

## Data Visualization from SAS® to Google Maps on Microsoft SharePoint

Xiaogang (Isaac) Tang, Wyndham Worldwide

### ABSTRACT

Google Maps is a very popular web mapping service developed by Google. Microsoft SharePoint is a popular web application platform and used for content management by companies and organizations. Connecting SAS with Google Maps and SharePoint combines the power of these three into one. As a continuation of my SAS Paper 1062-2017 “Data Visualization from SAS® to Microsoft SharePoint”, this paper expands on how to implement geocoding and data visualization from SAS to Google Maps on Microsoft SharePoint. The paper shows users how to use SAS Procedures to create and send XML data files from SAS to SharePoint Document Library. The XML data files serve as data feeds for Google Maps web pages on SharePoint and SAS code example is included. A couple of examples with different views on data visualization from SAS to Google Maps API on SharePoint are provided.

### INTRODUCTION

Google Maps is a web mapping service developed by Google. SharePoint is a popular web application framework and platform developed by Microsoft, widely used by large companies and organizations. SAS is highly proficient for data processing and analysis. SharePoint is convenient to share content and manage user permissions. Connecting SAS with Google Maps and SharePoint combines the power of these three into one.

In 2009, a SAS Global Forum Paper titled “Integrating SAS® Business Intelligence with Microsoft SharePoint” described a variety of approaches to integrate the two technologies.

SAS paper 11520-2016 “Releasing the Power of SAS® into Microsoft SharePoint” by me presents a simple and user-friendly solution to send SAS contents to SharePoint via an existing email system. SharePoint Document Library can be configured to assign a user-defined corporate email address. PDF reports and Excel files can be sent from a SAS server to user email addresses through a corporate email. By combining these two techniques, users can write a SAS program to send emails with SAS contents to a specific SharePoint Document Library with user-defined corporate email address.

SAS paper 1062-2017 “Data Visualization from SAS® to Microsoft SharePoint” by me expands on how to implement data visualization from SAS to Microsoft SharePoint. The paper shows users how to use SAS Procedures and SAS Output Delivery System (ODS) to create and send visualization output files from SAS to SharePoint Document Library. Several SAS code examples and output screenshots are included to show how to create tables, charts, plots and maps from SAS to SharePoint.

As a continuation of the two papers, this paper expands on how to implement geocoding and data visualization from SAS to Google Maps on Microsoft SharePoint. The paper shows users how to use SAS Procedures to create and send XML data files from SAS to SharePoint Document Library. The XML data files serve as data feeds for Google Maps web pages on SharePoint.

Firstly, this paper will review how to configure SharePoint to accept contents from SAS.

Secondly, the paper will present a couple of examples with different views on data visualization from SAS to Google Maps API on SharePoint.

- Example 1: Displaying hundreds of locations on Google Maps. SAS create the data feed and send it to SharePoint. Google Maps web page and data file are separate files. This approach is good for modularization. It can be applied to business reporting with daily refreshed data feeds.
- Example 2: Displaying three locations on Google Maps, with multimedia Info Windows. SAS creates the html file and sends it to SharePoint. Data are saved within Google Maps web page. This approach doesn't need to manually create html file and upload it to SharePoint.

## CONFIGURE SHAREPOINT TO ACCEPT CONTENTS FROM SAS

Here are the steps to configure SharePoint:

1. Create a SharePoint Document Library. Please check with your SharePoint administrator to make sure you have the right permissions to do so. You can also use an existing SharePoint Document Library.
2. Open the SharePoint Document Library. Go to “Library Tools” Tab - “Library”, click the “Library Settings” button. In the “Document Library Settings”, under “Communications” you can find “Incoming e-mail settings”. Open it. Figure 1 is the screenshot for “Incoming e-mail settings”.
3. You may follow this example to set up your library. You need to assign an email address to the library, so SAS can send email to SharePoint. In this mock-up example, [rm\\_viz@mycompany.com](mailto:rm_viz@mycompany.com) is the email address.

Tip: you may send some test emails with attachments from Outlook to the email address ([rm\\_viz@mycompany.com](mailto:rm_viz@mycompany.com)) to make sure it's properly set up, before we move on to the SAS code part.

The screenshot shows the 'Incoming E-Mail' settings dialog box in SharePoint. It is divided into several sections:

- Incoming E-Mail:** Contains a description and a checkbox 'Allow this document library to receive e-mail?' with 'Yes' selected. Below it is an 'E-mail address:' field with 'rm\_viz' entered in a text box and '@:...' in a dropdown menu.
- E-Mail Attachments:** Contains a description and a section 'Group attachments in folders?' with three radio button options: 'Save all attachments in root folder' (selected), 'Save all attachments in folders grouped by e-mail subject', and 'Save all attachments in folders grouped by e-mail sender'. Below this is 'Overwrite files with the same name?' with 'Yes' selected.
- E-Mail Message:** Contains a description and a checkbox 'Save original e-mail?' with 'No' selected.
- E-Mail Meeting Invitations:** Contains a description and a checkbox 'Save meeting invitations?' with 'No' selected.
- E-Mail Security:** Contains a description and a section 'E-mail security policy:' with two radio button options: 'Accept e-mail messages based on document library permissions' (selected) and 'Accept e-mail messages from any sender'.

At the bottom right, there are 'OK' and 'Cancel' buttons.

**Figure 1. Screenshot for SharePoint Library – Incoming e-mail settings**

You can send a variety of file formats (PDF, Excel, CSV, DBF, HTML, JPEG, GIF, etc.) as attachments to SharePoint. Please note that SharePoint does have a size limit (for example, 5MB) for incoming email attachments. Please check with your SharePoint admin to find out the actual limit.

Assume you have successfully sent email from your Outlook to the SharePoint Document Library via the email address [rm\\_viz@mycompany.com](mailto:rm_viz@mycompany.com), and let's also assume your own email address is [myname@mycompany.com](mailto:myname@mycompany.com). Let's continue to the SAS coding part.

All these examples are using tables from sashelp library. Sashelp is a native SAS library. SAS provides over 200 data sets in the sashelp library. For more information, you may check out the official SAS documentation “SAS Help Data Sets”.

## EXAMPLE 1: DISPLAYING LOCATIONS WITH LATITUDE AND LONGITUDE

It will be good to know the basics of Google Maps APIs, though it's not required to become an expert to apply this example. Please see <https://developers.google.com/maps/>. API means "application programming interface" which is a set of subroutine definitions, protocols, and tools for building application software. By using the Google Maps API, it is possible to embed Google Maps page into an external website, onto which site specific data can be overlaid.

Data mapping starts with points. This example is to illustrate how to display points on Google Maps using SAS as data source. The example data set is from sashelp.zipcode. The sashelp.zipcode data set provides US zip codes. The data set contains 41,252 observations. Each row is a unique zip code, X, Y is the longitude and latitude of the centroid of ZIP code region.

```
Sashelp.zipcode --- US Zipcodes; Source: Zipcodedownload.com Jul 2013, SAS Release 9.4

                                The CONTENTS Procedure

                                Variables in Creation Order

# Variable  Type Len Format Label

 1 CITY2     Char  64          Clean CITY name for geocoding
 2 STATENAME Char  64          Clean STATENAME for geocoding
 2
 3 ZIP       Num    8 Z5.        The 5-digit ZIP Code
 4 Y         Num    8 11.6      Latitude (degrees) of the center (centroid) of ZIP Code.
 5 X         Num    8 11.6      Longitude (degrees) of the center (centroid) of ZIP Code.
 6 ZIP_CLASS Char   1          ZIP Code Classification:P=PO Box U=Unique zip used for
                        large orgs/businesses/bldgs Blank=Standard/non-unique
 7 CITY      Char  35          Name of city/org
 8 STATE     Num    8          Two-digit number (FIPS code) for state/territory
 9 STATECODE Char   2          Two-letter abbrev. for state name.
10 STATENAME Char  25          Full name of state/territory
11 COUNTY    Num    8          FIPS county code.
12 COUNTYNM  Char  25          Name of county/parish.
13 MSA       Num    8          Metro Statistical Area code by common
                        pop-pre 2003; no MSA for rural
14 AREACODE  Num    8          Single Area Code for ZIP Code.
15 AREACODES Char  12          Multiple Area Codes for ZIP Code.
16 TIMEZONE  Char   9          Time Zone for ZIP Code.
17 GMTOFFSET Num    8          Diff (hrs) between GMT and time zone for ZIP Code
18 DST       Char   1          ZIP Code obeys Daylight Savings: Y-Yes N-No
19 PONAME    Char  35          USPS Post Office Name: same as City
20 ALIAS_    Char 300          USPS - alternate names of city separated by ||
  CITY
21 ALIAS_    Char 300          Local - alternate names of city separated by ||
  CITYN
```

Figure 2. Data structure of sashelp.zipcode.

This example shows all 649 zip code regions' center points (centroids) in the State of Colorado on Google Maps. Let's see the following figures for the screenshot output on the Google Maps. You can move around to change the map view, zoom in/zoom out, switch between Map view and Satellite view. You can drop the right corner little man icon to see the street view. You can click on any zip code point to see the zip code and the city name. You can do a [Ctrl] + [F] search to find the zip code (say, 80299) from the side bar, click the zip code, the point with info window will be displayed on the map.

For Map Example 1 in this paper, it's inspired by example 1 (sidebar with a scroll bar) in Mike Williams's web page <http://econym.org.uk/gmap/basic15.htm>. Please see the sample code of this paper to find the example1\_CO.html file, co\_zipcode.txt xml file and the related SAS code.

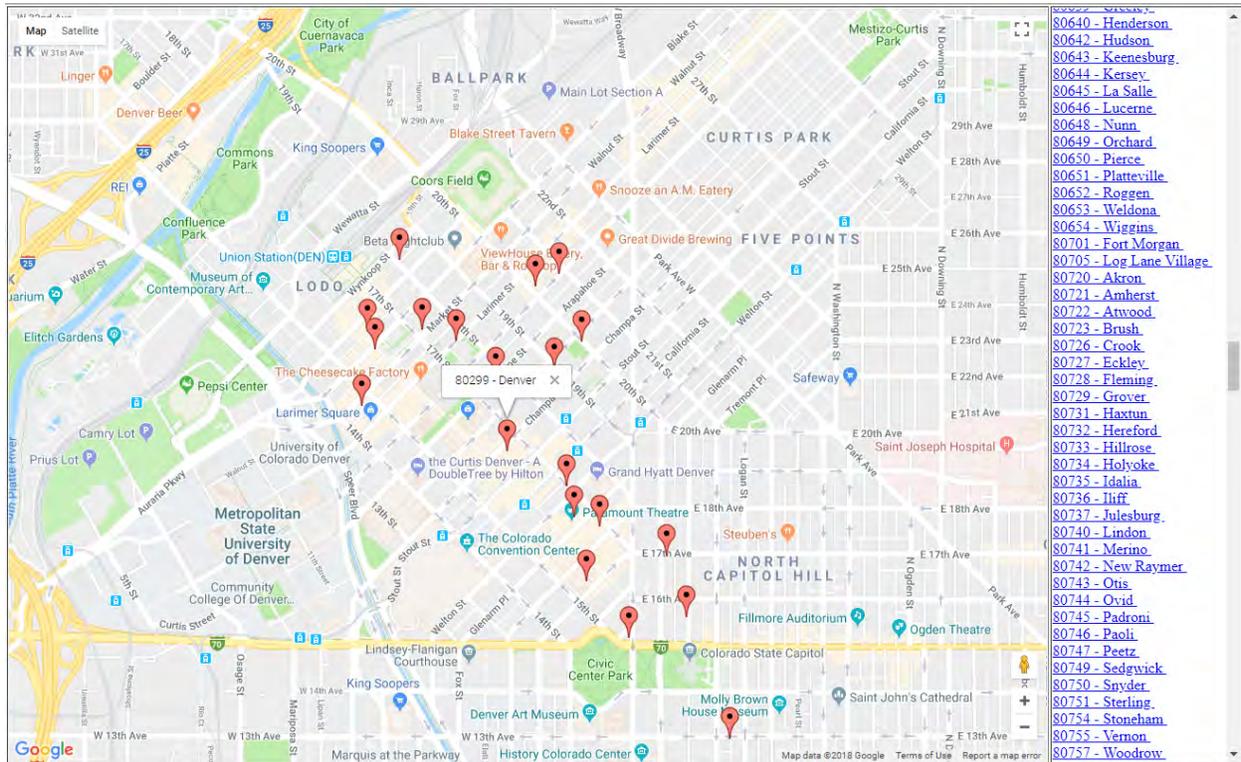
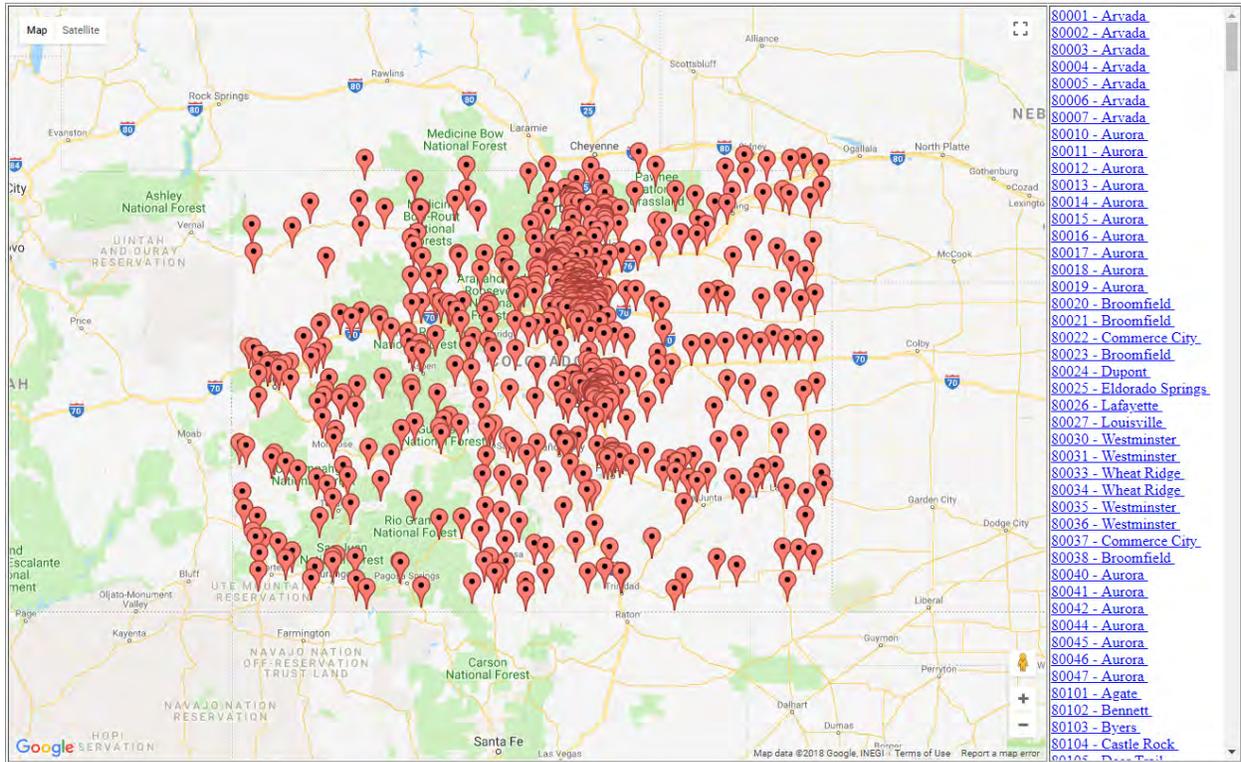


Figure 3a, 3b. Screenshots of Google Maps – example1\_CO.html. Displaying centers of zip code regions in State of Colorado.

Here are the data input in the example1\_CO.html file:

```
Example1_CO.html Line 69: GDownloadUrl("co_zipcode.txt", function (doc) {
```

co\_zipcode.txt is the XML format data feed for Google Maps, which contains all 649 zip code center points with latitude and longitude.

We will use SAS to create co\_zipcode.txt and send it from SAS to SharePoint.

Below is the first 4 lines and last 4 lines of co\_zipcode.txt. The fields: for each point marker, lat means latitude, lng means longitude, html is the text shown in the sidebar, label is the text shown in the info window on Google Maps.

#### co\_zipcode.txt

```
<markers>
<marker lat='39.801977' lng='-105.087152' html='80001 - Arvada' label='80001
- Arvada' />
<marker lat='39.794577' lng='-105.095896' html='80002 - Arvada' label='80002
- Arvada' />
<marker lat='39.824737' lng='-105.061827' html='80003 - Arvada' label='80003
- Arvada' />
...
<marker lat='39.269418' lng='-106.863474' html='81656 - Woody Creek'
label='81656 - Woody Creek' />
<marker lat='39.650220' lng='-106.360740' html='81657 - Vail' label='81657 -
Vail' />
<marker lat='39.621866' lng='-106.500236' html='81658 - Vail' label='81658 -
Vail' />
</markers>
```

The initial view of Google Maps will need to set the center point as (39.1, -105.6). Map zoom level is 7 in this example. For more details about zoom level, please refer to <https://developers.google.com/maps/documentation/static-maps/intro#Zoomlevels>.

```
Example1_CO.html Line 64: map.setCenter(new GLatLng( 39.1, -105.6), 7);
```

Here is the SAS Code for example 1.

#### SAS Code for example 1:

```
/*1. create the subset */
data co;
set sashelp.zipcode;
where statecode = "CO";
run;

/*2. create the xml format data feed file co_zipcode.txt */
DATA _NULL_;
SET co end=eof;
FILE "/user/&sysuserid./co_zipcode.txt";
if _n_ = 1 then put "<markers>";
PUT "<marker lat='" y +(-1)'" lng='" x +(-1)'" html='" zip "-" city
+(-1)'" label='" zip "-" city +(-1)'" />";
if eof then put "</markers>";
RUN;

/*3. send data feed file co_zipcode.txt to SharePoint */
```

```

filename outbox email 'nul';
data _null_;
file outbox
to=( "rm_viz@mycompany.com" , "myname@mycompany.com" )
from=( "myname@mycompany.com" )
attach= "/user/&sysuserid./co_zipcode.txt"
subject="Google Maps data feed file"; run;

/***** end of SAS code *****/

```

The code has three parts. It's quite straightforward. PUT statement is the key to output data to a text file. The extension name of output file can be flexible, it can be .txt, .xml, etc. as long it's consistent with the data file name in the html file.

The +(-1) pointer control expression is used to secure clean (without an extra space) quotation mark closure. This technique becomes necessary if you construct Javascript variable property references with dot notation.

The SharePoint email address [rm\\_viz@mycompany.com](mailto:rm_viz@mycompany.com) needs to be in the **To** line. In this case, SAS will send the web page to the SharePoint Document Library and to the user email box.

The **From** line is equally important. Please use the same email address when you send test emails from Outlook to SharePoint. In this case, it's [myname@mycompany.com](mailto:myname@mycompany.com). SharePoint server needs to recognize this email address within its email directory otherwise it will reject the email.

The **Attach** line includes the path of the SAS file in the SAS server.

Since we choose not to save original email, the **Subject** line is only for reference use. The email content text can be blank.

Please don't expect the file to arrive in the SharePoint Document Library instantly after SAS code runs. Allow 1 minute or so before refreshing the Library.

To use the example code, you will need to upload the example1\_CO.html to the same document library which receives the data feed file co\_zipcode.txt.

Example 1 is using the Google Maps API version 2. The current version is Google Maps version 3. It's recommended to use version 3. But I do like this example with the side bar which use can search the web page to find targeted strings. Then click the link from the side bar and update the map view.

If you just begin to learn Google Maps API, it's helpful to check out Mike Williams's original Google Maps API Tutorial. Here is the original link where I find the prototype of this example.

<http://econym.org.uk/gmap/basic15.htm>.

There are more Google Maps examples on the page which can be adapted. The select box is a good alternative.

- Sidebar with a scroll bar
- Multiple sidebars (a)
- Multiple sidebars (b)
- Select box

Here is the link for Mike Williams entire Google Maps API Tutorial.

<http://econym.org.uk/gmap/oldindex.htm>

## EXAMPLE 2: SHOWING SAS-GENERATED INFO WINDOWS IN GOOGLE MAPS

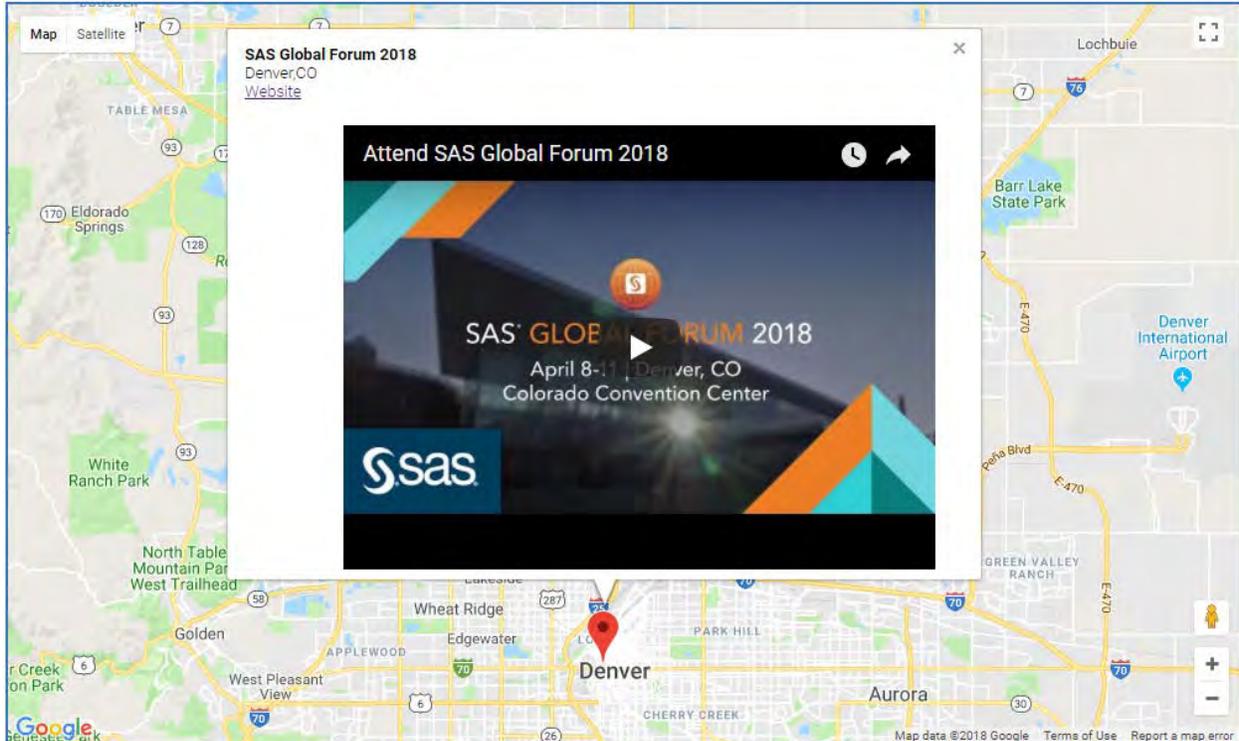
This example will focus on how to make the Info Window more interesting on Google Maps. In addition to text, the Info Window can include hyperlinks, images and even embedded YouTube video. SAS code can create the entire html page along with all the mapping data and related content links. The output html file name is example2\_SASGF.html.

This example is adapted from SAS code posted from SAS user blog “The power of SAS-generated InfoWindows in Google maps” by Leonid Batkhan, please refer to this link for more details.

<https://blogs.sas.com/content/sgf/2014/09/12/the-power-of-sas-generated-google-maps-infowindows/>

The example displays the locations for 2016, 2017 and 2018 SAS Global Forums. You may click on each of the three markers to open its info window. The info window shows the event name, event city, a link for the conference and an embedded Youtube video.





**Figure 4a, 4b, 4c, 4d. Screenshots of Google Maps – example2\_SASGF.html. Displaying locations of 2016-2018 SAS global forums. In the info windows, you can click web link or watch embedded video.**

The SAS code has six steps.

1. Data preparation. Columns are event name, event city, lat (latitude), lng (longitude), web (the web link), utub (Youtube video unique code, the last 11 characters of Youtube video links).
2. Determine map center. Calculate the mean latitude and longitude from data points.
3. Initialize map.
4. Add multiple markers.
5. Wrap up HTML file.
6. Send html file example2\_SASGF.html to SharePoint.

The key here is to understand how the code create the info window contents.

The +(-1) pointer control expression is used to secure clean (without an extra space) quotation mark closure. This technique becomes necessary if you construct Javascript variable property references with dot notation.

To read more about Google Maps Info Windows, please refer to <https://developers.google.com/maps/documentation/javascript/infowindows>.

Here is the SAS Code for example 2.

**SAS Code for example 2:**

```
/* HTML output file location */
filename fout "/user/&sysuserid./example2_SASGF.html" ;

/* 1. data preparation */
```

```

data loc_input;
  input event $21. city $14. lat lng web $55. utub $12.;
  datalines;
SAS Global Forum 2018 Denver,CO    39.742531 -104.996273
https://www.sas.com/en_us/events/sas-global-forum.html SQecEp9e-Hw
SAS Global Forum 2017 Orlando,FL   28.367949 -81.560350
http://support.sas.com/resources/papers/proceedings17/ 7Mgh0SARw5Q
SAS Global Forum 2016 Las Vegas,NV 36.121180 -115.169636
http://support.sas.com/resources/papers/proceedings16/ wZmy6SioLz0
;

/* 2. determine map center */
proc means data=loc_input noprint;
  var lat lng;
  output out=centerdata(keep=clat clng) mean(lat)=clat
mean(lng)=clng; run;

/* 3. initialize map */
data _null_;
  file fout;
  set centerdata;
  put
  '<html><head>' /
  '<title>Google Maps Example </title>' /
  '<script type="text/javascript"
src="https://maps.googleapis.com/maps/api/js?v=3.exp&sensor=false"></s
cript>' /
  '<script type="text/javascript">' /
  'function initialize() {' /
      'var cpoint = new google.maps.LatLng(' clat ',' clng ');' /
      'var mapOptions = {' /
        'zoom: 5,' /
        'center: cpoint,' /
        'mapTypeId: google.maps.MapTypeId.ROADMAP' /
      '};' /
      'var map = new
google.maps.Map(document.getElementById("map-canvas"), mapOptions);' /
      'var infowindow = new google.maps.InfoWindow();'
  ; run;

/* 4. add multiple markers */
data _null_;
  file fout mod;
  set loc_input;
  length i $3;
  i = strip(_n_);
  put
  'var info' i '=' '<div style="width:560px;height:420px;"><div
style="float:left;margin-top:5px;margin-right:5px;"><b>'

```

```

        event +(-1) '</b><br>' city +(-1) '<br><a href="' web +(-1)
'" target="_blank">Website</a></div>'
        '<div style="float:right;margin-top:20px"><iframe
width="480" height="360" src="//www.youtube.com/embed/'
        utub +(-1) '" frameborder="0" allowfullscreen></iframe>
</div> </div>''; ' /

        'var point' i '= new google.maps.LatLng(' lat +(-1)', ' lng +(-
1)');' /

        'var marker' i '= new google.maps.Marker({' /
        'position: point' i ', ' /
        'map: map,' /
        'title: "Click here for details"' /
        '});' /

        'google.maps.event.addListener(marker' i ', "click", function()
{' /
        'infowindow.setContent(info' i ');' /
        'infowindow.open(map,marker' i ');' /
        '});'
;
run;

/* 5. wrap up HTML file */
data _null_;
file fout mod;
put
'{'
'google.maps.event.addDomListener(window, "load", initialize);' /
'</script>' /
'</head><body>' /
'<p>Example 2 for SAS Paper 2604-2018: Data Visualization from
SAS® to Google Maps on Microsoft SharePoint. Move around, zoom in/out,
click on markers to display info. In info windows, click on links,
watch video.</p>' /
'<div id="map-canvas" style="width:1000px; height:600px; border:
2px solid #3872ac;"></div>' /
'</body></html>'
;
run;

/* 6. send html file example2_SASGF.html to SharePoint */
filename outbox email 'nul';
data _null_;
file outbox
to=("rm_viz@mycompany.com", "myname@mycompany.com")
from=("myname@mycompany.com")
attach= "/user/&sysuserid./example2_SASGF.html"
subject="Google Maps html file"; run;

/***** end of SAS code *****/

```

## CONCLUSION

The whole process of data mapping from SAS® to Microsoft SharePoint can serve as a practical low-cost Business Intelligence toolbox. For SAS and SharePoint, each only requires standard license. It can build up interesting infographics with Google Maps on SharePoint, without the need to buy additional Analytics products, which can be expensive.

One thing to note is that data mapping with Google Maps requires multidimensional skills (SAS, SharePoint, HTML, JavaScript) from a business analyst. With so many free online learning resources available now, the analyst should feel encouraged and empowered to learn the necessary skill sets to create a variety of info graphics and data visualization.

It's highly recommended to try this new approach with the example codes. Google Maps API is a very powerful tool. This paper only touches the tip of the iceberg.

Official Google Maps JavaScript API tutorial link is:

<https://developers.google.com/maps/documentation/javascript/>

You can start with the most basic examples and move on to more advanced examples. Here is the list.

- Adding a Google Map with a Marker to Your Website
- Marker Clustering
- Importing Data into Maps
- Visualizing Data: Mapping Earthquakes
- Combining and Visualizing Multiple Data Sources
- Collaborative Real time Mapping with Firebase
- Geolocation: Displaying User or Device Position on Maps
- Using MySQL and PHP with Google Maps
- From Info Windows to a Database: Saving User-Added Form Data

Here are a few SAS code examples for you to explore the world of Google Maps.

“Drawing overlays on SAS-generated Google maps” is a SAS blog post by Leonid Batkhan. It demonstrates how to draw a closed geometric shape, in particular, a polygon, on a Google map.

Web link: <https://blogs.sas.com/content/sgf/2014/08/29/drawing-overlays-on-sas-generated-google-maps/>

“SAS to the rescue: claiming your location on Google map by address only (without knowing latitude and longitude)” is also a SAS blog post by Leonid Batkhan. It shows several gas stations on Google Maps with just the street addresses as data input.

The post introduces an example of geocoding. SAS geocoding procedure PROC GEOCODE produces geographic coordinate information for the following entities: street addresses, cities, U.S. ZIP codes and ZIP+4 extension codes and foreign postal codes.

Web link: <https://blogs.sas.com/content/sgf/2015/11/03/sas-to-the-rescue-claiming-your-location-on-google-map-by-address-only-without-knowing-latitude-and-longitude/>

Another geocoding SAS code example can be found here:

[http://www.sascommunity.org/wiki/Geocoding\\_using\\_SAS\\_and\\_Google\\_Maps](http://www.sascommunity.org/wiki/Geocoding_using_SAS_and_Google_Maps)

As a data mapping alternative to Google Maps, you may also take a look at the GeoChart in Google Charts. Here is the link: <https://developers.google.com/chart/interactive/docs/gallery/geochart>

Good luck with data mapping! Hope you find this paper interesting.

## REFERENCES

1. Herderson, D. R. and S. Alexandre 2009. "Integrating SAS® Business Intelligence with Microsoft." *Proceedings of SAS Global Forum 2009*.
2. Xiaogang Tang 2016. "Releasing the Power of SAS® into Microsoft SharePoint." *Proceedings of SAS Global Forum 2016*.
3. Xiaogang Tang 2017. "Data Visualization from SAS® to Microsoft SharePoint." *Proceedings of SAS Global Forum 2017*.
4. Leonid Batkhan, "Spice up SAS output with live Google maps". Accessed March 1, 2018. <https://blogs.sas.com/content/sqf/2014/06/06/spice-up-sas-output-with-live-google-maps/>
5. Leonid Batkhan, "The power of SAS-generated InfoWindows in Google maps". Accessed March 1, 2018. <https://blogs.sas.com/content/sqf/2014/09/12/the-power-of-sas-generated-google-maps-infowindows/>
6. Leonid Batkhan, "SAS to the rescue: claiming your location on Google map by address only (without knowing latitude and longitude)". Accessed March 1, 2018. <https://blogs.sas.com/content/sqf/2015/11/03/sas-to-the-rescue-claiming-your-location-on-google-map-by-address-only-without-knowing-latitude-and-longitude/>
7. Leonid Batkhan, "Drawing overlays on SAS-generated Google maps". Accessed March 1, 2018. <https://blogs.sas.com/content/sqf/2014/08/29/drawing-overlays-on-sas-generated-google-maps/>
8. Mike Zdeb, "Driving Distances and Drive Times using SAS and Google Maps". Accessed March 1, 2018. [http://www.sascommunity.org/wiki/Driving\\_Distances\\_and\\_Drive\\_Times\\_using\\_SAS\\_and\\_Google\\_Maps](http://www.sascommunity.org/wiki/Driving_Distances_and_Drive_Times_using_SAS_and_Google_Maps)
9. Mike Williams, "Google Maps API Tutorial". Accessed March 1, 2018. <http://eonym.org.uk/gmap/basic15.htm>

## RECOMMENDED READING

- *Base SAS® Procedures Guide*
- *SAS Graphics Samples Output Gallery* <https://support.sas.com/sassamples/graphgallery/>
- *PROC GEOCODE tutorial* <http://support.sas.com/documentation/cdl/en/graphref/63022/HTML/default/viewer.htm#overview-geocode.htm>
- *PROC GMAP Graphics Samples Output Gallery* [https://support.sas.com/sassamples/graphgallery/PROC\\_GMAP.html](https://support.sas.com/sassamples/graphgallery/PROC_GMAP.html)
- *SAS ODS (Output Delivery System)* <http://support.sas.com/rnd/base/ods/index.html>
- *Google Maps JavaScript API documentation* <https://developers.google.com/maps/documentation/javascript/tutorial>
- *Google Charts Visualization: GeoChart.* <https://developers.google.com/chart/interactive/docs/gallery/geochart>

## CONTACT INFORMATION

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

Xiaogang (Isaac) Tang  
Wyndham Worldwide  
xiaogang.tang@rci.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.