

What's your data flow in SAS® LSAF (How to code in LSAF)

Shuang Fu, SAS R&D, Beijing, China

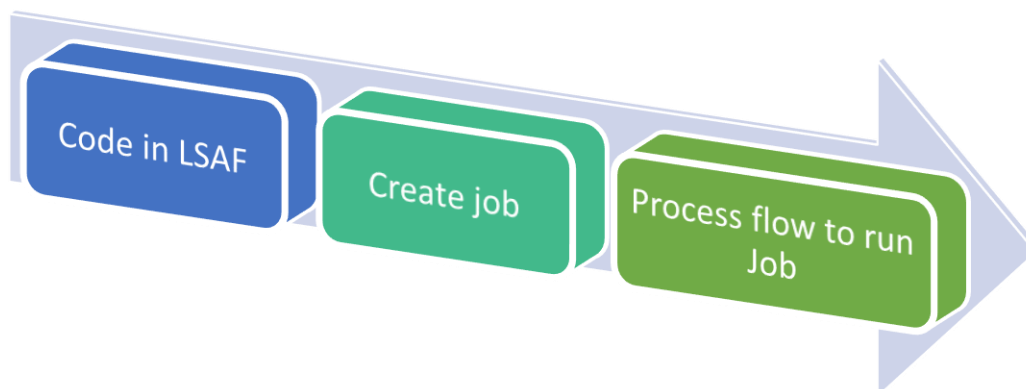
ABSTRACT

You may or may not have heard of LSAF, its full name is SAS® Life Science Analytics Framework. It is a powerful solution for pharmaceutical industry. The integrated system empowers you to better manage, analyze, report, and review clinical research information. This solution keeps your data secure and at the same time allows you to share data and information with all coworkers in your organization no matter where they are located. If you were a SAS programmer, you can code, submit and share your job in it. I have heard complaints from SAS programmers about how hard it is to switch back and forth between your PC and laptop. But using this product, all you needs, is a supported browser. There are many features in this product. In this paper, I only introduce the data flow in LSAF, how to code in it, and how to work effectively in it. Other advantages of LSAF, need your own ventures.

INTRODUCTION

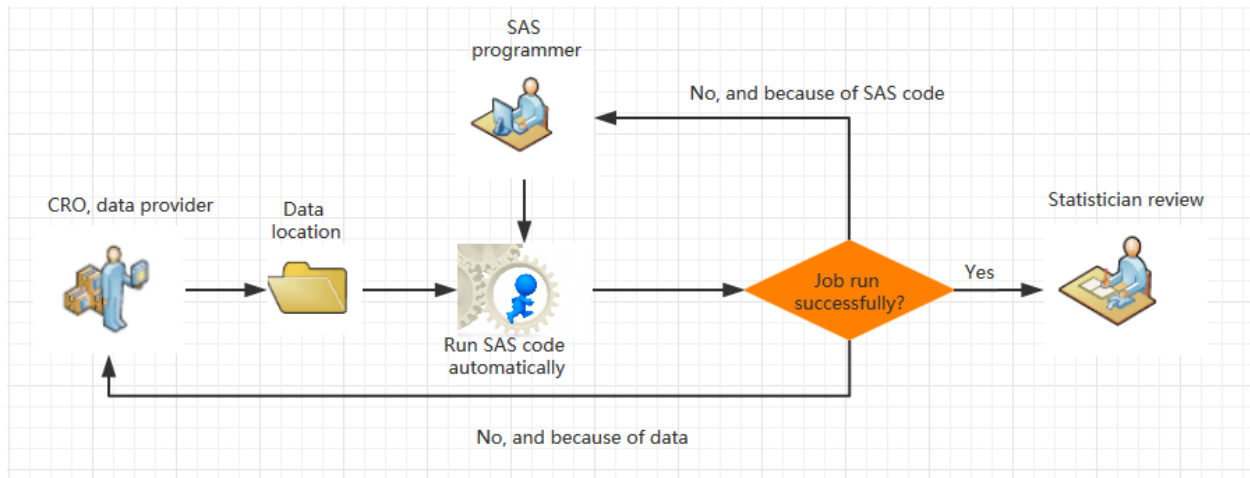
Nowadays, many companies operate business globally. The coworkers work in different places, but need to share data and information timely. SAS LSAF is such solution that makes all coworkers work in “same” place. All coworkers can view the data flow, manage and analyze clinical research content at the same time in the same level as long as they have privileges. Meanwhile, the SAS programmers develop and submit their codes to SAS server that is located on the SAS Life Science Analytics Framework server. So the programmers don't need to install SAS® foundation. They can run their code on any computer as long as it has a supported browser. Far more than just a web-based SAS foundation, SAS Life Science Analytics Framework provides the ability to meet industry expectations for managing research content in accordance with government regulations. The government regulations include 21 CFR Part 11, good industry practices (GxP), and sound business practices. For security reasons, all users in LSAF have different privileges and access permissions. For simplicity, I just assume that all users have correct privileges and access permissions in this paper.

I will introduce the data flow in SAS® LSAF from 3 parts:



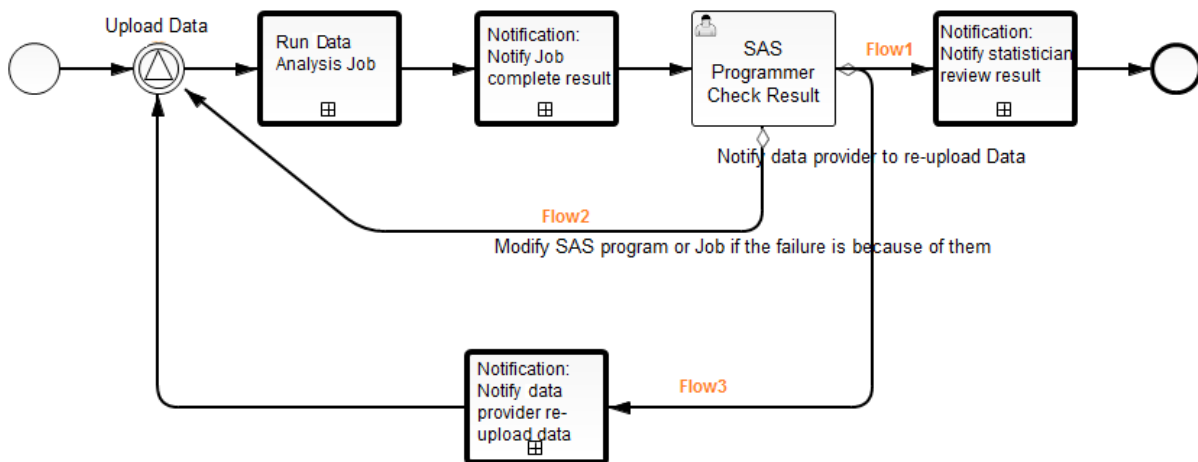
I'm going to use a fictitious scenario to introduce working in LSAF as a programmer. From now on, I'm a programmer in a pharmaceutical company working in Beijing, I'm requested to do survival analysis on data which is provided by CRO. There is a statistician in my company working in Shanghai to review the result. I want to follow a work flow as follows:

What's your data flow in SAS® LSAF(How to code in LSAF), continued



BUILD AN EFFECTIVE PROCESS FLOW

In SAS® LSAF, the process flow feature can help to execute the work flow. Base on the business request above, I build below process definition file using Camunda Modeler. About how to define process flow definition file, please refer to the paper [How to customize your business process more effectively by BPMN with SAS LSAF](#)



Display 1: Process flow for analyzing data

The working logical of this process flow is:

1. Once the data was uploaded to the location I specified in my code, the SAS program will be run automatically by the job;
2. After the job ran completely, the system will send me a message/email automatically to tell me the program run result.
3. If the run failed, I will check the reason. If the failure was because of the provided research data, I will select the completion data to go flow3, and the system will send a message/email to data provider to notify uploading correct data, then the process flow will go back to the beginning to wait the new data being uploaded.
4. If the run failed because of the SAS program or Job, I will select the completion data to go flow2, then the process flow will go back to the beginning, after the SAS program or Job is updated, the job will run again automatically.
5. If it ran successfully, I select the completion data to go flow1, the system will send a message/email to the statistician to review the result.

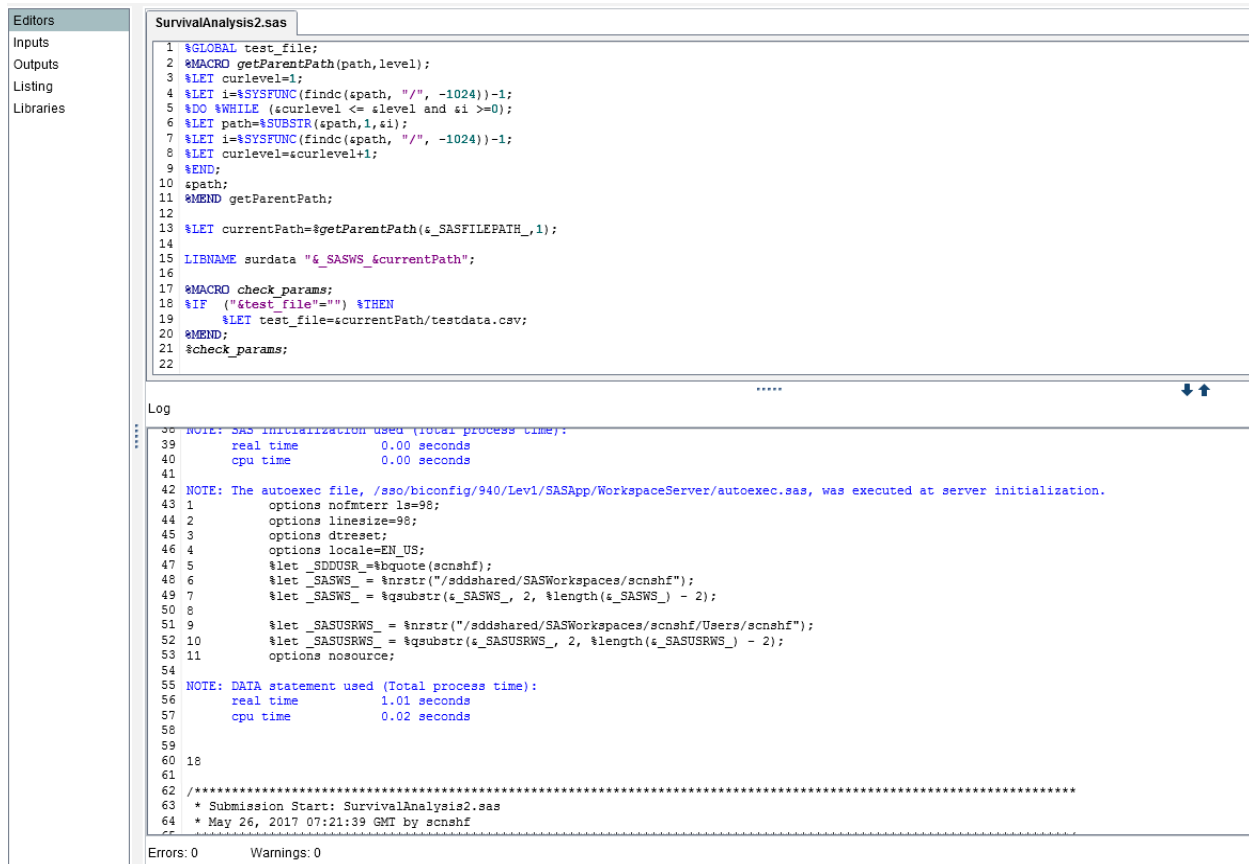
What's your data flow in SAS® LSAF(How to code in LSAF), continued

The benefit to utilize the process flow to run SAS program is that once I completed my SAS program, I don't need to run my code manually after CRO provides research data. As long as the CRO data provider signs in LSAF and uploads the research data to the location I specified, the SAS program will run automatically through job. I don't need to wait the run result of my SAS program if it takes a long time to run, the system will send me the result automatically. The cycle can be run again and again, until I get expected result. And the process flow can be re-used for any data analysis task.

CODE IN LSAF

Now I start to develop SAS program in LSAF. Open SAS session in my workspace, workspace is a private place to LSAF users, my workspace is only available to me. My workspace is the location in which I manipulate the objects that I need to perform my job. When the files are ready to be used by other users, I check the files into the repository. The repository is a collection of objects that are available to all SAS LSAF users. Both workspace and repository are located in a secure server.

SAS session editors in LSAF is a color-coded editor for editing new or existing SAS programs.



The screenshot displays the SAS LSAF editor interface. On the left, a sidebar contains navigation options: Editors, Inputs, Outputs, Listing, and Libraries. The main editor window shows a SAS program named 'SurvivalAnalysis2.sas' with the following code:

```
1 %GLOBAL test_file;
2 %MACRO getParentPath(path,level);
3 %LET curlevel=1;
4 %LET i=%SYSFUNC(findc(spath, "/", -1024))-1;
5 %DO %WHILE (scurlevel <= slevel and si >=0);
6 %LET path=%SUBSTR(spath,1,si);
7 %LET i=%SYSFUNC(findc(spath, "/", -1024))-1;
8 %LET curlevel=scurlevel+1;
9 %END;
10 spath;
11 %MEND getParentPath;
12
13 %LET currentPath=%getParentPath(%_SASFILEPATH_,1);
14
15 LIBNAME surdata "%_SASWS_currentPath";
16
17 %MACRO check_params;
18 %IF ("%test_file"="") %THEN
19     %LET test_file=%currentPath/testdata.csv;
20 %MEND;
21 %check_params;
22
```

Below the code editor is a 'Log' window showing the execution output:

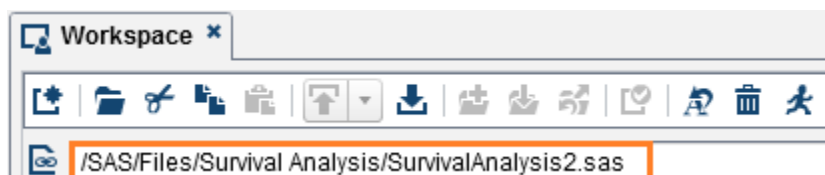
```
30 NOTE: SAS initialization used (total process time):
39   real time      0.00 seconds
40   cpu time       0.00 seconds
41
42 NOTE: The autoexec file, /sso/biconfig/940/Lev1/SASApp/WorkspaceServer/autoexec.sas, was executed at server initialization.
43 1 options nofmterr ls=98;
44 2 options linesize=98;
45 3 options dstore;
46 4 options locale=EN_US;
47 5 %let _SDDUSR_=%bquote(%scnhf);
48 6 %let _SASWS_=%nrstr("/addshared/SASWorkspaces/%scnhf");
49 7 %let _SASWS_=%qsubstr(%_SASWS_, 2, %length(%_SASWS_) - 2);
50 8
51 9 %let _SASUSRWS_=%nrstr("/addshared/SASWorkspaces/%scnhf/Users/%scnhf");
52 10 %let _SASUSRWS_=%qsubstr(%_SASUSRWS_, 2, %length(%_SASUSRWS_) - 2);
53 11 options nosource;
54
55 NOTE: DATA statement used (Total process time):
56   real time      1.01 seconds
57   cpu time       0.02 seconds
58
59 18
60 18
61
62 /*****
63 * Submission Start: SurvivalAnalysis2.sas
64 * May 26, 2017 07:21:39 GMT by scnhf
65 *****/
```

At the bottom of the log window, it indicates 'Errors: 0' and 'Warnings: 0'.

Display 2: Session Editors in Workspace

When opening SAS session in LSAF, an auto executed SAS program is executed at server initialization. Which creates several SAS macro variables, this maybe the main difference between coding in LSAF and SAS foundation. These variables specify the locations of folders or file path in workspace.

In LSAF, the path displayed in location toolbar is a path relative to the workspace. The pre-defined macro variable `_%SASWS_` can be used to refer to the location of the root-level folder for the workspace.



What's your data flow in SAS® LSAF(How to code in LSAF), continued

Display 3: The path displayed in location toolbar

Define libref in my SAS code:

```
LIBNAME surdata "&_SASWS_/SAS/Files/Survival Analysis";
```

Define Parameters:

For analyzing different source data file without modifying my code, I use %GLOBAL statement to set parameter, then specify parameter value when creating/running Job is an efficient way to run SAS program with different situation. The Job parameter value will be passed to the SAS program when it is running to get expected result.

```
%GLOBAL Input_file;
```

```
%MACRO check_params;
  %IF ("&Input_file"="") %THEN
    %LET Input_file=&_SASWS_/SAS/Files/Survival Analysis/testdata.csv;
  %MEND;
%check_params;

DATA surdata.survivaldataset;
infile "&Input_file" dlm=',';
input USUBJID ARM DUR STATUS;
run;
```

I was notified by CRO data provider that the survival analysis data will be provided in a CSV file, so I create a small test file to help to debug my SAS program. Then I create a SAS data set from the CSV file for PROC LIFETEST to do survival analysis, the CSV file path references the global variable in code. After I create job for this SAS program, when I run the Job I have the ability to select another data file to do analysis without modify my code.

The types of parameters for SAS programs are listed in Table 1, I might use them in my other code, in this paper, I only use file parameter and folder parameter.

Type	Description
Character	Any printable character.
Character (masked)	The same as Character. Each character is replaced by * for enhanced security. Note: A default value cannot be specified for a Character (masked) parameter.
Date	A date.
File	A file, including the absolute or relative path.
Folder	A folder, including the absolute or relative path.
Numeric	A numeric value.

Table1: Type of SAS program parameters

Review result in LSAF:

After adding PROC LIFETEST steps in my SAS program, my first version of SAS program is completed. Then I submit the code. After SAS program runs completely, I get 0 errors and 0 warnings in log. Meanwhile, I can see the

What's your data flow in SAS® LSAF(How to code in LSAF), continued

result in listing section, and the data set from Input/Output section.

The screenshot shows the SAS LSAF interface with the 'Listing' section selected. The main window displays the output of a LIFETEST procedure. The output is organized into sections for two strata: Stratum 1 (ARM = 1) and Stratum 2 (ARM = 2). Each stratum includes Product-Limit Survival Estimates and Summary Statistics for the time variable DUR.

Stratum 1: ARM = 1

Product-Limit Survival Estimates

DUR	Survival	Failure	Survival Standard Error	Number Failed	Number Left
0.0000	1.0000	0	0	0	9
1.0000	0.8889	0.1111	0.1048	1	8
3.0000	0.7778	0.2222	0.1386	2	7
6.0000	0.6667	0.3333	0.1571	3	6
11.0000	0.5556	0.4444	0.1656	4	5
13.0000	0.4444	0.5556	0.1656	5	4
17.0000	0.3333	0.6667	0.1571	6	3
18.0000*	.	.	.	6	2
21.0000*	.	.	.	6	1
24.0000	0	1.0000	.	7	0

NOTE: The marked survival times are censored observations.

Summary Statistics for Time Variable DUR

Quartile Estimates

Percent	Point Estimate	Transform	95% Confidence Interval (Lower)	Upper
75	24.0000	LOGLOG	11.0000	24.0000
50	13.0000	LOGLOG	1.0000	24.0000
25	6.0000	LOGLOG	1.0000	13.0000

Mean: 13.6667, Standard Error: 3.1111

Stratum 2: ARM = 2

Product-Limit Survival Estimates

DUR	Survival	Failure	Survival Standard Error	Number Failed	Number Left
0.0000	1.0000	0	0	0	9

Display 4: Code result in Listing section

What's your data flow in SAS® LSAF(How to code in LSAF), continued

Columns

- 123 USUBJID
- 123 ARM
- 123 DUR
- 123 STATUS

Attribute	Value
Name	USUBJID
Label	
Length	8
Type	num
Format	
Informat	
Varnum	1
Sorted By	0
Index Type	
Not Null	no
Position	0
Transcoded	yes

Obs	USUBJID	ARM	DUR	STATUS
1	1	1	1	1
2	2	2	2	0
3	3	3	1	3
4	4	4	2	5
5	5	5	1	6
6	6	6	2	9
7	7	7	2	10
8	8	8	1	11
9	9	9	2	12
10	10	10	1	13
11	11	11	2	14
12	12	12	1	17
13	13	13	1	18
14	14	14	2	19
15	15	15	1	21
16	16	16	2	23
17	17	17	1	24
18	18	18	2	25

Rows: 18 Filtered rows: 18 Columns: 4 Selected row: 0

Display 5: Review data set from Input/Output section

Version Control:

After I finish my code, I'm not very confidence about it as I'm a junior SAS programmer, so I check in the SAS program and input data file to repository while enable versioning. Then let my colleague – a senior SAS programmer to review my code. Although we don't work in same place, I don't need to send code and test file to him by email or put them to other shared location. The senior SAS programmer only needs to sign in LSAF, check out the SAS program and test data file to his workspace, review and modify my code, then check in them to repository. I will see the repository version of my code is different with the one in my workspace. But I don't worry about losing my previous version, anytime I can copy any version of file to my workspace. All versions are stored in server, this keeps file secure and traceable. Version control can be applied to any type of files in LSAF, SAS program, Job, data set, pdf, csv, etc, as long as enable versioning when uploading files to repository or check in files to repository.

Name	Size	Modified By	Date Modified	Signed	Repository Version	Workspace Status
SurvivalAnalysis2.sas	822 bytes	Lilly	May 26, 2017 01:54 P...		1.1	1.0

Display 6: Version changed in repository

What's your data flow in SAS® LSAF(How to code in LSAF), continued

Name	Location	Size	Date Modified	Repository Version	Workspace St...
SurvivalAnalysis2.sas	/SAS/Files/Survival Analysis	822 bytes	May 26, 2017 01:54 PM ...	1.1 v 1.1 1.0	⚠ 1.0

Display 7: Copy any version of file to workspace

I check out the latest SAS program to my workspace, and see the senior SAS programmer added a ODS statement to create output PDF file, so we can let statistician review result via PDF file, not just listing file. Which will be more flexible. And he also modified PROC LIFETEST method to LT, because the data file provided by CRO will create a large data set. This method is better than km for large data set. For more convenient if I want to re-use this SAS program to analyze other data and create another output file, I modify the code again to use a parameter passing the output file location.

Code in version 1.0:

```
proc lifetest DATA=surdata.survivaldata method=km plots=(survival) OUTSURV=ttesurv;
strata ARM;
time DUR*STATUS(0);
run;
```

Code in version 1.1:

```
ods pdf file="& SASWS_/SAS/Files/Survival Analysis/SurvivalAnalysis.pdf";
proc lifetest DATA=surdata.survivaldata method=LT plots=(survival) OUTSURV=ttesurv;
strata ARM;
time DUR*STATUS(0);
run;
ods pdf close;
```

Code in version 1.2:

```
%GLOBAL Input_file;
%GLOBAL Output_folder;
```

```
%MACRO check_params;
%IF ("&Input_file"="") %THEN
  %LET Input_file=%_SASWS_/SAS/Files/Survival Analysis/testdata.csv;
%IF ("&Output_folder"="") %THEN
  %LET Output_folder=%_SASWS_/SAS/Files/Survival Analysis;
%MEND;
%check_params;
```

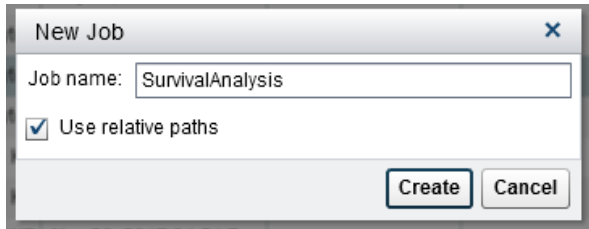
```
ODS pdf file="&Output_folder/SurvivalAnalysis.pdf";
PROC LIFETEST DATA=surdata.survivaldataset method=LT plots=(survival) OUTSURV=ttesurv;
strata ARM;
time DUR*STATUS(0);
run;
ODS pdf close;
```

CREATE JOB TO RUN SAS PROGRAM IN REPOSITORY

Now, all coding work are done. Next, I will create Job using the SAS program as I want to run SAS program in repository. When job is published to repository, all users who have permission can run it.

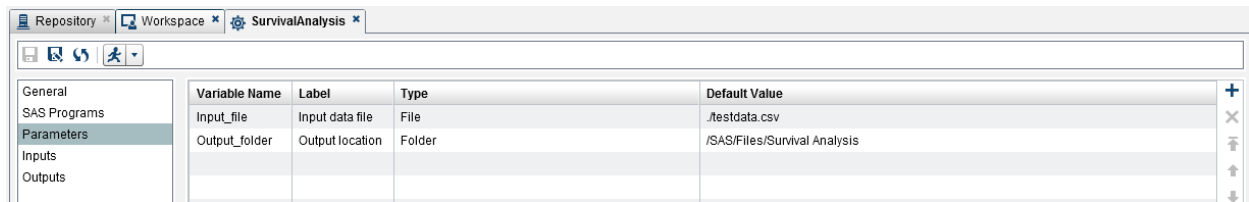
The first step is creating the job in my workspace. Check the "Use relative paths" option will make my job more flexible, copy or move it to other location, it could no need to modify locations in it again.

What's your data flow in SAS[®] LSAF(How to code in LSAF), continued



Display 8: Create new Job

After the job is created, I can specify additional Job information. A job can run many SAS programs in sequence. In this paper, I only run one SAS program. Then specify parameters defined in SAS program if it has. The parameters type was introduced in “Define Parameters” section of this paper. Although I'm using relative path here for file type parameter, but once the job runs, it will be resolved to absolute path and pass it to my SAS code. The parameters might be changed in the run job dialog after clicking Run button, this will be very convenient to analyze different source file without modifying my SAS code. If next time I want to do survival analysis on another data, I don't need to do more extra work, just run this job, select another input data file and output location on UI, then click “Run”. My work is done! Also specify Inputs and Outputs files and locations if the SAS programs are referencing them.



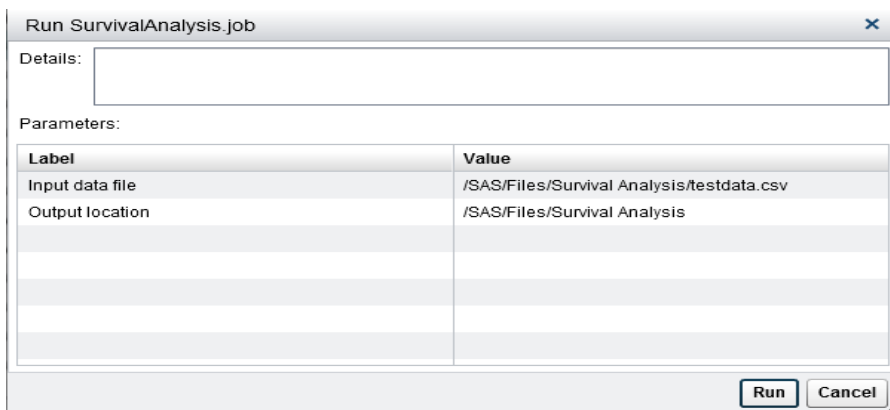
Display 9: Specify parameters in Job

Run Job in Repository:

After checking in my Job and related SAS programs, Inputs and Outputs files to repository, all users in my organization who have permissions can run my job now.

In SAS LSAF, there are three ways to run a job in repository: run interactively, run when scheduled time is up or run in process flow.

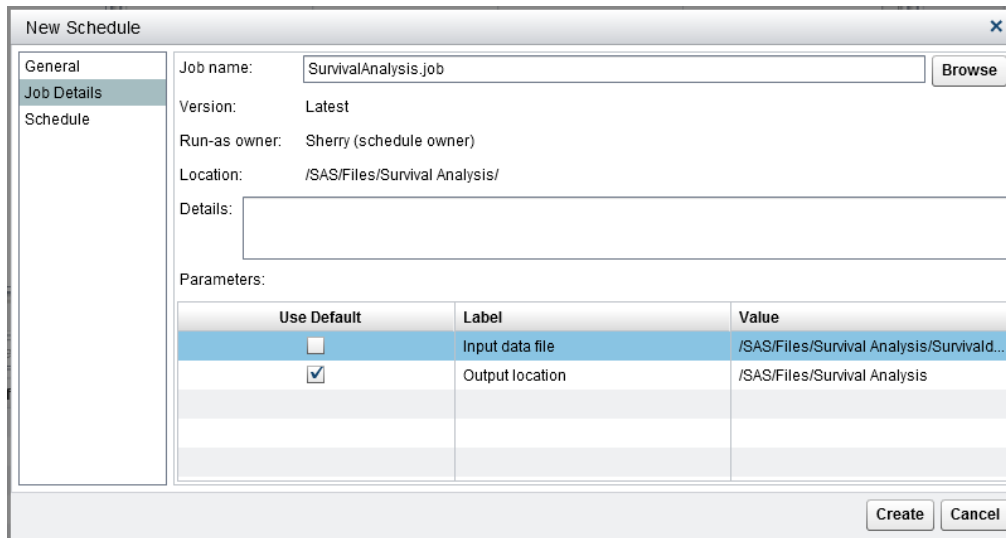
- **Run job interactively:** Click “Run” button when selecting the job in repository, run job dialog will show up if the job has parameters. If no parameter, the job will run directly. The parameters could be changed at this time. After clicking “Run” button, my Job will run with the default or changed parameters and I can get job run status from action status report window.



Display 10: Run job dialog

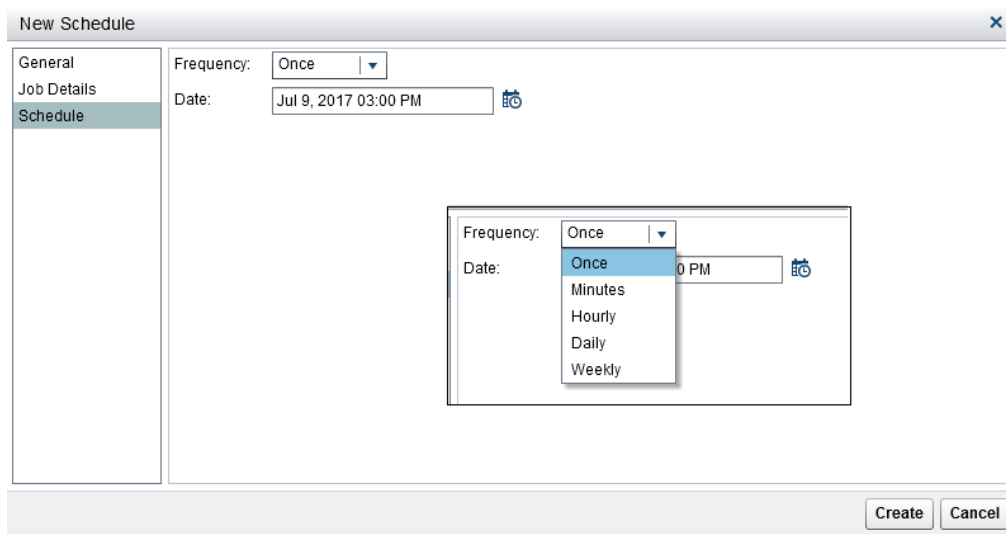
- **Run job when scheduled time is up:** If CRO data provider told me that he will upload the data on a certain time, I can create a schedule to run the job on the certain time after specifying the job location and parameters in Job Details section. The job will run when the time is up automatically. In other case, a job might need to run frequently, setting frequency when creating the schedule by minutes, hourly, daily, weekly depends on project needs.

What's your data flow in SAS® LSAF(How to code in LSAF), continued



Use Default	Label	Value
<input type="checkbox"/>	Input data file	/SAS/Files/Survival Analysis/Survivald...
<input checked="" type="checkbox"/>	Output location	/SAS/Files/Survival Analysis

Display 11: Specify Job in Schedule



Display 12: Schedule Job

- **Run job in Process Flow:** A job also can be run in Process Flow, the details will be introduced in next section.

SETTING AND ACTIVATE PROCESS FLOW TO WORK EFFECTIVELY

I have built an effective process flow definition file in first section of this paper using Camunda Modeler. I upload it to LSAF, deploy process flow definition file and create a process flow use it.

Now let me step one by one to perform this process flow to see how it works effectively for data analysis.

Step1: Flow Setup:

- 1.1 Set up signal event: Once the files or locations are updated or modified, the signal event will be triggered, the flow will go to next element.

What's your data flow in SAS® LSAF(How to code in LSAF), continued

Set Up: SignalEvent

General

Signal Details

Location

- /SAS/Files/Survival Analysis/survivaldata.csv
- /SAS/Files/Survival Analysis/SurvivalAnalysis2.sas

Display 13: Set up signal event

- 1.2 Set up Job, similar with set up Job details in schedule, refer to **Display10**.
- 1.3 Set up notification to programmer or reviewer: Using the expression `${RunJob.completionStatus}` in email subject or body, this expression will be replaced by the status of a completed job.

Set Up: Notify_Programmer

General

Notification Details

To: Sherry

Priority: Medium

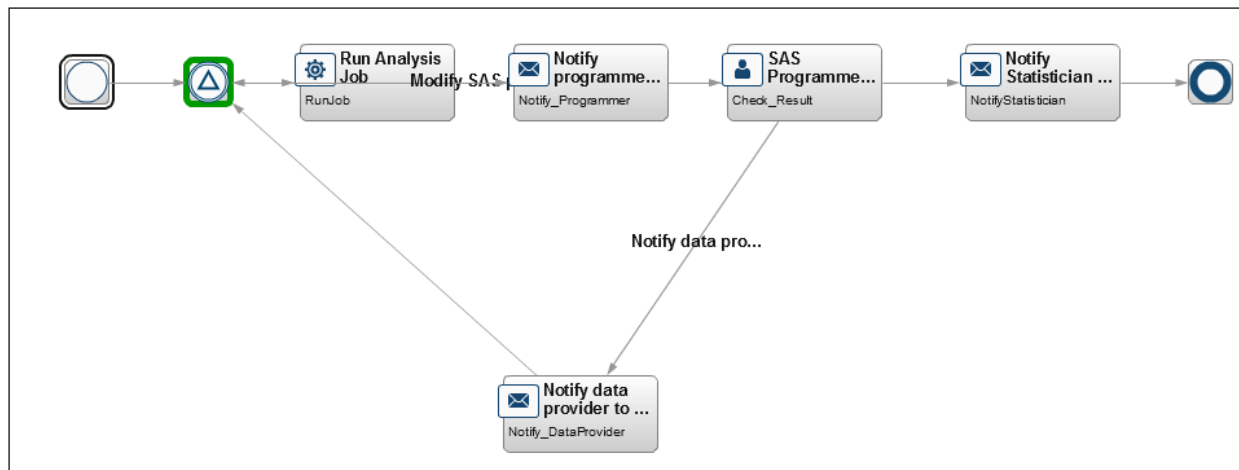
Subject: Job run completed, please review result

Message: `${RunJob.completionStatus}`

Display 14: Set up notification

- 1.4 Set up notification to data provider, statistician. Similar with step 1.3.

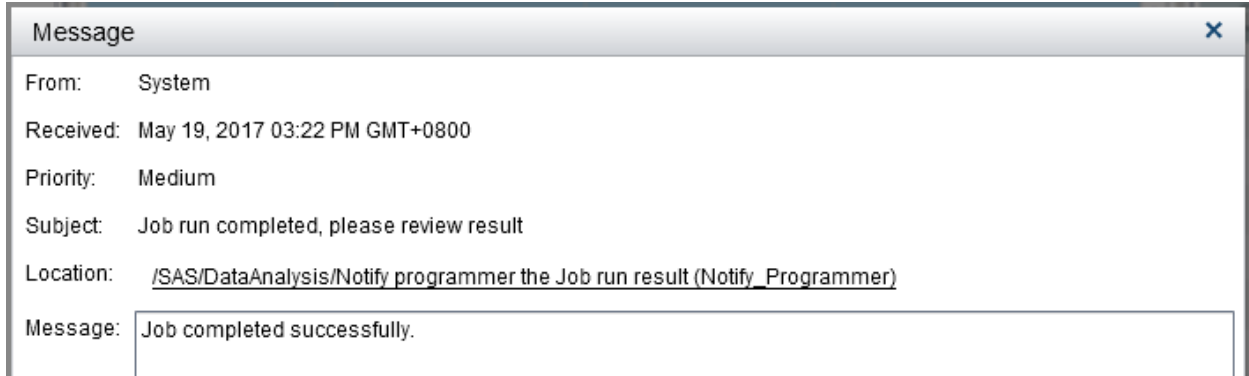
Step2: Activate the process flow



Display 15: View diagram in LSAF

The process flow is waiting at signal event. I tell the CRO data provider if the data is ready, please upload it to the location I specified and rename it as I specified. Then I don't need to know when the data will be uploaded. Once the data is uploaded, the process flow will go to next node- the job will run automatically. After the job is completed, I will receive a message/email to know the job completion status.

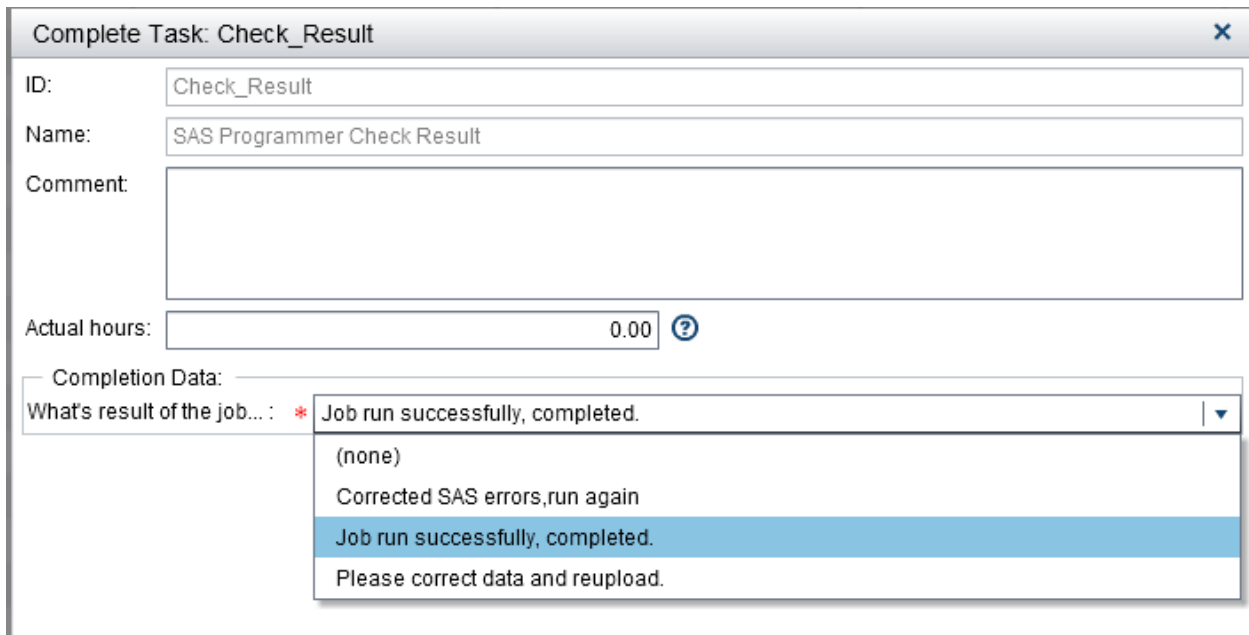
What's your data flow in SAS[®] LSAF(How to code in LSAF), continued



Display 16: Message received from system after Job run completed

Step3: Select completion data

If the job run result is expected, I set completion data to “completed” when completing user task. If there is SAS program errors, then select “Correct SAS errors, run again”, the process flow will go back to signal event, once the program is modified and check in, the process flow will start to run job again. If the job run fails because of data errors, select “Please correct data”, then the notification will send to data provider. Once the new data is uploaded, the signal event will trigger run job again. The completion data is defined when creating the process flow definition file. The circle can be repeated again and again, until the final result is expected. Then the process flow can be completed.



Display 17: User task's completion data

After Job run successfully, system will send a message/email to the statistician in Shanghai, who can sign in LASF and review the PDF file in LSAF. Then the whole work is completed.

All programmers in my company can use this process flow to do data analysis.

CONCLUSION

In SAS LSAF, I can develop, debug my SAS program on any computer with supported browsers. I can also share data, information, programs with my co-workers all over the world. Using version control, my organization can store versions of a file in the repository which keep data secure and traceable. Jobs and process flows make my work more effective and efficient.

What's your data flow in SAS[®] LSAF(How to code in LSAF), continued

REFERENCES

SAS[®] Life Science Analytics Framework 4.7: User's Guide

<http://support.sas.com/documentation/onlinedoc/develop/index.html>

Survival Analysis Translation from Statistician to Programmer

<http://www.lexjansen.com/pharmasug-cn/2016/DS/PharmaSUG-China-2016-DS07.pdf>

ACKNOWLEDGMENTS

I want to give big thanks to my manager, Han Liu, who encourages me to write this paper and gives me very valuable advice.

Also thank all the reviewers who give me valuable comments and make the paper got improved.

CONTACT INFORMATION

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

Name: Sherry (Shuang) Fu
Enterprise: SAS Research and Development (Beijing) Co., Ltd.
Address: Motorola Plaza, No. 1 Wang Jing East Road
City, State ZIP: Beijing, 100102
Work Phone: (8610) 83193355-3657
Fax: (8610) 6310-9130
E-mail: Shuang.fu@sas.com
Web: 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.