PharmaSUG 2013 - Paper PO10

How to make SAS Drug Development more efficient

Xiaopeng Li, Celerion Inc., Lincoln, NE
Chun Feng, Celerion Inc., Lincoln, NE
Peng Chai, Celerion Inc., Lincoln, NE

ABSTRACT

SAS Drug Development (SDD) is a web-based SAS system on UNIX operating system. Internet status, limited abilities to run multiple processes at a time and some SDD unique features take more time to operate a program in SDD. To run SDD efficiently is important for accomplishing tasks for programmers. In this paper, several methods that can make SDD more efficient will be examined.

INTRODUCTION

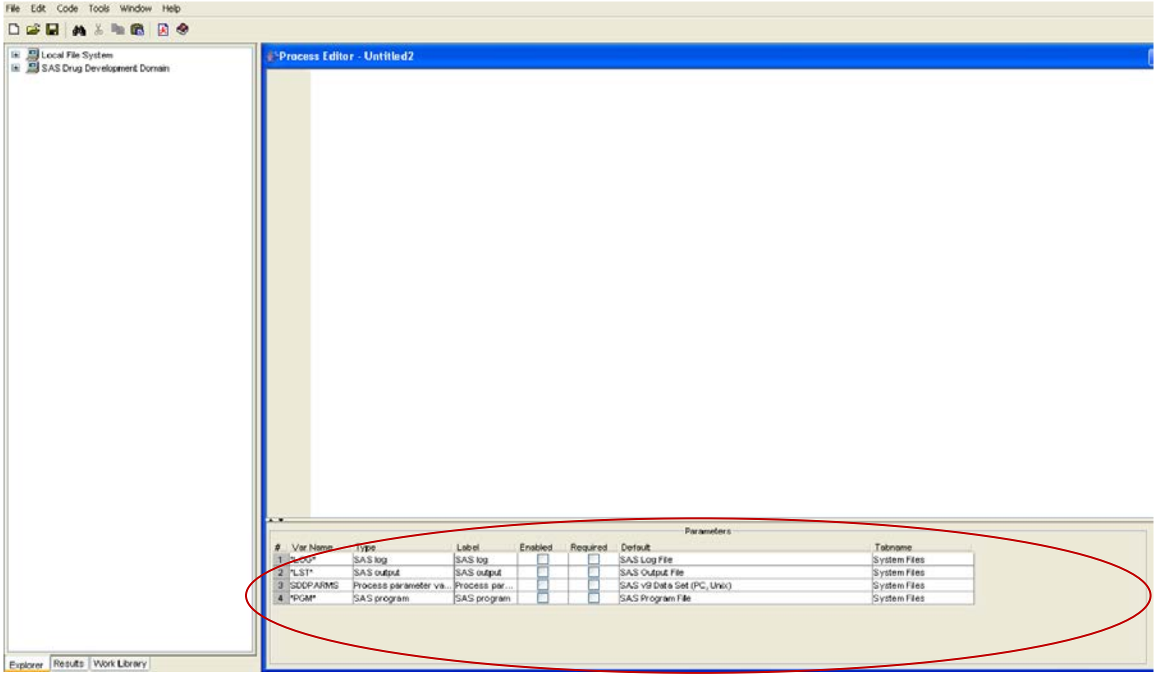
SAS® Drug Development (SDD) is a web-based SAS system with UNIX operating system with a user interface platform. SDD is developed by SAS for efficiently executing SAS programs, managing database, and reporting clinical research results. It contains some programming-friendly features such as parameter window and scheduler.

Recently, several pharmaceutical, biotechnology companies, and clinical research organizations start to use SDD to analyze and report clinical trial data. Reading or storing information in SDD takes more time due to the web-based distance transmission. To help programmers complete tasks with less time, several methods for improving SDD efficiency are discussed.

METHODS

CALLING IN APPROPRIATE MACROS AND DATA SETS FROM PARAMETER WINDOW

In SDD, calling in macros and data sets are different from PC SAS®. In SDD, the parameter window can define and locate macros and data sets (Display 1).



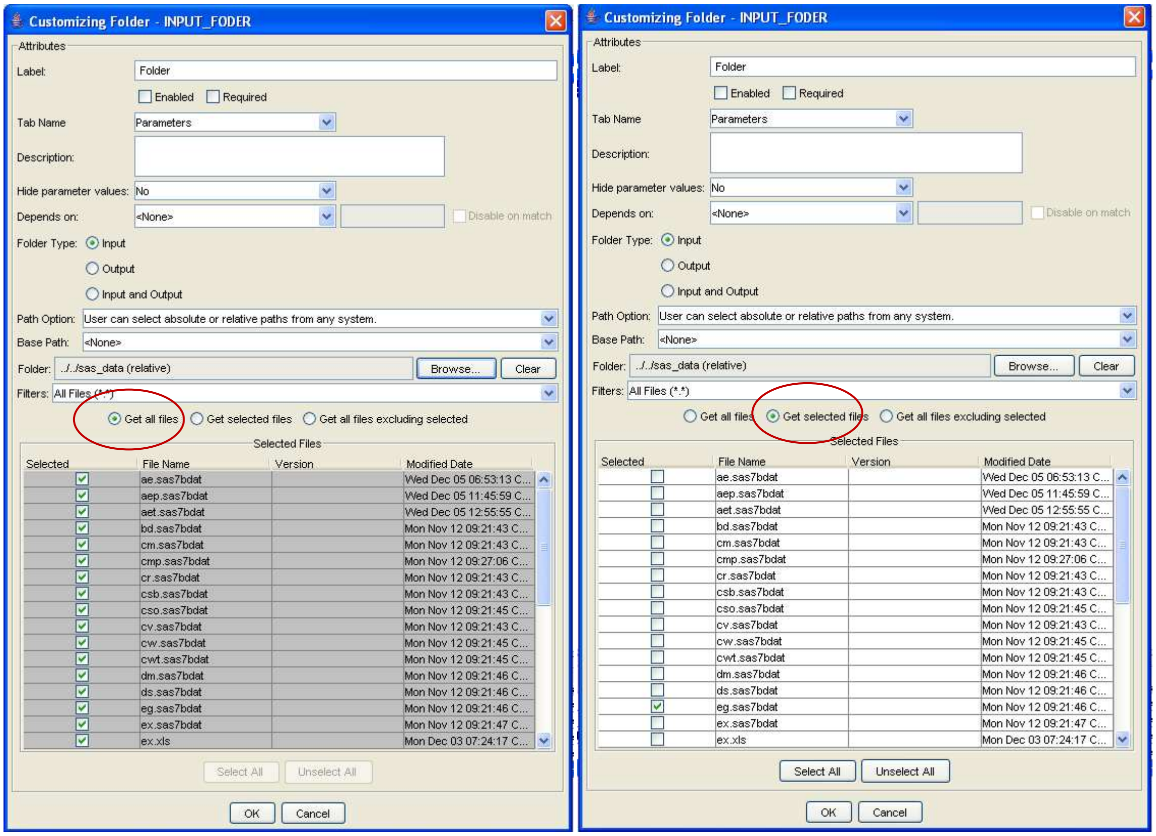
Parameters

#	Var Name	Type	Label	Enabled	Required	Default	Tabname
1	*LOG*	SAS log	SAS log	□	□	SAS Log File	System Files
2	*LST*	SAS output	SAS output	□	□	SAS Output File	System Files
3	SDDPARMS	Process parameter va...	Process par...	□	□	SAS v9 Data Set (PC, Unix)	System Files
4	*PGM*	SAS program	SAS program	□	□	SAS Program File	System Files

Display 1. Parameter window in SDD

Since SDD is a web-based system, to execute programs on SDD requires sending and receiving information to the server. The more data and codes are involved, the longer it will take to execute a program. When executing a program, eliminating useless data, macros and codes can improve SDD efficiency.

To define SAS data sets location in SDD parameter window, the location should be set as a folder. In SDD parameter window, there are three kinds of folder: “Input”, “Output” and “Input and Output”. The “Input” folder defines the location of a folder for data used in a program. One or more data sets from the defined folder could be read in when a program is executed. If the input folder is defined as “Get all files”, all the SAS data in the defined folder will be read in. If the input folder is defined as “Get selected files”, only selected files in the defined folder will be read in (Display 2). The “Output” folder defines the location of a folder where an output data from a program will be saved. There is no need to specify any data in the “Output” folder (Display 3). The “Input and Output” folder defines the location of a folder where data will be read from and saved to. Similar to the “Input” folder, read-in data sets need to be specified (Display 4).



Customizing Folder - INPUT_FODER

Attributes

Label: Folder

□ Enabled □ Required

Tab Name: Parameters

Description:

Hide parameter values: No

Depends on: <None> □ Disable on match

Folder Type: ● Input ○ Output ○ Input and Output

Path Option: User can select absolute or relative paths from any system.

Base Path: <None>

Folder: ../../sas_data (relative)

Filters: All Files (*.*)

● Get all files ○ Get selected files ○ Get all files excluding selected

Selected Files

Selected	File Name	Version	Modified Date
☑	ae.sas7bdat		Wed Dec 05 06:53:13 C...
☑	aep.sas7bdat		Wed Dec 05 11:45:59 C...
☑	aet.sas7bdat		Wed Dec 05 12:55:55 C...
☑	bd.sas7bdat		Mon Nov 12 09:21:43 C...
☑	cm.sas7bdat		Mon Nov 12 09:21:43 C...
☑	cmp.sas7bdat		Mon Nov 12 09:27:06 C...
☑	cr.sas7bdat		Mon Nov 12 09:21:43 C...
☑	csb.sas7bdat		Mon Nov 12 09:21:43 C...
☑	cso.sas7bdat		Mon Nov 12 09:21:45 C...
☑	cv.sas7bdat		Mon Nov 12 09:21:43 C...
☑	cw.sas7bdat		Mon Nov 12 09:21:45 C...
☑	cwt.sas7bdat		Mon Nov 12 09:21:45 C...
☑	dm.sas7bdat		Mon Nov 12 09:21:46 C...
☑	ds.sas7bdat		Mon Nov 12 09:21:46 C...
☑	eg.sas7bdat		Mon Nov 12 09:21:46 C...
☑	ex.sas7bdat		Mon Nov 12 09:21:47 C...
☑	ex.xls		Mon Dec 03 07:24:17 C...

Customizing Folder - INPUT_FODER

Attributes

Label: Folder

□ Enabled □ Required

Tab Name: Parameters

Description:

Hide parameter values: No

Depends on: <None> □ Disable on match

Folder Type: ● Input ○ Output ○ Input and Output

Path Option: User can select absolute or relative paths from any system.

Base Path: <None>

Folder: ../../sas_data (relative)

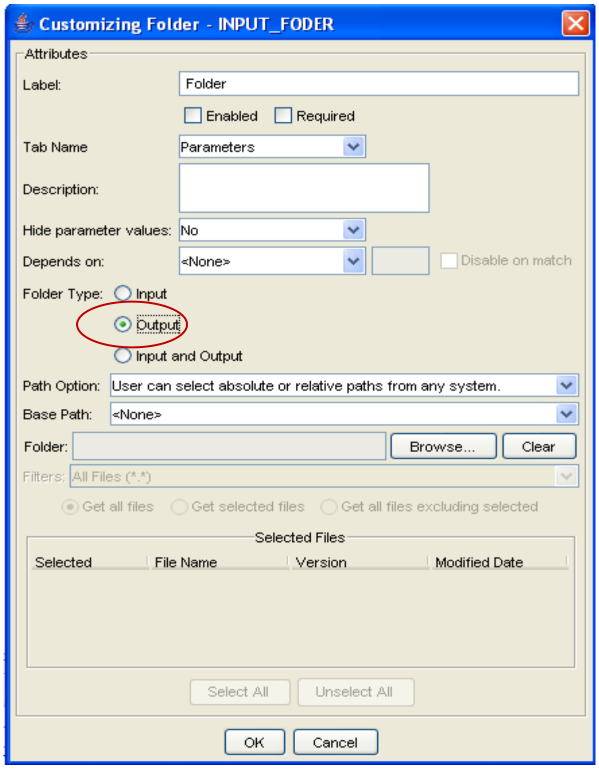
Filters: All Files (*.*)

○ Get all files ● Get selected files ○ Get all files excluding selected

Selected Files

Selected	File Name	Version	Modified Date
□	ae.sas7bdat		Wed Dec 05 06:53:13 C...
□	aep.sas7bdat		Wed Dec 05 11:45:59 C...
□	aet.sas7bdat		Wed Dec 05 12:55:55 C...
□	bd.sas7bdat		Mon Nov 12 09:21:43 C...
□	cm.sas7bdat		Mon Nov 12 09:21:43 C...
□	cmp.sas7bdat		Mon Nov 12 09:27:06 C...
□	cr.sas7bdat		Mon Nov 12 09:21:43 C...
□	csb.sas7bdat		Mon Nov 12 09:21:43 C...
□	cso.sas7bdat		Mon Nov 12 09:21:45 C...
□	cv.sas7bdat		Mon Nov 12 09:21:43 C...
□	cw.sas7bdat		Mon Nov 12 09:21:45 C...
□	cwt.sas7bdat		Mon Nov 12 09:21:45 C...
□	dm.sas7bdat		Mon Nov 12 09:21:46 C...
□	ds.sas7bdat		Mon Nov 12 09:21:46 C...
☑	eg.sas7bdat		Mon Nov 12 09:21:46 C...
□	ex.sas7bdat		Mon Nov 12 09:21:47 C...
□	ex.xls		Mon Dec 03 07:24:17 C...

Display 2. Input folder with get all files option (left) and Input folder with get selected files option (right)



Attributes

Label: Folder

□ Enabled □ Required

Tab Name: Parameters

Description:

Hide parameter values: No

Depends on: <None> □ Disable on match

Folder Type: ○ Input ● Output ○ Input and Output

Path Option: User can select absolute or relative paths from any system.

Base Path: <None>

Folder:

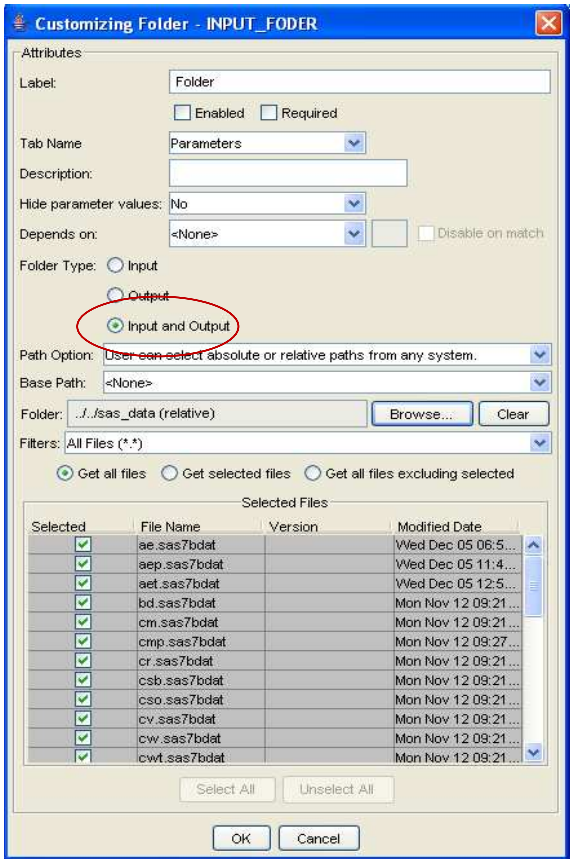
Filters: All Files (*.*)

● Get all files ○ Get selected files ○ Get all files excluding selected

Selected Files

Selected	File Name	Version	Modified Date

Display 3. Output folder



Attributes

Label: Folder

□ Enabled □ Required

Tab Name: Parameters

Description:

Hide parameter values: No

Depends on: <None> □ Disable on match

Folder Type: ○ Input ○ Output ● Input and Output

Path Option: User can select absolute or relative paths from any system.

Base Path: <None>

Folder: ../../sas_data (relative)

Filters: All Files (*.*)

● Get all files ○ Get selected files ○ Get all files excluding selected

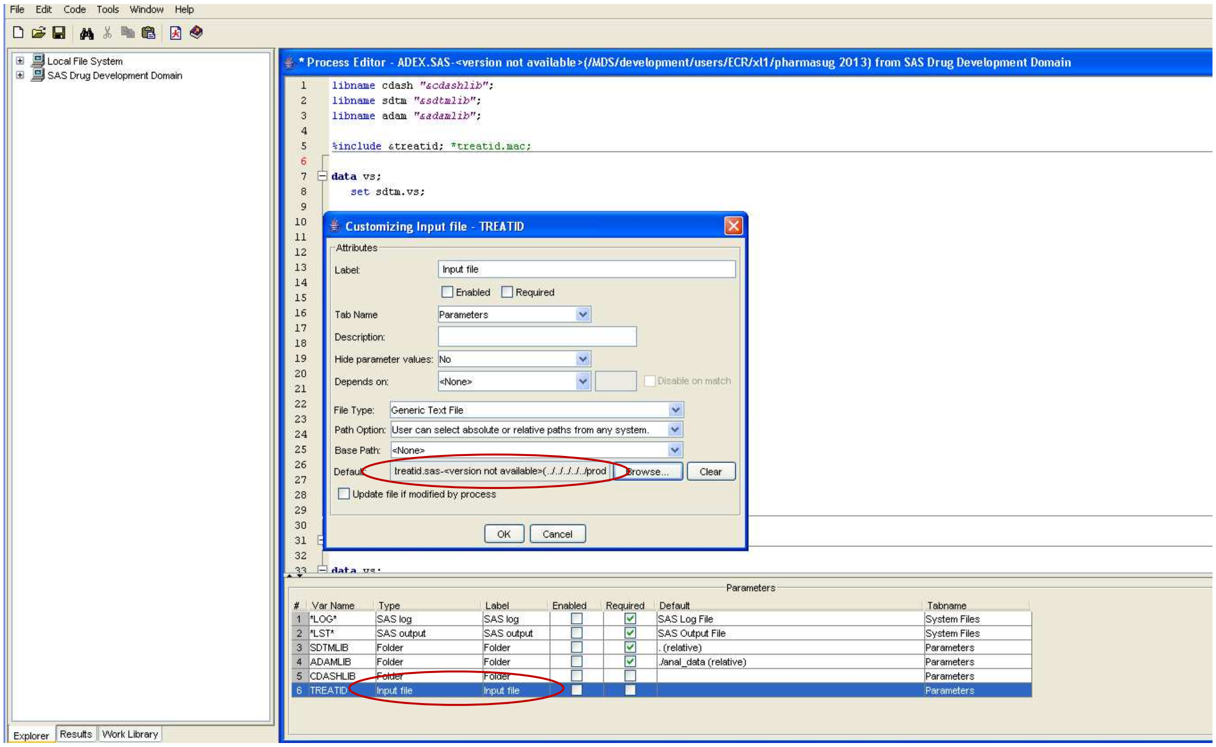
Selected Files

Selected	File Name	Version	Modified Date
☑	ae.sas7bdat		Wed Dec 05 06:5...
☑	aep.sas7bdat		Wed Dec 05 11:4...
☑	aet.sas7bdat		Wed Dec 05 12:5...
☑	bd.sas7bdat		Mon Nov 12 09:21...
☑	cm.sas7bdat		Mon Nov 12 09:21...
☑	cmp.sas7bdat		Mon Nov 12 09:27...
☑	cr.sas7bdat		Mon Nov 12 09:21...
☑	csb.sas7bdat		Mon Nov 12 09:21...
☑	cso.sas7bdat		Mon Nov 12 09:21...
☑	cv.sas7bdat		Mon Nov 12 09:21...
☑	cw.sas7bdat		Mon Nov 12 09:21...
☑	cwt.sas7bdat		Mon Nov 12 09:21...

Display 4. Input and Output folder

The data to be used should be defined clearly to avoid missing data or reading in data multiple times. In addition, avoiding calling in data that will not be used in a program helps SDD work more efficiently.

Macros in SDD can be called into a program by two ways. First, similar to SAS data sets, macros can be called from a defined folder. It is more efficient to call-in only the macros that will be used in the program. Second, macros can be called in as input files. As shown in Display 5, the name and location of an input file are referenced in the parameter window. The second way is more efficient and preferable.



* Process Editor - ADEX.SAS-<version not available>(/MDS/development/users/ECR/xl1/pharmasug 2013) from SAS Drug Development Domain

```
libname cdash "&cdashlib";
libname sdtm "&sdtmlib";
libname adam "&adamlib";

%include &treatid; *treatid.mac;

data vs;
   set sdtm.vs;
```

```
data vs;
```

Customizing Input file - TREATID

Attributes

Label: Input file

☐ Enabled ☐ Required

Tab Name: Parameters

Description:

Hide parameter values: No

Depends on: <None> ☐ Disable on match

File Type: Generic Text File

Path Option: User can select absolute or relative paths from any system.

Base Path: <None>

Defaul treatid.sas-<version not available>(../../../../../prod

☐ Update file if modified by process

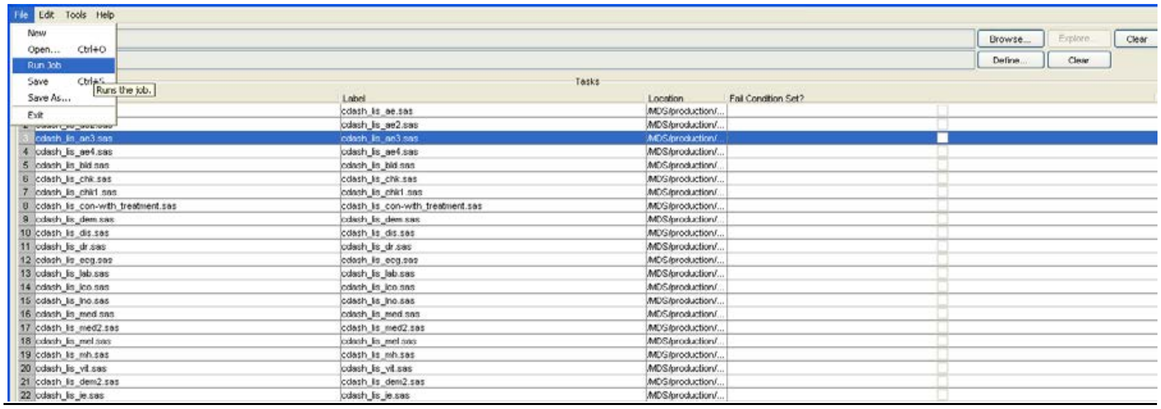
Parameters

#	Var Name	Type	Label	Enabled	Required	Default	Tabname
1	*LOG*	SAS log	SAS log	☐	☑	SAS Log File	System Files
2	*LST*	SAS output	SAS output	☐	☑	SAS Output File	System Files
3	SDTMLIB	Folder	Folder	☐	☑	. (relative)	Parameters
4	ADAMLIB	Folder	Folder	☐	☑	./anal_data (relative)	Parameters
5	CDASHLIB	Folder	Folder	☐	☐		Parameters
6	TREATID	Input file	Input file	☐	☐		Parameters

Display 5. Input file

USING JOB EDITOR

Job editor which can execute all the programs together instead of opening all the programs one by one is a unique feature in SDD (Display 6).



Tasks

		Label	Location	Fail Condition Set?
1		cdash_lis_ae.sas	/MDS/production/...	☐
2		cdash_lis_ae2.sas	/MDS/production/...	☐
3	cdash_lis_ae3.sas	cdash_lis_ae3.sas	/MDS/production/...	☐
4	cdash_lis_ae4.sas	cdash_lis_ae4.sas	/MDS/production/...	☐
5	cdash_lis_bld.sas	cdash_lis_bld.sas	/MDS/production/...	☐
6	cdash_lis_chk.sas	cdash_lis_chk.sas	/MDS/production/...	☐
7	cdash_lis_chk1.sas	cdash_lis_chk1.sas	/MDS/production/...	☐
8	cdash_lis_con-with_treatment.sas	cdash_lis_con-with_treatment.sas	/MDS/production/...	☐
9	cdash_lis_dem.sas	cdash_lis_dem.sas	/MDS/production/...	☐
10	cdash_lis_dis.sas	cdash_lis_dis.sas	/MDS/production/...	☐
11	cdash_lis_dr.sas	cdash_lis_dr.sas	/MDS/production/...	☐
12	cdash_lis_ecg.sas	cdash_lis_ecg.sas	/MDS/production/...	☐
13	cdash_lis_lab.sas	cdash_lis_lab.sas	/MDS/production/...	☐
14	cdash_lis_ico.sas	cdash_lis_ico.sas	/MDS/production/...	☐
15	cdash_lis_ino.sas	cdash_lis_ino.sas	/MDS/production/...	☐
16	cdash_lis_med.sas	cdash_lis_med.sas	/MDS/production/...	☐
17	cdash_lis_med2.sas	cdash_lis_med2.sas	/MDS/production/...	☐
18	cdash_lis_mel.sas	cdash_lis_mel.sas	/MDS/production/...	☐
19	cdash_lis_mh.sas	cdash_lis_mh.sas	/MDS/production/...	☐
20	cdash_lis_vit.sas	cdash_lis_vit.sas	/MDS/production/...	☐
21	cdash_lis_dem2.sas	cdash_lis_dem2.sas	/MDS/production/...	☐
22	cdash_lis_ie.sas	cdash_lis_ie.sas	/MDS/production/...	☐

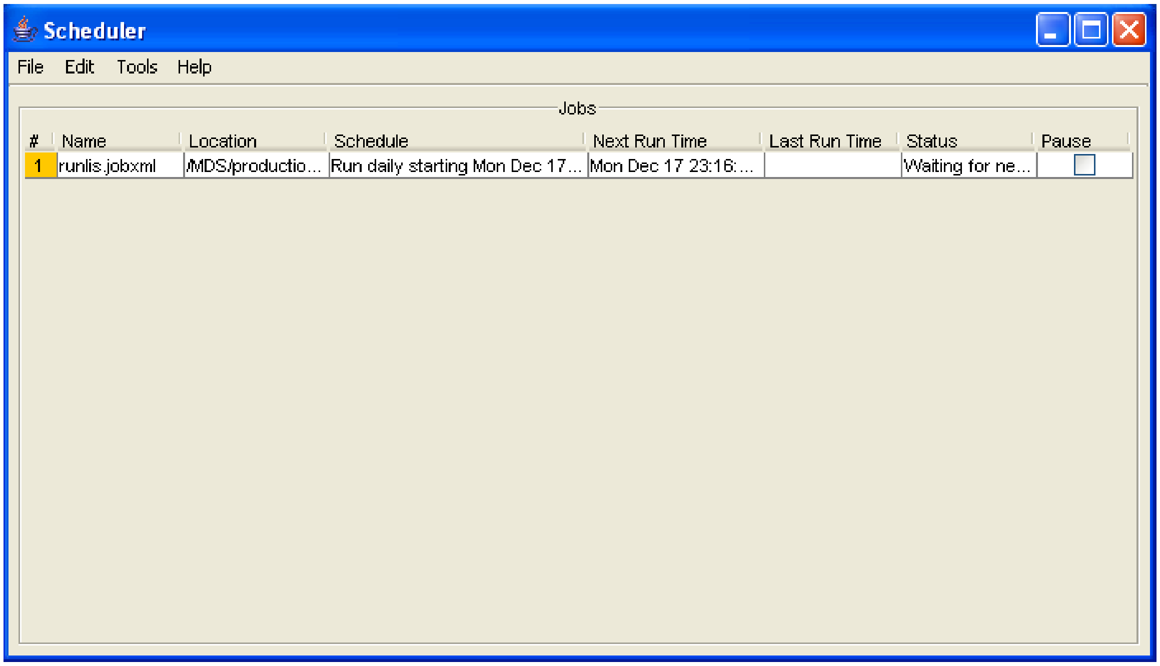
Display 6. Job editor

By using a Job editor, time is saved since we do not open individual SAS programs. However, if there are any errors within any individual program, job editor will not provide error messages before all the programs are executed. Moreover, errors cannot be corrected in a job editor. So we need open individual programs where errors occur to debug and fix. In that way, it will cost more time if any of the programs in the job editor has errors. The program with errors needs to be executed and updated separately. It is important to make sure that all programs work correctly before using the job editor.

USING SCHEDULER

Scheduler is another unique feature in SDD (Display 7). It can run a set of programs in order. Scheduler can define a time to execute one or more job editors. It can also define a frequency that you want to execute the scheduler, such as once, daily, weekly, and monthly. It makes SDD more efficient, especially for programs requiring long execution time.

SDD is limited to execute one program at one time. By using scheduler, more programs can be handled at the same time. It is better to use scheduler to execute programs requiring long execution time that are ready. At the same time, another program can be modified and executed.



#	Name	Location	Schedule	Next Run Time	Last Run Time	Status	Pause
1	runlis.jobxml	/MDS/productio...	Run daily starting Mon Dec 17...	Mon Dec 17 23:16:...		Waiting for ne...	□

Display 7. SDD Scheduler

USING APPROPRIATE DATA SETS

Creating tables, figures and listings (TFLs) is a common task for programmers. To make data ready for analysis, a program first needs to process many macros and loops which will cost time. Therefore, developing analysis data sets prior to TFL production will improve the efficiency, for example, generating TFLs from Analysis Data Model (ADaM) data. ADaM data sets contain baseline, analysis flags, grouping variables, and treatment variables to help create TFLs.

CONCLUSION

In conclusion, calling in appropriate macros and data sets from parameter window, using job editor, using scheduler, and using appropriate data sets can make SDD more efficient. Programmers can choose the combination of these methods to finish SDD tasks in less time.

REFERENCES

Clinical Data Interchange Standards Consortium, “ Analysis Data Model v2.1.” December 17th, 2009. Available at www.cdisc.org

Chun Feng, Xiaopeng Li, 2012, Export SAS Data to Excel in SAS® Drug Development, *Proceedings of the PharmaSUG 2012*, San Francisco, CA.

Xiaopeng Li, Katrina Canonizado, Chun Feng, Nancy Wang, 2012, Streamlining Regulatory Submission with CDISC/ADaM Standards for Non-standard Pharmacokinetic/Pharmacodynamic Analysis Datasets, *Proceedings of the PharmaSUG 2012*, San Francisco, CA.

CONTACT INFORMATION

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

Name: Xiaopeng Li
Enterprise: Celerion Inc.
Address: 621 Rose Street
City, State ZIP: Lincoln, NE 68502
Work Phone: 402-437-6260
E-mail: xiaopeng.li@celerion.com, xiaopeng.li@huskers.unl.edu

Name: Chun Feng
Enterprise: Celerion Inc.
Address: 621 Rose Street
City, State ZIP: Lincoln, NE 68502
E-mail: chun.feng@celerion.com

Name: Peng Chai
Enterprise: Celerion Inc.
Address: 621 Rose Street
City, State ZIP: Lincoln, NE 68502
Work Phone: 402-437-6260
E-mail: peng.chai@celerion.com