

Releasing the Power of SAS® into Microsoft SharePoint

Xiaogang (Isaac) Tang, Wyndham Destination Network

ABSTRACT

SharePoint is a web application framework and platform developed by Microsoft, mostly used for content and document management by midsize businesses and large departments. Linking SAS with SharePoint will combine the power of these two into one. This paper shows users how to send PDF reports and files in other formats (such as Excel files, HTML files, JPEG files, zipped files, etc.) from SAS to a SharePoint Document Library. The paper demonstrates how to configure SharePoint Document Library settings to receive files from SAS. A couple of SAS code examples are included to show how to send files from SAS to SharePoint. The paper also introduces a framework for creating data visualization on SharePoint by feeding SAS data into HTML pages on SharePoint. An example on how to create an infographics SharePoint page with data from SAS is also provided.

INTRODUCTION

SharePoint is a popular web application framework and platform developed by Microsoft, mostly used for document and content management by large departments and midsize businesses.

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. These approaches include: 1. Writing SAS Content to SharePoint Document Libraries directly using SAS Output Delivery System (ODS). 2. Using SAS BI Web Parts to Include Real-Time Content on a SharePoint Page; 3. SAS Stored Process Web Part; 4. SAS BI Dashboard Web Part.

These approaches have certain limitations to be applied in a corporate environment.

Due to IT firewalls between SharePoint system and SAS system, using SAS ODS filename statement to directly write SAS content to SharePoint often is not viable. It's both complex and difficult to ask IT to find a solution to allow it.

Since Microsoft SharePoint is a completely different product from SAS, many corporate SharePoint solutions simply do not have SAS BI Web Parts, SAS Stored Process Web Part or SAS BI Dashboard Web Part installed. So using SAS Web Parts to connect SAS with SharePoint may also not be feasible.

This paper presents a simple and friendly solution to send SAS contents to SharePoint via an existing email system. Sending PDF reports and Excel files from a SAS server to user email addresses through a corporate email system is a standard practice. Interestingly, SharePoint Document Library can easily be configured to assign a user-defined corporate email address. This allows users to send emails with content from Microsoft Outlook to SharePoint Document Library.

By combining these two techniques, users can write a SAS program to send emails with SAS contents to a specific SharePoint Document Library. For example, users can send a PDF report directly from SAS to a report document Library.

More interestingly, now users can create data visualization web pages (for example, a Google Chart page) and save them in a SharePoint document library. The new approach can send JSON/CSV format data files from SAS to SharePoint, serving as source data to refresh the data visualization web pages.

This whole process can serve as a new Business Intelligence framework. SAS is highly proficient for data processing and analysis. SharePoint Intranet Web system is convenient to share content and manage user permissions. This approach combines the strength of both SAS and SharePoint into one. Now we can release the power of SAS into Microsoft SharePoint.

HOW TO 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 is the email address.

E-Mail Attachments – you have three choices, please select one. SharePoint will only allow email sent to one level subfolder. You cannot send email to a folder within a folder. “Overwrite files with the same name?” Make sure to check “Yes” if you want to overwrite.

- Save all attachments in root folder
- Save all attachments in folders grouped by e-mail subject
- Save all attachments in folders grouped by e-mail sender

E-Mail Message – Choose “No” to not save original email, if you only need to transfer the attachments from SAS to SharePoint. Typically we choose “No” to keep the SharePoint folder clean.

E-Mail Security – If you choose “Accept e-mail messages based on document library permissions”, you will need to make sure that the “From” email address in SAS code part also has write permission in this SharePoint Library.

4. Don’t forget to click “OK” after set-up.

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

Incoming E-Mail
Specify whether to allow items to be added to this document library through e-mail. Users can send e-mail messages directly to the document library by using the e-mail address you specify.

Allow this document library to receive e-mail?
☒ Yes ☐ No

E-mail address:
@.com

E-Mail Attachments
Specify whether to group attachments in folders, and whether to overwrite existing files with the same name as incoming files.

Group attachments in folders?
☒ Save all attachments in root folder
☐ Save all attachments in folders grouped by e-mail subject
☐ Save all attachments in folders grouped by e-mail sender

Overwrite files with the same name?
☒ Yes ☐ No

E-Mail Message
Specify whether to save the original .eml file for an incoming e-mail message.

Save original e-mail?
☐ Yes ☒ No

E-Mail Meeting Invitations
Specify whether to save e-mailed meeting invitations in this document library.

Save meeting invitations?
☐ Yes ☒ No

E-Mail Security
Use document library security for e-mail to ensure that only users who can write to the document library can send e-mail to the document library.

E-mail security policy:
☒ Accept e-mail messages based on document library permissions
☐ Accept e-mail messages from any sender

Caution: If you allow e-mail from any sender, you are bypassing the security settings for the document library. This means that anyone could send an e-mail to the document library's address and their item would be added. With this option turned on, you are opening your document library to spam or other unwanted e-mail messages.

OK Cancel

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

If you want to test how to save the attachments into folders within the root folder by e-mail subject or by e-mail sender, it's better to do some tests from Outlook, before coding in SAS.

You can send a variety of file formats (PDF, Excel, CSV, DBF, etc.) as attachments to SharePoint. Please be aware that SharePoint does have a size limit (typically 5MB) for incoming email attachments. Please check with your SharePoint admin to find out the actual limit or do some tests to try out.

EXAMPLE SAS CODES TO SEND FILES TO SHAREPOINT

Now you have successfully sent email from your Outlook to the SharePoint Document Library via the email address rm_viz@mycompany.com. Let's move on to the SAS coding part.

CODE EXAMPLE 1 – HOW TO SEND A SINGLE FILE TO SHAREPOINT

This example uses SAS table **sashelp.cars** as data source.

Sashelp is a native SAS library. SAS provides over 200 data sets in the sashelp library. These data sets are available for you to use for examples and for testing code.

Firstly, this code part will export the SAS table to a CSV file:

```
proc export data=sashelp.cars
  outfile="/user/&sysuserid./cars.csv"
  dbms=csv replace; run;
```

A CSV is a comma separated values text file, which allows data to be saved in a table structured format. Before SAS can send the file to SharePoint, SAS will need to create the file and save it on the SAS server.

To know more about PROC EXPORT Statement, you may read SAS.com online documentation.

In this example, output file "cars.csv" is saved in the root SAS folder "/user/&sysuserid./" of the SAS user who runs the code. File names and path on SAS server are case-sensitive. "cars.csv" is different from "CARS.csv" or "cars.CSV". Please keep this in mind when designating the file name in the file path.

Secondly, this code below will send the cars.csv as email attachment to SharePoint Document Library:

```
filename outbox email 'nul';

data _null_;
  file outbox
    to=( "rm_viz@mycompany.com" )
    from=( "myname@mycompany.com" )
    attach= "/user/&sysuserid./cars.csv"
    subject="SAS table sashelp.cars";

run;
```

The SharePoint email address rm_viz@mycompany.com needs to be in the **To** line.

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. 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.

If you want to understand more about this SAS coding, please read SAS.com online documentation: FILENAME Statement, EMAIL (SMTP) Access Method.

CODE EXAMPLE 2 – HOW TO SEND MULTIPLE FILES TO SHAREPOINT

The first code example demonstrates sending a single file from SAS to SharePoint. This code is to show you how to send multiple files in one email to SharePoint.

This example uses SAS table **sashelp.cars** and **sashelp.class** as data sources.

Here is the code:

```
proc export data=sashelp.cars outfile="/user/&sysuserid./cars.csv"
  dbms=csv replace; run;

proc export data=sashelp.class outfile="/user/&sysuserid./class.csv"
  dbms=csv replace; run;

filename outbox email 'nul';

data _null_;
  file outbox
    to= ("rm_viz@mycompany.com")
    from= ("myname@mycompany.com")
    attach= ("/user/&sysuserid./cars.csv" "/user/&sysuserid./class.csv")
    subject="SAS tables sashelp.cars and sashelp.class";

run;
```

The code difference here is that you can include multiple files in the **Attach** line, separated by space.

CODE EXAMPLE 3 – HOW TO SEND A ZIPPED FILE TO SHAREPOINT

The first code example can be easily modified to send a variety of file formats from SAS to SharePoint.

Sometimes a file can be too big to send to SharePoint. Or you have too many files to send. You can create a zipped file in SAS and then send the zipped file to SharePoint.

There could be a file corruption issue when SAS directly sends an Excel file (.xlsx) from SAS to SharePoint. The Excel file is received in the SharePoint Library, but becomes corrupted and un-readable.

Sending a zipped file from SAS to SharePoint could solve all these issues.

This example uses SAS table **sashelp.attrlookup** and **sashelp.cars** as data sources. Sashelp.attrlookup table has 40,136 rows. We want to send the first table as Excel file and the second table as CSV file to SharePoint.

Firstly, this code part will export the SAS tables to a CSV file and an Excel 2007 file (.xlsx) respectively:

```
proc export data=sashelp.cars
  outfile="/user/&sysuserid./cars.csv"
  dbms=csv replace;
run;

proc export data=sashelp.attrlookup
  outfile="/user/&sysuserid./attrlookup.xlsx"
  dbms=xlsx replace;
run;
```

The output file ATTRLOOKUP.xlsx has a file size of 2.3MB, which is relatively big.

Secondly, this code will create a zipped file named **attrlookup_cars.zip** that contains both **cars.csv** and **attrlookup.xlsx**. It is saved in the same SAS folder.

```
ods package(newzip) open;
ods package(newzip) add file="/user/&sysuserid./cars.csv";
ods package(newzip) add file="/user/&sysuserid./attrlookup.xlsx";
ods package(newzip) publish archive
  properties(
    archive_name="attrlookup_cars.zip"
    archive_path="/user/&sysuserid./"
  );
ods package(newzip) close;
```

The zipped file attrlookup_cars.zip only has a file size of 1.3 MB.

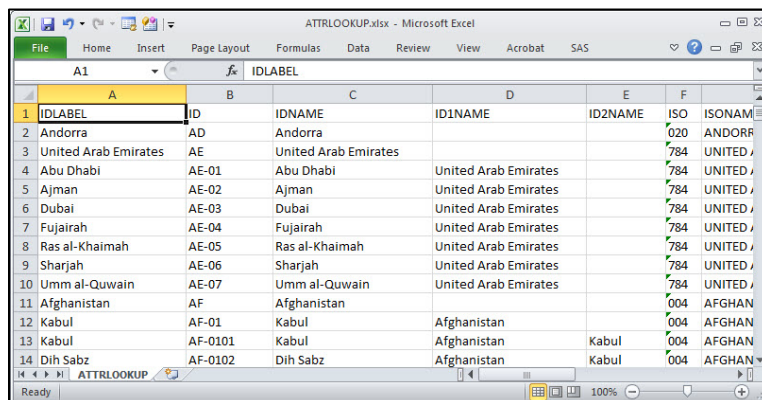
To know more about the ODS PACKAGE Statement, you may read SAS.com online documentation.

Finally, the code below will send the cars.csv as email attachment to SharePoint Document Library:

```
filename outbox email 'nul';
data _null_;
  file outbox
    to= ("rm_viz@mycompany.com")
    from= ("myname@mycompany.com")
    attach= ("/user/&sysuserid./attrlookup_cars.zip")
    subject="SAS tables sashelp.attrlookup and sas.help.cars";
run;
```

Allow 1 minute or so before you refresh the Library.

Now the zipped file successfully arrives in the Library. Download the zipped file and unzip it. As Figure 2 shows, the Excel file is not corrupted and good for use.



	A	B	C	D	E	F
1	IDLABEL	ID	IDNAME	ID1NAME	ID2NAME	ISONAM
2	Andorra	AD	Andorra		020	ANDORR
3	United Arab Emirates	AE	United Arab Emirates		784	UNITED /
4	Abu Dhabi	AE-01	Abu Dhabi	United Arab Emirates	784	UNITED /
5	Ajman	AE-02	Ajman	United Arab Emirates	784	UNITED /
6	Dubai	AE-03	Dubai	United Arab Emirates	784	UNITED /
7	Fujairah	AE-04	Fujairah	United Arab Emirates	784	UNITED /
8	Ras al-Khaimah	AE-05	Ras al-Khaimah	United Arab Emirates	784	UNITED /
9	Sharjah	AE-06	Sharjah	United Arab Emirates	784	UNITED /
10	Umm al-Quwain	AE-07	Umm al-Quwain	United Arab Emirates	784	UNITED /
11	Afghanistan	AF	Afghanistan		004	AFGHAN
12	Kabul	AF-01	Kabul	Afghanistan	004	AFGHAN
13	Kabul	AF-0101	Kabul	Afghanistan	Kabul 004	AFGHAN
14	Dih Sabz	AF-0102	Dih Sabz	Afghanistan	Kabul 004	AFGHAN

Figure 2. Screenshot for ATTRLOOKUP.xlsx

Tip: for large tables, it's a good practice to export to DBF file and create a zipped file. This will greatly reduce the output file size. And then send the zipped file to SharePoint.

DBF stands for DataBase File which is a binary file format typically used by database software. DBF files can also be opened by Microsoft Excel and Microsoft Access. DBF files are commonly used as data source for Excel PivotTables.

HOW TO CREATE DATA VISUALIZATION ON SHAREPOINT WITH SAS DATA

Above SAS code examples demonstrate sending files from SAS to SharePoint can fulfill common business reporting needs, such as sending PDF reports, Excel files, DBF files, CSV files, etc.

This section will cover a more advanced topic – creating data visualization on SharePoint.

Good and timely data visualization can provide great insights for business decisions, such as a daily refreshed KPI dashboard. There are many open source JavaScript Libraries available now. They can be implemented to create beautiful and meaningful infographics.

Combining the visualization tools with our new approach, we can create data visualization web pages on SharePoint, using the data files sent by SAS. Here is the flow of this framework:

1. Run SAS code to process business database and create summary data.
2. Choose an applicable JavaScript Library and data visualization demo example(s), upload demo HTML page and demo data file to SharePoint Document Library. Do tests to make sure that the demo page looks fine.
3. Identify data file format from demo HTML page, run SAS code to format the summary data and export to data file with proper formatting which can feed into the HTML page. Make change to the demo page to read from the new data file.
4. Once the new visualization page can read the new data file, schedule the SAS code to run the whole process: data processing, data formatting, data exporting and sending data feeds to SharePoint.

CODE EXAMPLE 4 – HOW TO CREATE A DATA VISUALIZATION ON SHAREPOINT

This example uses SAS table **sashelp.citiday** as data source. The sashelp.citiday data set provides Citibase financial daily indicators: from Jan.1988 to Feb. 1992. We are going to visualize the daily NY DOW JONES COMPOSITE Index data for year 1991, the SNYDJCM variable in this SAS table.

The JavaScript Library in this example is D3.js. D3.js is a very popular JavaScript library for manipulating documents based on data. D3 helps you bring data to life using HTML, SVG, and CSS. Please see reference for more about D3.js.

The original demo used in this example is d3.js Alternative Calendar View created by Kathy Zhu. Please see reference for Kathy's web page.

This code part will process the SAS table and export to a CSV file (nydj.csv):

```
proc sql; create table nydj1991 as
select date format=YYMMDD10., round(SNYDJCM,.1) as SNYDJCM
from sashelp.citiday
where year(date) = 1991 and SNYDJCM is not null; quit;
proc export data=nydj1991 outfile="/user/&sysuserid./nydj.csv"
dbms=csv replace; run;
```

This code part will send the csv file from SAS to SharePoint:

```
filename outbox email 'nul';
data _null_;
file outbox
to=("rm_viz@mycompany.com")
```

```

from= ("myname@mycompany.com")
attach= ("/user/&sysuserid./nydj.csv")
subject=" Subset of SAS table sashelp.citiday"; run;

```

The demo data visualization HTML page needs to be modified and use the right file name and column names. Here is a key part in the html page source code (This is NOT SAS code).

```

d3.csv("nydj.csv", function(error, csv) {
  var data = d3.nest()
    .key(function(d) { return d.DATE; })
    .rollup(function(d) { return d[0].SNYDJCM; })
    .map(csv);

```

Typically, customizations can be made in the html page source code. You may put in a new color scheme for your data.

Here is the screenshot of the D3.js data visualization (Figure 3) with source data from **sashelp.citiday**. From red to blue, daily index number ranges from lowest (881) to highest (1157).

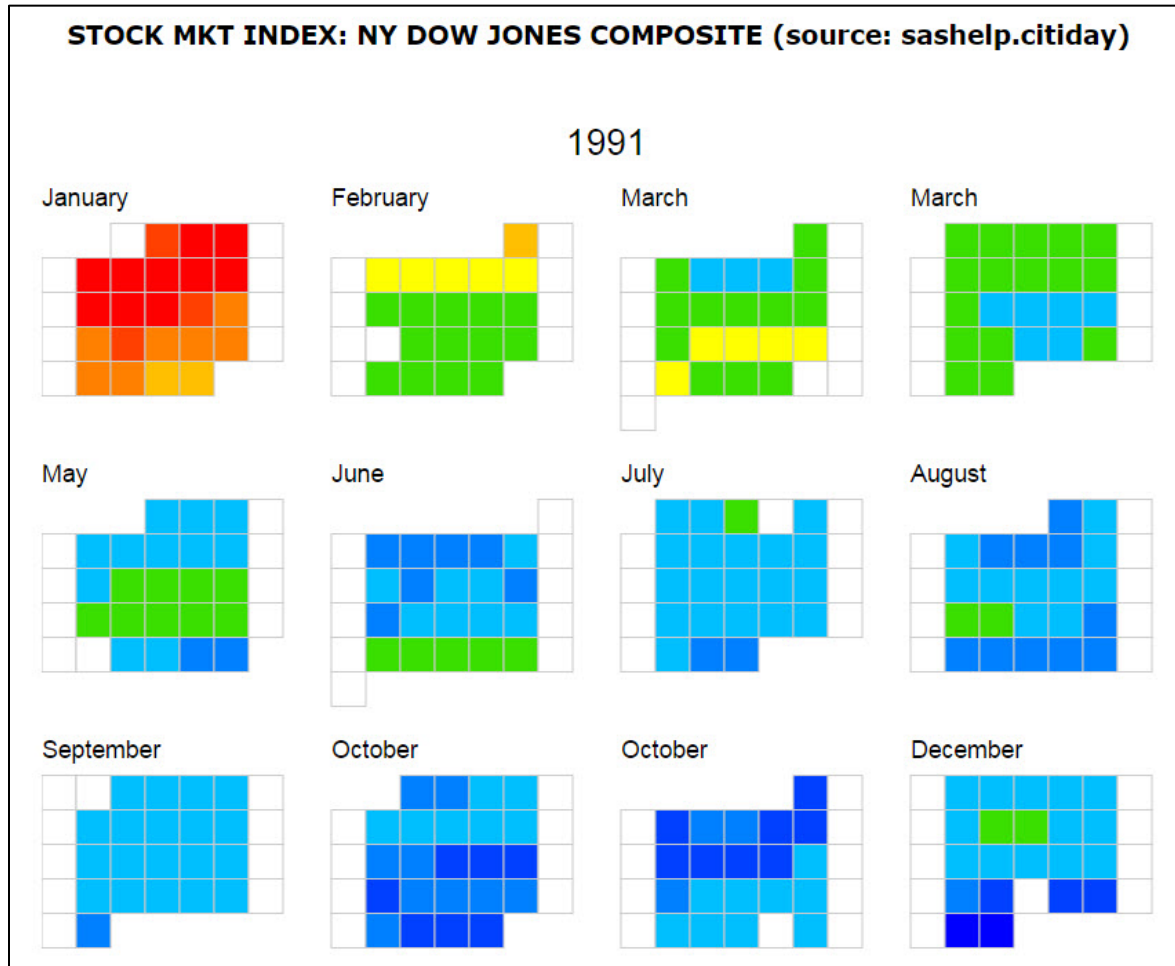


Figure 3. Screenshot of data visualization web page on SharePoint (data source: sashelp.citiday). From red to blue, daily index number ranges from lowest (881) to highest (1157).

CONCLUSION

A practical approach is introduced in this paper to transfer files from SAS to SharePoint through email system. Several examples are used to show how to send a variety of file formats, or a zipped file from SAS to SharePoint. A framework for creating data visualization web pages on SharePoint is also introduced. In this framework, data can be processed in SAS and sent to SharePoint as the source data for visualization pages.

Both SharePoint and SAS are widely popular in Enterprise environment. The approach in this paper is straight forward and easy to be applied. It can release the power of SAS programming into SharePoint.

The new approach provides huge benefits for timely business reporting. It also provides high potentials to create meaningful data visualization for business intelligence.

Yet just knowing the new approach is not enough. The key is implementation. Every company is different and there will be no universal solution for all.

To create a successful implementation will require multidimensional skills (SAS, SharePoint, HTML/CSS, JavaScript, etc.) from a business analyst. To create a meaningful visualization, the analyst will also need guidance and user feedback from business stake holders.

You are highly encouraged to try out this new approach with the example codes. At first it may need some dedicated efforts to fit it into your current framework, and then there will be a great journey ahead.

REFERENCES

Herderson, D. R. and S. Alexandre 2009. "Integrating SAS® Business Intelligence with Microsoft." *Proceedings of SAS Global Forum 2009*.

Kathy Zhu, "d3.js Alternative Calendar View". Accessed February 24, 2016.
<http://bl.ocks.org/KathyZ/c2d4694c953419e0509b>.

RECOMMENDED READING

- *Base SAS® Procedures Guide*
- *Introduction to D3.js JavaScript Library* <https://d3js.org/>
- *Html Color Codes* <http://html-color-codes.info/>

CONTACT INFORMATION

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

Xiaogang (Isaac) Tang
Wyndham Destination Network
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.