Paper 1765-2014

Geo Reporting: Integrating ArcGIS Maps in SAS® Reports

Nathan Clausen, CACI; Aaron House, CACI

ABSTRACT

This paper shares our experience integrating two leading data analytics and GIS software products - SAS® and ArcGIS - to provide integrated reporting capabilities. SAS is a powerful tool for data manipulation and statistical analysis. ArcGIS is a powerful tool for analyzing data spatially and presenting complex cartographic representations. Combining statistical data analytics and Geographic Information Systems (GIS) provides increased insight into data and allow for new and creative ways of visualizing the results. While products exist to facilitate the sharing of data between SAS and ArcGIS, there are no ready-made solutions for integrating the output of these two tools in a dynamic and automated way. Our approach leveraged the individual strengths of SAS and ArcGIS, as well as the report delivery infrastructure of SAS Information Delivery Portal.

INTRODUCTION

While SAS has significantly better data analytics tools, ArcGIS is considered the industry standard in Geographic Information Systems (GIS) and map production. GIS is a constantly growing and changing field, and as this develops there is an ever increasing need to access data from different sources. SAS is already a common tool for compiling and manipulating large volumes of data with its own analysis capabilities. We have linked these programs in a number of ways, ranging from the SAS-Esri Bridge and the ArcGIS OLE DB connections to SAS datasets. However, these have either been too slow, in the case of the latter, or designed for a different[1] (and more expensive) platform configuration than what was being used[2]. Accessing SAS data from ArcGIS was the first technical challenge in the merging of these two analytical tools. The second challenge was being able to combine the analytical output of each tool into unified reports that took advantage of each product's relative strengths.

We have developed a solution that leveraged SAS's ability to prepare and export whole or modified datasets in a variety of formats, and then generate and call a system command to access ArcGIS's external geoprocessing capabilities. The key that links these two products together is ArcPy, an ArcGIS Python scripting module that accesses ArcGIS functionality. By exporting SAS data to a temp directory then calling an ArcPy script, we produced maps by modifying and processing the most up-to-date data into a geospatial format. Finished map products were output to the same temp directory in any one of a number of common image formats, that SAS then imported into a report and combine with further graphic and tabular analysis generated from the same data. This combined approach produced complex automated reports in minutes.

THERE AND BACK AGAIN: THE SAS-ARCGIS SOLUTION CYCLE

To provide a maximum of programming flexibility and reusability, we relied heavily on convention and wrapped all of the SAS-ArcGIS interactions in SAS Macro code. An overview of the report generation process flow is shown in Figure 1.

[1] ArcGIS Server

[2] Our configuration is based on a local installation of ArcGIS Desktop on the SAS Application Server. This installation includes a Python executable file and the ArcPy libraries.

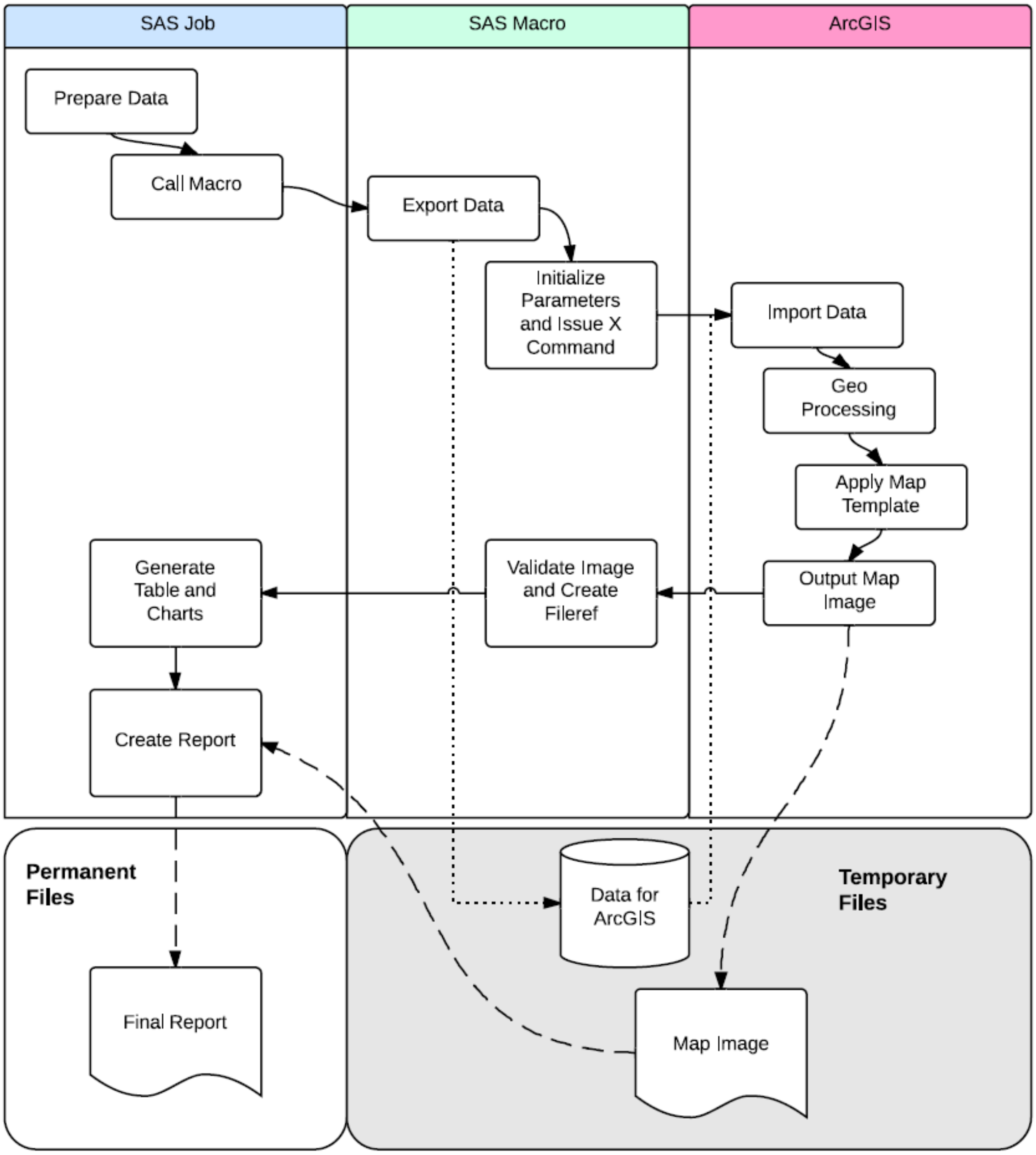

Figure 1. Overview of the report generation process

1. **Data Preparation.** Report data is prepared using SAS code. In our example, this is a stored process that requests users to provide an event date range and optional region of interest. All data, including any produced by ArcGIS, is stored in the temporary work directory to lessen the storage overhead.

2. **Macro Initiation.** After the filtered and formatted dataset has been created, the mapping macro is called. This macro has several parameters including:

   - the name and path of the dataset prepared in step 1.
   - parameters to identify which map which ArcGIS should generate (Example: State, County, etc.).
   - the name of the map image output file.

3. **Calling the Python Script.** The macro code exports the SAS data in an Access Database format and constructs a system call based on the parameters provided in step 2. In addition to the location of the exported data, SAS must also communicate to ArcGIS the expected location of the output file and other parameters needed to create the map.

4. **ArcGIS Image Creation.** The Python script, using the provided parameters, ingests the Access Database and converts it to a geospatial format called a shapefile. This format is able to hold spatial information based on coordinates within the provided database table, as well as the table itself in a dBase format. The data undergoes any necessary geoprocessing (conversion to different data types, kernel density generation, distance calculations, etc.) at this time using ArcGIS's Python Module, ArcPy. The outputs from these processes are used to populate an ArcGIS Map Document that has been pre-prepared. This document holds empty layers which are essentially views of the geospatial data. As the final stage of this step, these layers are sourced to the spatial data created above and the final map is exported, as an image, to the output directory provided by SAS.
5. **Return to SAS.** When the python script has completed, the macro validates that the image output file exists and creates a fileref to point to it. At this point, the macro code has completed and the process returns to the SAS code.
6. **Including the Map Image in a Report.** Finally, a report is created referencing the file name created in the macro. We used proc gslide with the iback goption to pull the map image into a report generated with proc greplay, as shown in the code below:

```
%generateMap(dir=&tempDir, img=&imageName, ied=RCIED);
goptions reset=all nodisplay device=png300 ymax=3.4in xmax=4.4in;
goptions iback="&tempDir.\&imageName..png" imagestyle=fit;
proc gslide; run; quit;
```

… additional report generation and ODS destination setup …

```
proc greplay igout=work.gsreg nofs tc=sashelp.templt template12r2;
  treplay 1:1;
run; quit;
```

TIPS AND SUGGESTIONS

In the process of developing and refining this approach to generating and combining ArcGIS maps with SAS reporting output, we have identified several best practices and compiled a few tips and suggestions for those interested in applying this approach to their own project.

Categories and Coordinates. SAS needs categorical variables to filter and group geographic data. To report on a geographical region, say Narnia, having a "Region" variable with values like "Narnia", and "Mordor" makes it easy for SAS to quickly identify that subset of relevant data measures for reporting. ArcGIS, on the other hand, is able to work quickly with latitude and longitude values to subset the relevant records. Using Proc GINSIDE to define category variables on the dataset provides the variables each tool needs.

Temporary File Handoff. The data files that SAS passes to ArcGIS, and the image files that ArcGIS returns need to be stored somewhere on a common file system, accessible by each process. When running multiple concurrent reporting processes, it is important that one process does not overwrite a file needed by another process. It is also important to make sure that temporary files do not build up over time wasting storage space. Our practice to avoid any of these issues was to rely on the SAS temporary work directory. The path to this directory can be referenced throughout the SAS code by setting a macro variable using the following code:

```
%let tempDir=%sysfunc(pathname(work));
```

Some key benefits to using this path are that it cannot have permission errors for the current SAS session, it isolates the current users output from other users, and all content is automatically removed when the SAS sessions exists.

A Map by Any Other Name. Depending on the context of the reporting being created, it may be useful to include more than one map. For example, comparing two countries, detailing a district within a region, or showing one area for two periods of time. The same Macro code can be used to generate multiple maps for a single report, if the map image name is derived from macro parameters. For simplicity, we included an 'imageName' macro variable that tells ArcPy what to name the final image.

Titles and Annotations. If a picture is worth a thousand words, a title on that picture can answer a thousand questions. Fortunately, PROC GSLIDE works well with SAS title and footnote statements, as well as other annotations, allowing for the placement of dynamic text on the map image.

CONCLUSION

When creating reports integrating less complex ArcGIS map output, we found report generation time to be usually less than a minute and often under 30 seconds. This allows the combined output reports to be on-demand through Web reports or Stored Processes reports. However, more complex maps can often take several minutes to create, making static HTML or PDF reporting destinations more appropriate. Also, it is often desirable to have a static version of a dynamic report to distribute to a wider audience via email, for example. By leveraging SAS ODS, no modification is needed for the map-generating macro code, and very little, if any, additional work is needed to send the final report to additional destinations. Finally, this method opens up the entire suite of ArcGIS geoprocessing capabilities to use by SAS.

CONTACT INFORMATION

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

Name: Nathan Clausen
Email: nclausen@caci.com
Name: Aaron House
Email: ahouse@caci.com