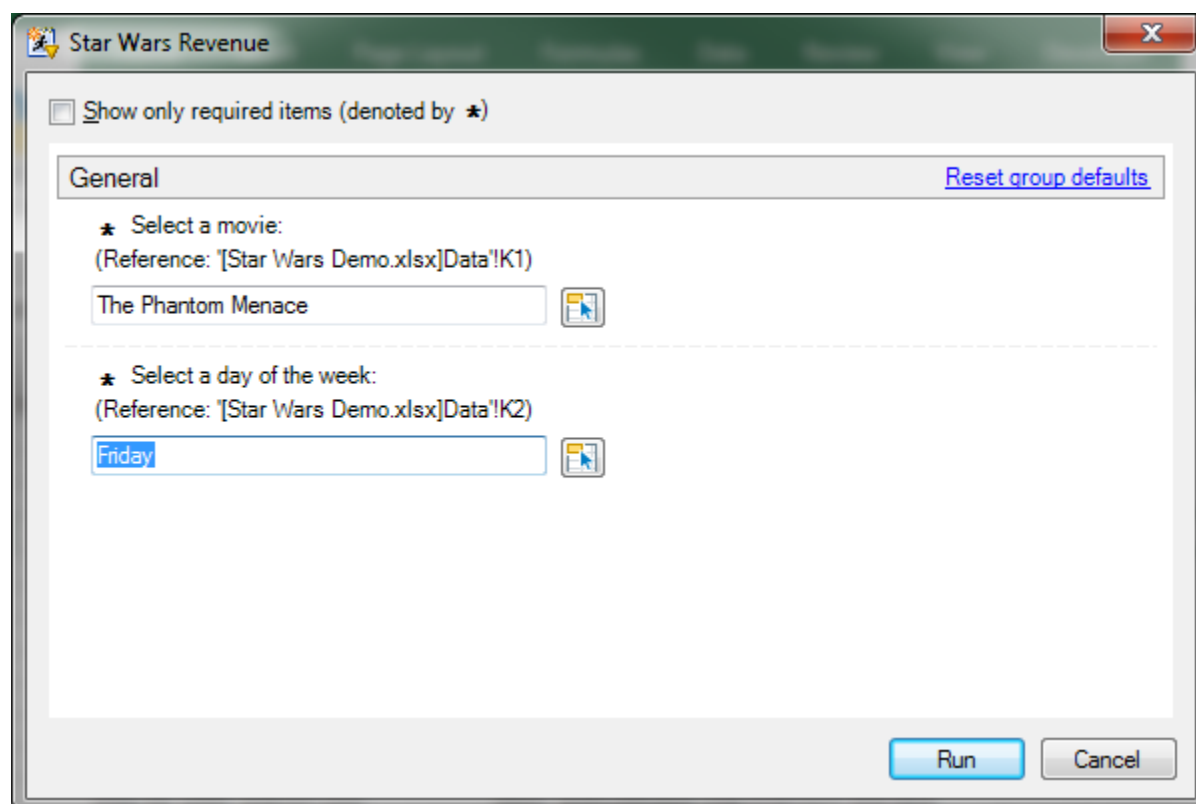
When you run your stored process, you are prompted to provide values for any prompts that are defined in the stored process. There are two ways that you can provide the prompts.

The simplest approach is to enter the value when prompted by the stored process. For more complex prompts, you might be able to select from a list of predetermined values or a specific date from a calendar control. In this example, enter 'The Phantom Menace' for the name of the movie and 'Friday' for the day of the week, and those values are sent to the stored process.

For a more interactive experience, you can link the prompt values to specific cells in the worksheet. This linking is only available for simple numeric, text, or date prompts. When the stored process results are refreshed, the updated cell value is used in the stored process instead of prompting the user for a value.

In this example, the two prompts are for the name of the movie and the day of the week. Add an area on your results page where you can provide those values. Enter 'The Phantom Menace' in cell K1 and 'Friday' in cell K2.

Click  and you are prompted for the cell that contains the value for the prompt. Select the cell in your worksheet. Now, you see a reference to the cell location in the dialog box for the prompts. (See Display 12.) When the stored process runs, it gets the values from that cell.



Display 12. Connecting a Stored Process Prompt to a Specific Excel Cell

Run the stored process, and you see the prompt values used to create the report. Try changing the values in the cells and refreshing the stored process results. You will no longer be prompted, but the new values from the cells are used. You have an interactive worksheet!

Defining an Input Stream

After the prompts have been provided, another dialog box appears that lets you choose the input data, output location, and any output parameters. If the stored process defines an input stream, you must provide one; the stored process cannot run without it. You'll see the name of your input stream, and if you position your mouse pointer over it, a tooltip displays the author's description.

You can enter the range of cells where the data lives, or click  to the right of the text box. A dialog box appears that lets you select the range. If you select a single cell in a large block of data, the selection is expanded to include the entire range. If you select multiple cells, only the selected cells are used. Display 13 shows the selection of the input data for the stored process.

	A	B	C	D	E	F	G	H	I	J
1	Date	Gross	Theaters	GrosstoDate	Movie					
2	May 19, 1999	\$28,542,349	2970	\$28,542,349	The Phantom Menace					
3	May 20, 1999	\$12,307,918	2970	\$40,840,267	The Phantom Menace					
4	May 21, 1999	\$18,467,513	2970	\$59,307,780	The Phantom Menace					
5	May 22, 1999	\$24,414,123	2970	\$83,721,903	The Phantom Menace					
6	May 23, 1999	\$21,929,334	2970	\$105,661,237	The Phantom Menace					
7	May 24, 1999	\$10,881,272	2970	\$116,542,509	The Phantom Menace					
8	May 25, 1999	\$8,181,275	2970	\$1	SAS Add-In					
9	May 26, 1999	\$7,751,876	2970	\$1						
10	May 27, 1999	\$7,719,100	2970	\$1						
11	May 28, 1999	\$13,531,561	3023	\$1						
12	May 29, 1999	\$19,330,780	3023	\$1						
13	May 30, 1999	\$18,537,522	3023	\$1						
14	May 31, 1999	\$15,504,435	3023	\$2						
15	June 1, 1999	\$5,459,637	3023	\$212,558,695	The Phantom Menace					
16	June 2, 1999	\$5,295,848	3023	\$217,854,543	The Phantom Menace					
17	June 3, 1999	\$5,011,928	3023	\$222,866,471	The Phantom Menace					
18	June 4, 1999	\$8,413,783	3024	\$231,280,254	The Phantom Menace					
19	June 5, 1999	\$13,854,324	3024	\$245,134,578	The Phantom Menace					
20	June 6, 1999	\$10,623,546	3024	\$255,758,124	The Phantom Menace					
21	June 7, 1999	\$3,919,371	3024	\$259,677,495	The Phantom Menace					
22	June 8, 1999	\$4,002,249	3024	\$263,679,744	The Phantom Menace					
23	June 9, 1999	\$3,812,157	3024	\$267,491,901	The Phantom Menace					
24	June 10, 1999	\$3,840,149	3024	\$271,332,050	The Phantom Menace					
25	June 11, 1999	\$6,695,326	3024	\$278,027,376	The Phantom Menace					
26	June 12, 1999	\$10,555,346	3024	\$288,582,722	The Phantom Menace					

Display 13. Selecting the Input Data to Use within the Stored Process

Choosing Where to Display the Results

The first time that you run a stored process you are prompted for a location to display the results. You can choose a location in an existing worksheet, or you can display the results in a new worksheet. When choosing the existing worksheet, if you enter a cell address, the cell from the active worksheet is used. If

you click , you can select the cell where you want to display your results.

When you refresh the stored process, the results are displayed in the same location as before. The old results are updated with the new results. Therefore, there is no need to prompt for the results location after the stored process has already been executed. Display 14 shows where you choose the location for your stored process results.

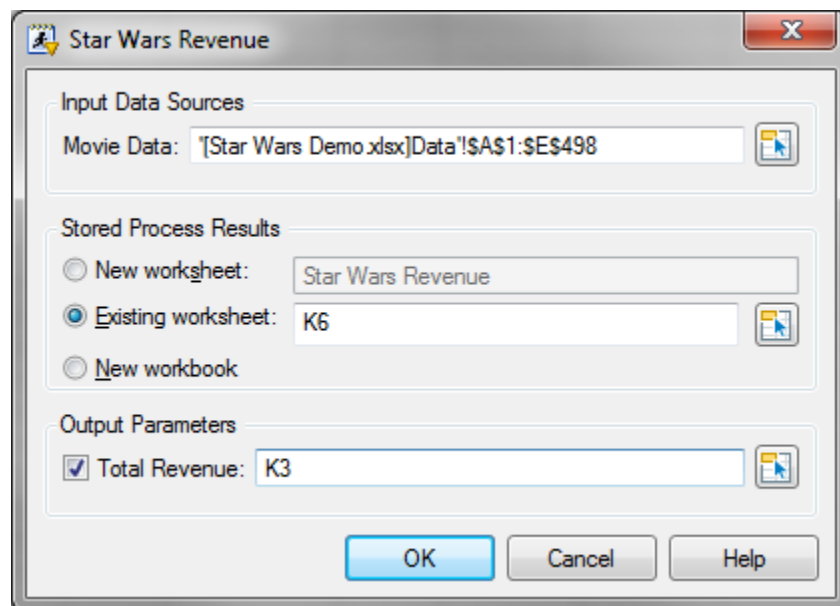
Assigning Output Parameters

An output parameter is a macro variable that is assigned by the stored process. When running a stored process with the SAS Add-In for Microsoft Office, you have the option of choosing whether to include the output parameter as part of the results that are displayed in your worksheet.

In the dialog box where you choose the input stream and output parameters, select the check box for each output parameter that want to include in your worksheet. Then you can select where to place these output parameters. If you simply enter a cell address in the text box, the stored process uses that cell

from the active worksheet. Otherwise, you can click  to open a dialog box that lets you select the cell where you want the output parameter to be displayed.

This example creates an output parameter for the total revenue for a given day of the week. The value for the total revenue from the stored process is written to the cell you provided. You can change the cell containing your prompt values and refresh the stored process, and the value of your output parameter is updated each time you run. Display 14 shows where you define the output parameters to use and where in the worksheet to display their values.



Display 14. Choosing the Results Location and Output Parameters

This is another interactive part of the stored process and the Excel worksheet and enables you to easily explore your data from the Excel workbook.

ADDING VBA CODE TO YOUR WORKBOOK

Now that you have all the pieces from the stored process put together and working in your Excel workbook, you can continue to enhance the user experience by including buttons and event handlers to automate refreshes of your stored process. Microsoft Office has built-in support for Visual Basic for Applications, or VBA for short. You can write visual basic code that can manipulate the workbook and respond to events within the workbook.

The SAS Add-In for Microsoft Office exposes an interface to VBA that allows you to use VBA to interact with your stored process results. To get to the VBA Editor, go to the **Developer** tab on the ribbon and click **Visual Basic**. If the **Developer** tab is not available, right-click on an unused portion of the ribbon and customize your ribbon so that the **Developer** tab is shown. Another option is to simply use Alt+F11 to bring up the VBA Editor directly.

Refreshing Your Results Using VBA

Inside the VBA Editor, you need to choose where to put your VBA code. You can select **Insert > Module** to insert your own module. After you select where to insert your code, you need to add a reference to the SAS object so that the SAS objects are built into the VB editor functions.

Select **Tools > References** to bring up the dialog box. Find the entry for SAS Add-In 7.1 for Microsoft Office. Check the box that corresponds to this entry. Now you're ready to write VBA code. This example creates a procedure that refreshes all of the SAS content in this workbook. Use the **Run** button to run this procedure when you're ready. This is the equivalent of using the refresh button on the **SAS** tab of the ribbon.

```

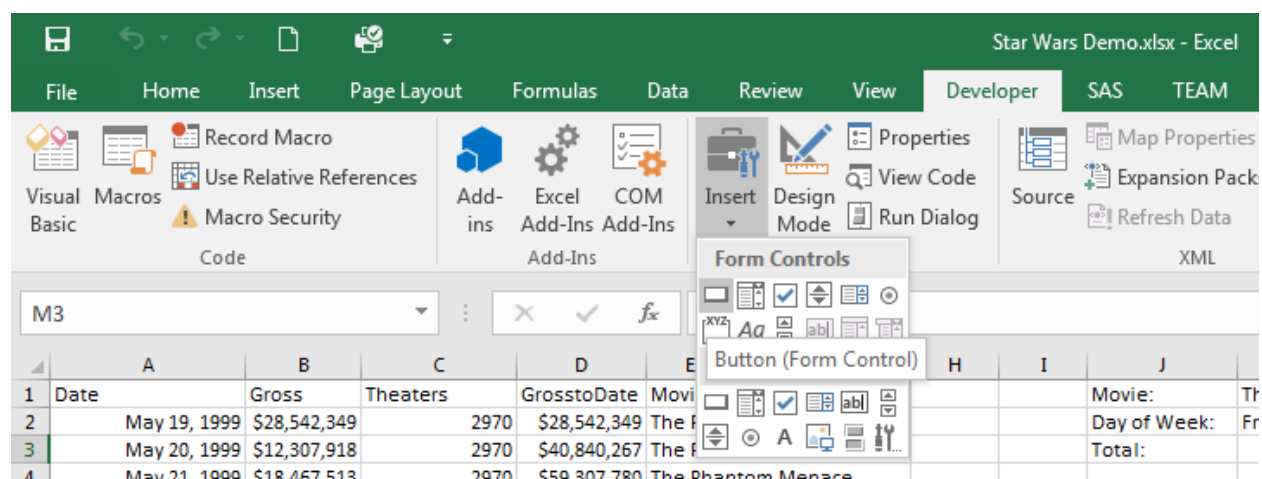
Sub Refresh()
    Dim sas As SASExcelAddIn
    Set sas = Application.COMAddIns.Item("SAS.ExcelAddIn").Object
    sas.Refresh
End Sub

```

For detailed information about all the objects, properties, and methods that are available with the SAS Add-In for Microsoft Office, look in the built-in Help. Go to the **SAS** tab on the ribbon and select **Help > Help for the SAS Add-In for Microsoft Office**. When the Help appears, find the **Automating the SAS Add-In with Visual Basic Code** section in the table of contents. The entire API is documented here, along with examples of VBA code and a section on getting started.

Adding Your Own Button to the Worksheet

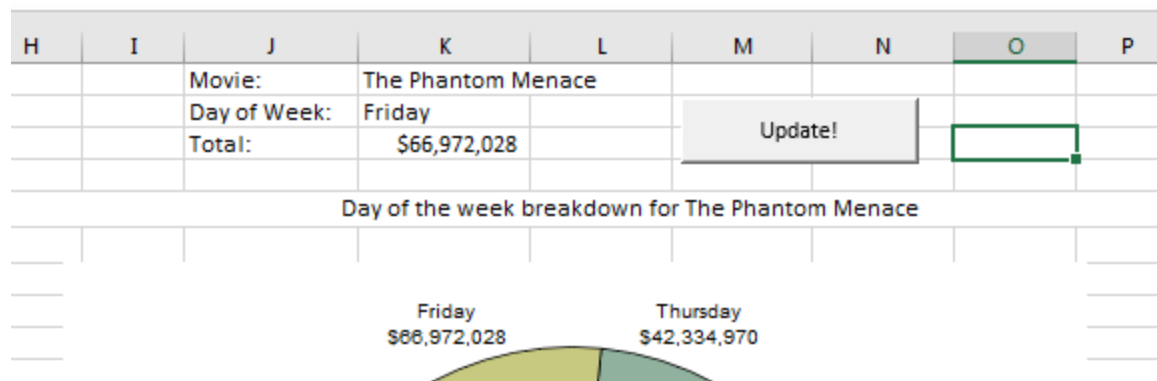
Now that you have the ability to refresh your SAS content with VBA code, you can add your own custom controls to your Microsoft Excel worksheet. Go to the **Developer** tab on the ribbon and select **Insert**. You will see a collection of controls that can be added to your worksheet. Under the **Form Controls** heading, choose the very first item, which is a Button. Display 15 shows the different controls that can be inserted into your Excel document.



Display 15. Inserting a Button from the Developer Tab on the Ribbon

After you select the button control, use the mouse to draw where the button should be placed in the worksheet. When you click the button, you are immediately prompted for a macro to run. In the Assign Macro dialog box, select the **Refresh** macro as the macro to run and click **OK**. You can rename the text on the button after it has been created.

After assigning the macro, change the cell values that are linked to the input prompts. Now click the button on the worksheet, and the results are automatically refreshed. There's no need to even go to the **SAS** tab anymore. Now you're just interacting with the worksheet and getting updated results from SAS. Display 16 shows an example of what your worksheet could look like after adding a button called **Update**.

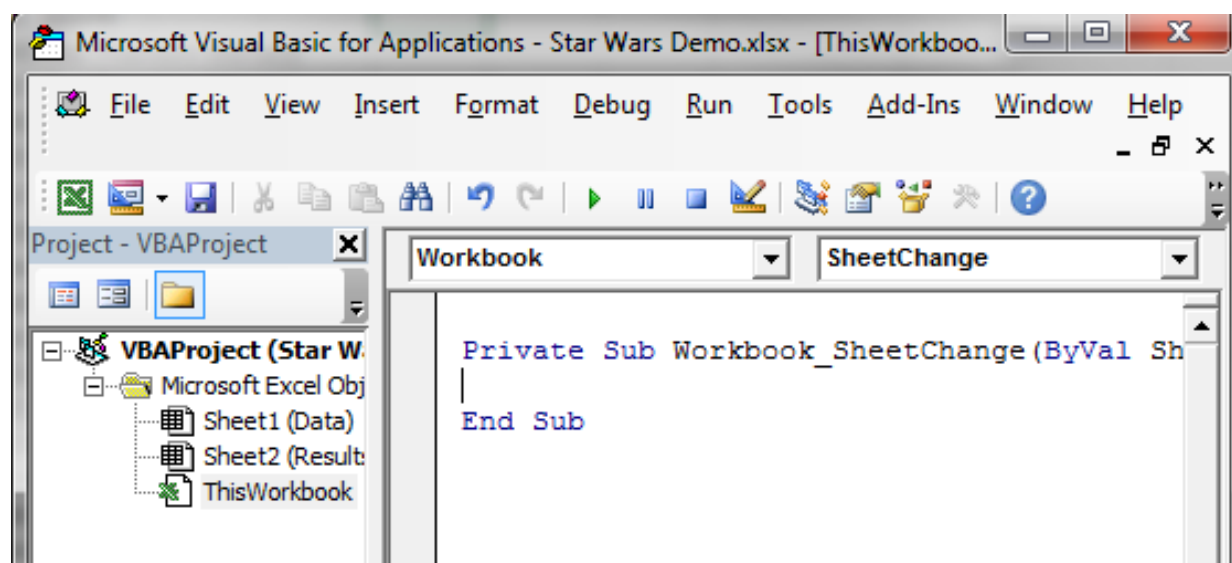


Display 16. A Worksheet with a Button Included

Adding an Event Handler

Instead of making the user click a button to invoke the refresh, wouldn't it be nice if the results refreshed automatically anytime a prompt value changed? This is very easy to do using Visual Basic for Applications.

First, you need to detect when the cell value changes for either of your prompt values. Back in the Visual Basic Editor, go to the **ThisWorkbook** page and choose **Workbook** from the first drop-down list. The next drop-down list displays a list of all the workbook events. The event that you need to listen for is the **SheetChange** event. Display 17 shows where you access the **SheetChange** event in the VBA Editor.



Display 17. Accessing the SheetChange Event in VBA

Selecting **SheetChange** from the **Procedures** drop-down list stubs out the procedure for you. You don't want to call the **Refresh** procedure when any cell changes. That would result in the data being updated too often. You need to add some code to detect which cell changed, and refresh the results only if the cell value that changed intersects with the cells where your prompt values are stored.

```
Private Sub Workbook_SheetChange(ByVal Sh As Object, ByVal Target As Range)
    Dim sheet As Worksheet
    Set sheet = Sh
    If sheet.Name = "Results" Then
        Dim r As Range
        Set r = sheet.Range("K1:K2")
        If (Not (Application.Intersect(Target, r) Is Nothing)) Then
            Refresh
        End If
    End If
End Sub
```

```
End If  
End If  
End Sub
```

Add this code. Now, when you change one of your prompt values in cells K1 or K2, the stored process automatically refreshes, and you have up-to-date information based on your cell selections. Now the worksheet functions almost like an application by itself: taking your inputs, communicating with the SAS server through a stored process, and displaying your results directly in the Microsoft Excel worksheet. You now have a completely interactive experience.

CONCLUSION

Building an interactive worksheet makes it quicker and easier to explore and analyze your data. Whether you are interacting with a SAS Visual Analytics report to visualize your data or running a stored process that displays SAS results in Excel, you have access to all the functionality without the end user having to write code or understand different SAS procedures.

You can focus on the results and interact with them to better understand what the data is telling you and use this information to make better decisions.

REFERENCES

Bailey, David. 2015 "Take Your Data Analysis and Reporting to the Next Level by Combining SAS Office Analytics, SAS Visual Analytics, and SAS Studio." *Proceedings of the SAS Global Forum 2015 Conference*, Cary, NC: SAS Institute Inc. Available <http://support.sas.com/resources/papers/proceedings15/SAS1804-2015.pdf>

Beese, Tim. 2011 "Tips and Tricks for Automating the SAS Add-In for Microsoft Office using Visual Basic for Applications." *Proceedings of the SAS Global Forum 2011 Conference*, Cary, NC: SAS Institute Inc. Available <http://support.sas.com/resources/papers/proceedings11/012-2011.pdf>

Beese, Tim, and Greg Granger. 2012 "Excelling with Excel." *Proceedings of the SAS Global Forum 2012 Conference*, Cary, NC: SAS Institute Inc. Available <http://support.sas.com/resources/papers/proceedings12/036-2012.pdf>

Box Office Mojo. Available <http://www.boxofficemojo.com>. Accessed on December 11, 2015.

CONTACT INFORMATION

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

Tim Beese
100 SAS Campus Dr
Cary, NC 27513
SAS Institute Inc.
tim.beese@sas.com
<http://www.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.