Tricks Using SAS® Add-In for Microsoft Office

Tricia Aanderud, And Data Inc., Raleigh, NC

ABSTRACT

SAS® Add-In for Microsoft Office remains a popular tool for people who are not SAS® programmers due to its easy interface with the SAS servers. In this session, you'll learn some of the many tricks that other organizations use for getting more value out of the tool.

INTRODUCTION

The SAS Add-In for Microsoft Office is sold as part of the Office Analytics and Enterprise Business Intelligence (EBI) solutions. The add-in is designed for users who are familiar with the Office applications but new to SAS. It extends the functionality of Microsoft Excel, Microsoft Word, Microsoft Outlook, and Microsoft PowerPoint by enabling users to access SAS analytics and reporting functionality without any prior programming experience.

In this paper, we will explore how a fictional customer support organization applies the add-in to solve their reporting needs. The organization has over 250 global users and nearly all are able to use reports and dashboards from the EBI and SAS Visual Analytics system to complete their work. However, a smaller portion of the organization always has special reporting needs and they seem to spend all of their time sending requests to IT. This paper provides some tips for helping these users address their needs and ideas for further extending the add-in functionality.

INTRODUCING THE SAS ADD-IN FOR MICROSOFT

SAS Add-In for Microsoft Office uses the Microsoft COM add-in technology to extend the data access, reporting, and analytic capabilities of Microsoft Office suite. Once the SAS Add-In is installed, an integrated SAS toolbar or ribbon is available in the Office environment as shown below. Users must have a metadata user ID with appropriate permissions before they can access the data. Refer to the SAS Intelligence Platform: Security and Administrators guide for more information regarding roles and permissions within SAS metadata.

![Figure 1 SAS Ribbon in MS Excel 2013](image)

SAS TOOLS INTEGRATION WITH MS OFFICE

Each MS Office application offers the functionality in different ways. The following table describes the capabilities of each application. Excel has the most integration with the SAS metadata server and allows users the most interaction.

<table>
<thead>
<tr>
<th>Task</th>
<th>Microsoft Excel</th>
<th>Microsoft Outlook</th>
<th>Microsoft Word</th>
<th>Microsoft PowerPoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open Web Report Studio Reports</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Open OLAP Cubes</td>
<td>Yes</td>
<td>Yes &amp; No*</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Open Information Maps</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>View BI Dashboards</td>
<td>Yes</td>
<td>Yes &amp; No*</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Run SAS Stored Processes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Access SAS Datasets</td>
<td>Yes</td>
<td>Yes**</td>
<td>Yes**</td>
<td>Yes**</td>
</tr>
<tr>
<td>Update SAS Datasets</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Open SAS Visual Analytics Designer Reports</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

*There was a previous experimental feature in version 9.2.
**You can build reports from them.
QUICK OVERVIEW OF THE ADD-IN

Many users want to access corporate databases and data warehouses but fail to realize these tables contain massive amounts of data that would cripple an Excel if opened directly. However, with the add-in, the data processing is accomplished on the SAS server and the much smaller results are delivered to the Office application.

Accessing the Data and Reports

From the SAS ribbon, users, given appropriate access permissions, can access Visual Analytics reports, Web Report Studio Reports, and SAS Stored Processes by clicking the Reports icon, as shown in the following figure. Any prompts for respective reports or stored processes are displayed as well.

Figure 2 Accessing the Reports and Stored Processes

From the SAS ribbon, users, given appropriate access permissions, can access the SAS Server, which contains data information maps, and OLAP cubes, as shown in the following figure.

Figure 3 Accessing the SAS Server

Once users learn how to access the data, they begin to understand why the add-in is so useful. However, there may be some other ways to use the tool that allow the user to be more self-sufficient.
Using the Basic Features

Besides being able to open prompted report or data, the SAS ribbon provides some quick start features that new users appreciate:

- Automatic Chart panel creates various charts based on the data role assignments. This helps new users get started with creating charts.

- Quick Stats panel calculates basic statistics and charts based on the input dataset. This task helps users explore the data and its values.

- Task Gallery shows sample charts that the each task can create. This helps users find the specific task they need because sometimes people have different names from charts or procedures.

Using the Advanced Features

The add-in provides some advanced features that developers (even aspiring ones) might find useful:

- There is an automation interface that allows the user to write scripts using Visual Basic. Tim Beese wrote an extensive paper about this topic "Tips and Techniques for Automating the SAS® Add-In for Microsoft Office with Visual Basic for Applications", which provides examples to get you started.

- While not technically a part of the add-in, SAS does provide a way to work with the Microsoft.Net framework, which is called custom tasks. Chris Hemedinger has written a book about this topic, Creating Custom Add-In Tasks for SAS Enterprise Guide and has examples available at his SAS Dummy blog. Paul Homes of Metacoda created custom tasks that allow you to explore the metadata, which is useful for developers.
WORKING WITH USERS TO SOLVE ISSUES

After you have deployed your business intelligence tool with its fancy data and cool reports, you may think it is finally time to enjoy a beverage and catch-up on your backlog. In many ways, your journey has just begun. Even if you have the most robust reporting system available, there are users who need something different or the data has to be available for a unique purpose.

Figure 6  Example Customer Service Organization

In this paper, you are responsible for supporting the SAS system for the Customer Service department. Customer Service is responsible for supporting the company’s hardware and software products in the field for more than 1,200 worldwide businesses.

The department is divided into three major areas: Technical Support, Special Projects, and Analytics. Technical Support responds to customer issues and logs the issues into Trouble Ticket database. Special Projects is a team of project managers assigned to the larger customers. This team oversees new installations, meets with larger customers to ensure satisfaction, and handles any extremely unhappy customers. The Analytics organization is responsible for creating reports, analyzing systems, and responding to customer requests from the Sales team.

Generally what seems to happen is the standard reporting and dashboards work for 80% of the team. For instance, the technical support managers and team rarely need anything different from a list of issues and performance measurements. Their reports change infrequently. When this team has a request, it is for a new dashboard.

The other departments are working with those outside the organization and their work may not follow a standard practice so they often need atypical reports or the data represented in a different way. Some of these users may not have a SAS programming background and they are more comfortable with the Microsoft Office applications.

Let’s review some common scenarios the users experience and how SAS Add-In for Microsoft Office can resolve their issues.
TIP 1: SUPPORT THE SPECIAL NEEDS REPORTERS

Within the customer support organization, a Special Projects team exists. Their mission is to support high profile customers. Many of the high profile customers require reports in a particular format or with different information. The standard customer reports are not suitable because the customer reports information to a government agency or maybe the customer purchased a unique product and the data does not fit the standard reporting.

In this example, the customer required a monthly report that provided analysis of the installed equipment and supporting processes. Some metrics were standard and similar to what nearly all customers required. However, this customer required detailed analysis of product trouble areas that they could share with their global team. The project manager found that it was easier to write the information in MS Word but she was spending countless hours each month recreating data within MS Excel and then cutting and pasting the data into MS Word and MS PowerPoint. It was time consuming and painful.

SOLUTION – GET IT STRAIGHT FROM THE SOURCE

This report supported a short-term project that required information from the data warehouse. While this report was unique, the overall need for a specialized report is not. Since it is short-term and unique, it doesn’t make sense to engage IT resources for these needs. So the question becomes how does the organization support requests like these without a special team of dedicated staff?

Figure 7 Examples in MS Word and MS PowerPoint

If the data is available but just needs special presentation, the project managers find the Add-In for MS Word or MS PowerPoint is better able to meet their needs. Using the add-in, the project manager would be able to create the graphs and tables once and then refresh the items with new data each month. Then she could write the analysis and create any needed supporting graphs with the Office application easily.

After creating the report in MS Word, the project manager adds graphs or tables using the SAS ribbon. For this analysis, she wants to show how trouble tickets were resolved quicker than the usual. If this was a prototype project, the company would want to ensure the customer the project was receiving their complete focus.

Create Charts and Graphs from Task List

Users can access data and create the charts or tables from the SAS ribbon with just a few clicks. The process is the same for MS Word and MS PowerPoint. Users can drag and drop to resize the object within the application.

To add a line graph to this report, use these steps:

1. From the SAS Ribbon, select the Tasks > Graph > Line Plot.
2. In the Choose Data window, use the Browse button to navigate to the dataset. Use the Filter & Sort button to control the data returned. This report is for the Company XYZ for the past year, so the filter would look similar to this figure.

Tip: Always ensure it is a good query by pressing the Validate button!
3. After selecting the data, you can build the line chart. The task gives you control over the variables, colors, axis, legend, and titles. For quick results, just complete the Data tab and click Run. The results are returned to the application in a box. In the following example, the user reduced the graph to a single column.

Once the report is complete, it's easy to update it each month when the new data is available. With a simple click on the Refresh button, the report is updated. If the user wants to change the filter to have a different date range or even a different customer, she would click the Modify button to change the filter and update the chart.

**Figure 8 Refresh or modify the report from the SAS Ribbon**

**BEHIND THE SCENES WORK**

Depending on the project, you may need to create some views within the database or some SAS datasets that contain the desired data. You may spend more time showing the end user some tricks on how to filter the data than
anything else. Seasoned MS Office users will expect search behavior that is similar to that tool and may need a quick reference or a cheat sheet to assist with using the SAS filters.

**TIP 2: MAINTAIN SMALLER DATASETS FROM EXCEL**

This company is continually changing the organizational structure to meet customer needs. The technical support group is divided by product line and in some cases region. Managers assign a team lead to each group, who is responsible for ensuring trouble tickets are resolved in a timely manner. However, this is not an official position and sometimes team leads are assigned arbitrarily based on customer needs.

Managers want team leads to see the tickets they own. With so much churn it seems impossible to keep the reports current with the organization. It would be great if a single manager could update the situation as it changed.

**CREATIVE SOLUTION-CUSTOM PROMPT DATA**

Many SAS BI users do not realize that you can update SAS datasets through Excel using the add-in. For this solution, you would create a dataset on the SAS server and then allow someone in the organization to update the data as needed. The resulting dataset would be used with a stored process and a prompt to ensure the correct results were returned.

---

**Figure 9 One user updates a SAS dataset for everyone else**

This is a simple example where the team leads are assigned based on product and state. Since we need a way to query the data we create a dataset that contains the `team_lead` name, state, and product name. A prompt is created that refers to the dataset and used to load the prompt values, as you can see in the following figure.

**Figure 10 Using the Prompt TeamLead dataset to support a prompt**
Creating a Stored Process to Support the Prompt

When you build the stored process, you could either join the datasets based on the prompt or create a more elaborate method based on how the report should appear. The following figure shows a very simple example.

```sql
/* === Example Code of using the supplemental dataset ===*/
proc sql;
create table result as
    select * from tickets T1,
          (select * from prompt_teamlead
           where team_lead="&prompt_teamlead.
           ) T2
    where t1.product = t2.product
    and t1.state = t2.state
    and t1.ticket_status = "Active"
QUIT;
proc print data=result; run;
```

Updating the SAS Dataset in MS Excel

When a manager wants to update the Prompt_TeamLead dataset, she would use MS Excel. Use the SAS data button to open the SAS dataset. To make changes, click the Begin Edit button and update the dataset. After the changes are made, click the Commit or End button. The dataset is updated on the server immediately.

**Figure 11 Using MS Excel to edit SAS datasets**

**BEHIND THE SCENES WORK**

To access the dataset directly, the user must have associated save and distribute capabilities granted to her user account through metadata roles. Your organization should consider if all users need this capability since there is nothing to stop a user from opening other datasets and making changes. Within SAS Management Console, review the roles associated with the Add-In for Microsoft Office under the Save or Distribute capability.

When using this method to support a prompt, ensure the user tests the newly added rows to ensure it works properly. Obviously if the manager updates the dataset with a team lead for a product or state that does not exist in the dataset, the stored process may not return any results. So your stored process might need to be robust enough to handle the situation.
**TIP 3: TAKE A SLICE OF THE CUBE**

Users can open OLAP cubes in MS Excel using the SAS Add-In. Cubes have a lot of information available and can be used in a variety of reports. Users may not be aware of all the tricks that a cube can perform just from Excel. In these cases the users copy and paste information or that additional datasets are created to support the questions they have.

**SOLUTION**

Users need to understand some of the ways the cube can be manipulated from within MS Excel to answer the questions they have or how it can be used to create reports. One way to build a simple report is create a slice of the cube.

OLAP Cube Analyzer provides the same functionality as opening the cube with SAS Enterprise Guide. The team can easily explore data and trends. The OLAP Cube Analyzer allows you to build charts and graphs, but sometimes different charts are needed. In these cases, the user can get a slice of the cube and build any chart they please with native Excel functionality.

In this example, the user created the drill path to show the overdue tickets for the customer. Once the data the user wants to plot is showing, she selects the Create Slice button to open the data in an Excel table. From here the new worksheet can be used as desired.

![Figure 12 Create a slice of the cube for other purposes](image)

**BEHIND THE SCENES WORK**

This is another situation where you may need more training for the users on how to manipulate the cube. Users can also create custom filters and measures to assist with some of the metrics they may need to supply to managers or customers. Once the users understand the cube capabilities they can make better use of it in their work.

**TIP 4: ALLOW SOME EASY AD-HOC REPORTING**

Many times the users want similar information from the dataset but just want to see it with a slightly different filter. The project manager only wants to see information about a particular customer or a sales manager wants to know all trouble tickets open on a specific release in customer sites. One thing these users have in common that they want to analyze the data in Excel.

**SOLUTION - WHEN THIS WORKS**

If you have the information summarized elsewhere then you want to encourage the user to that path – especially if it’s 95% what they want. However, there are times when the data is not accessed that often and you can write a custom query to the database to handle it.
HOW DO YOU HANDLE IT

How about writing a single stored process that has multiple prompts that allows the user to consume the information as they desire? The beauty of this method is that you do not have to maintain multiple stored processes and multiple users can get multiple uses from a single stored process.

You might determine that all the users really want is an updated list that can access from Excel. If you examine the user requests you learn that they all want the same columns from the table they just want different filters. Here’s how the prompts might look:

![Figure 13 Prompt the User]

You can change the prompts to be dependent. For instance, in the above example it would probably be smarter to ask the user to select a Product Name and then allow the software release to be selected based on the chosen product.

**Write a Stored Process**

Write one stored process with a dynamic where statement. The where clause can be as fancy you want. In the following figure, you can see the basic logic flow for the stored process. The key is writing a dynamic where clause that builds the where clause for you. The resulting dataset is tested for returned rows and takes appropriate action.

![Figure 14 Dynamic where clause process flow]
Here’s how the code might look. The basic idea is that you test each prompt variable to see if it’s populated and then append the value to the where clause macro variable.

**EXAMPLE STORED PROCESS CODE FOR DYNAMIC WHERE CLAUSE**

```sas
%macro BuildReport;
Title;footnote;
    %global where_stmt;
%let where_stmt = 1; /*Start the statement with something to resolve*/
%if %length(&customerPrompt) > 0 %then %do;
    %let where_stmt = &where_stmt and customer = "&customerPrompt";
%end;
%if %length(&TicketNumberPrompt) > 0 %then %do;
    %let where_stmt = &where_stmt and ticket_num = &TicketNumberPrompt;
%end;
%if %length(&DaysOpenPrompt) > 0 %then %do;
    %let where_stmt = &where_stmt and DaysOpenCnt = &DaysOpenPrompt;
%end;

Note: You need an IF/THEN statement for each prompt you created
%put WHERE STMT IS: &where_stmt.;
proc sql;
    create table temp as select * from service.tickets
    where &where_stmt.;
quit;
%if &sqlobs gt 0 %then %do; /*Test if any values were returned*/
    /Don’t use a titles or footnotes since it exports to Excel
    You want the values to start the first row*/
    proc print data=temp noobs;
    run;
%end;
%else %do; /* Let the user know what happened */
    Data temp; MSG="No values found. Try Again."; run;
    Title “Results not Found”;
    /*You could list the prompts with choices if you wanted*/
    proc print data=temp noobs;run;
%end;
%mend;

%STPBEGIN;
%BuildReport;
%STPEND;
```

*Figure 15 Code for Dynamic Where Clause*
TIP 5: ALLOW SOME PROJECT DASHBOARD SUPPORT

One of the project managers in the organization works with troubled customers mostly. These customers have purchased prototype software or otherwise have a special installation, which means the projects require a lot of collaboration with the Engineering group. It is not unusual for customers become agitated at the process due to the compressed schedules and multiple field issues.

The project manager wants to remind the customer that even though the process is frustrating, the company is responding and the project is moving forward. He wants to show the ticket count and resolution rate at the top of the weekly action list that he manages from Excel and use during the video conference call.

ISSUE

Since the project manager is using Excel to handle the list of activities, this is also where he wants to display the graph. What we need to do is marry our SAS data with the Excel charts. We can accomplish this with a stored process, a dataset, and the Excel chart functions.

**SOLUTION – USE A BACKGROUND PROCESS**

After the user has created the Action Register worksheet, he can add another worksheet (such as Summary Data) that populates the first worksheet (shown in Figure 16). In the Summary Data worksheet he has added a SAS dataset or the results from a SAS stored process (such as the one created in the last topic) to have the data needed for the charts. As shown in the following figure, the chart is created from the Summary Data tab.

**Figure 16 Project management action register spreadsheet**

**Figure 17 Create the Excel chart from SAS Data**
Each week, he could update the stored process or SAS dataset, which in turn would update the Action Register report. The workflow is shown in the following figure.

**Figure 18 Update Process**

**BEHIND THE SCENES**

It’s less work when a dataset or stored processes exists to support these tasks. However if that is not the case then you might need to create a stored process to assist with special needs. One stored process that can support many one-offs makes it easier for everyone.

**TIP 6: BRING YOUR OWN DATA**

One of the data analysts receives spreadsheets with various data from the sales team about the customers. For instance, sometimes the data is about system performance and sometimes the customer has been tracking events. As you can imagine, the data contains a variety of columns because it originates from different people for diverse situations. At times, the data is incomplete, perhaps it doesn’t contain useful information for analysis or it’s missing values or something similar.

The data analyst wants to review the data before loading it into SAS Enterprise Guide so she can give feedback to the sales representative quickly.

**SOLUTION – TAKE A QUICK LOOK**

Many users think they can only use data from the SAS server with the SAS tasks, however, the SAS task can also use data from Excel spreadsheet. There are more than 80 tasks available.

For this example, the data analyst wanted to characterize the data so she could learn about its contents. Here’s the process:

1. Select the data you want to analyze in the spreadsheet.
2. Select the task from the Task Icon. In this case, the user selected Describe>Characterize Data. This task provides a lot of summary statistics about each variable so you can understand it better.
3. Select where the results should be located. A new worksheet seems like a logical choice but you can place the results in the same worksheet or a completely new spreadsheet if you want.
4. Each task requires different information and but the user is lead through the steps (or interview process) so she can select the variables, the desired output, and other choices.
5. After completing the steps, click Finish to generate the results. The results are placed where indicated.
**TIP 7: USE THE INBOX**

Some users seem to live in their Inbox. The Radicati Group noted that email is still the go-to communication form for business users and they expect it to continue growing by 5% over the next 4 years (Sara Radicati, 2013). When you think of users who send a lot of email – executives, managers and salespeople come to mind first. These users are often a hub for others in the organization and often distribute information between groups. However, these users may be less frequent users but you want to encourage them to access the reports and tools without a lot of effort.

Also these users may complain that it’s too much effort to login and search through so much information for the few items they want to monitor and request that reports are emailed to them each morning.

**SOLUTION – BRING THE REPORTS TO THEM**

It makes sense that any functionality added to Microsoft Outlook that makes the reports and dashboards more easily accessible would be desirable to this user population. Since you want to encourage user adoption and usage, teach these users how to use Outlook to gather, monitor, and distribute information. The Add-In for Outlook allows users to view and share content through SAS Central.

**SAS Central Provides Access to Reports**

From the SAS Central folder, users can see all the SAS Visual Analytics, Web Report Studio and Stored Processes available. The user can generate the reports and then refresh the information as needed. This might be useful for a salesperson to generate a report that he can email to the customer after a meeting or for a manager who wants to review information.
SHARING COMMENTS

Along with viewing and refreshing reports, the users can share the report with other Office applications or even forward the report to other users. Another nice feature is the availability share comments about the reports. Instead of having emails that may be hidden from other users or easily deleted, the comments can be stored along with the report.

The Comment pane is beneath each report and users with access can ask or respond to questions. This is especially useful for cross department discussions and helpful to review during meetings. The comments are also stored with Web Report Studio.

Figure 20 Leaving comments on reports

BEHIND THE SCENES

This solution is based on the metadata contents and user permissions. Once the user has permission to review the metadata, the add-in shows what is registered and allows access. As the reports are updated, the new information is available. There is little for a support person to do.

For more details about using the add-in with Outlook, refer to the user documentation or the “A New Outlook into Your Business” by Anand Chitale and Mike Barnhart.

REFERENCES


RECOMMENDED READING

SAS User Documentation
- Administering the SAS Add-In for Microsoft Office in the SAS 9.4 Intelligence Platform: Desktop Application Administration Guide.
- SAS Intelligence Platform: Security and Administrators
- SAS Add-In 6.1 for Microsoft Office: Getting Started in Microsoft Excel, Microsoft Word, and Microsoft PowerPoint.

Related Books


Related Sites/Blogs
• BI Notes for SAS Users (http://www.bi-notes.com)
• Real BI for Real Users (http://blogs.sas.com/content/bi/)
• Metacoda (http://metacoda.com)
• SAS Dummy (http://blogs.sas.com/content/sasdummy/)

ACKNOWLEDGEMENTS
The author would like to thank all of those who have provided ideas, proofreading, and commentary in the past year, which includes Chris Hemedinger, Paul Homes, Michelle Homes, I-Kong Fu, Ken Aanderud and Steve Overton. Any errors or omissions may be credited to me alone.

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

    Tricia Aanderud
    And Data Inc
    tricia.aanderud@and-data.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.