

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

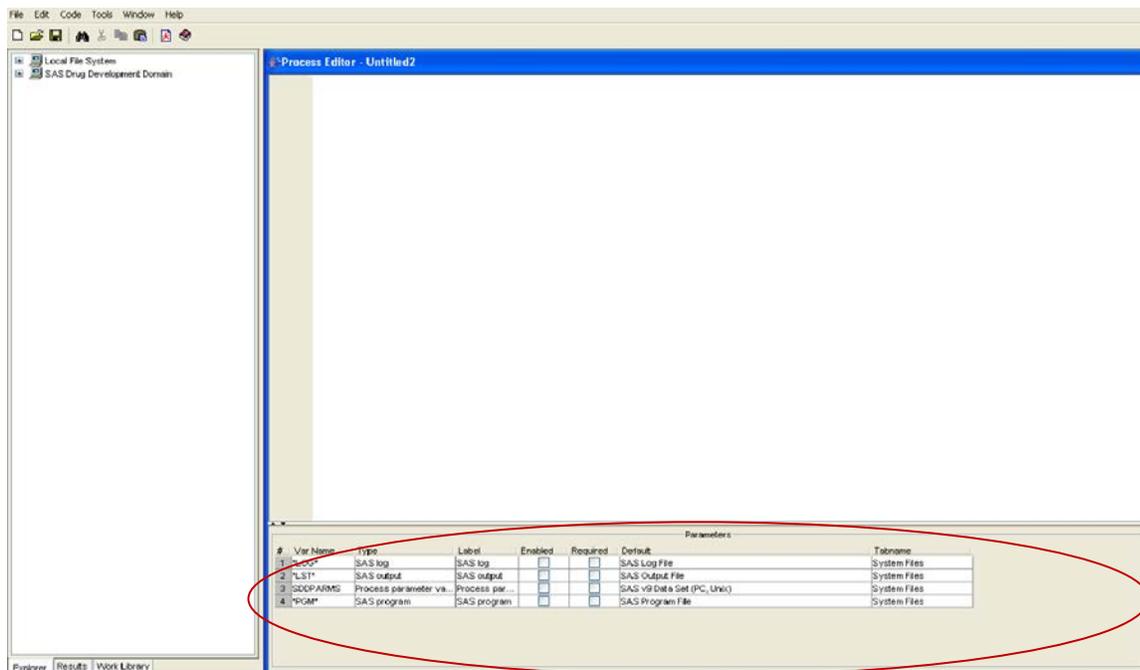
SAS[®] Drug Development (SDD) is a web-based SAS system with UNIX operating system with a user interface platform. SDD is developed by SAS for efficiently executing SAS programs, managing database, and reporting clinical research results. It contains some programming-friendly features such as parameter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

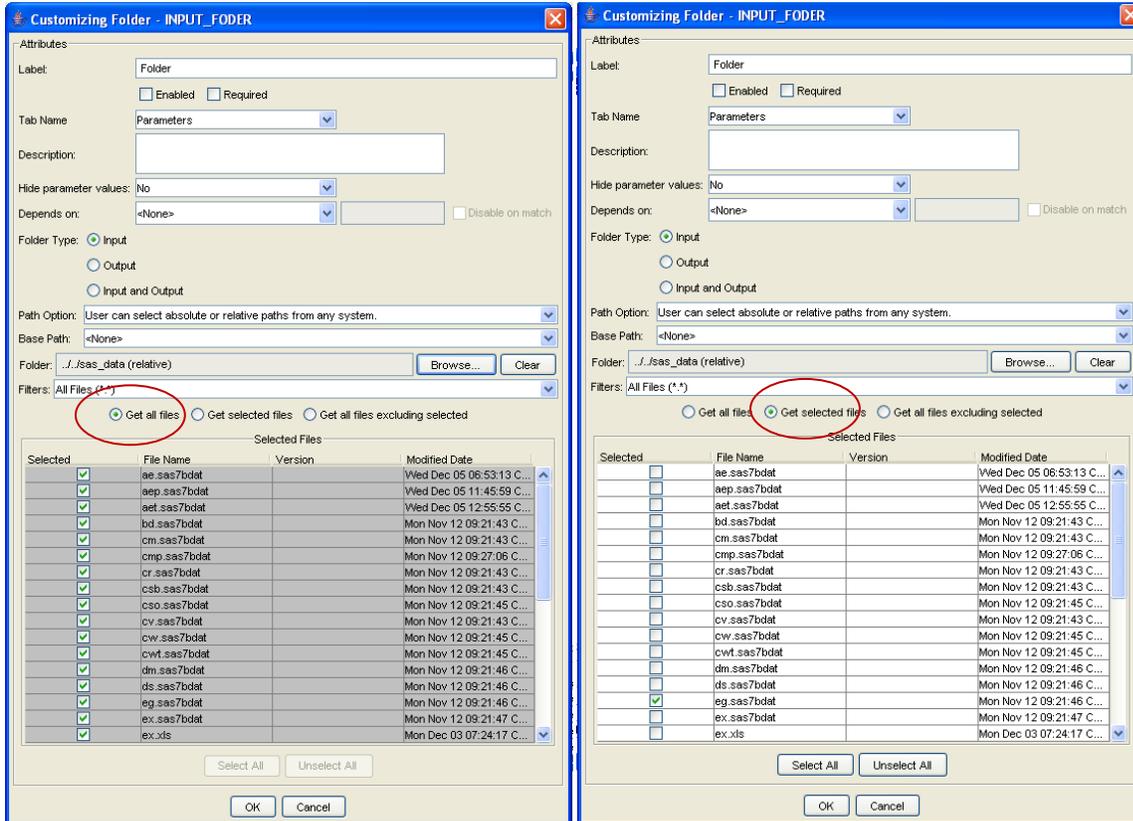


Display 1. Parameter window in SDD

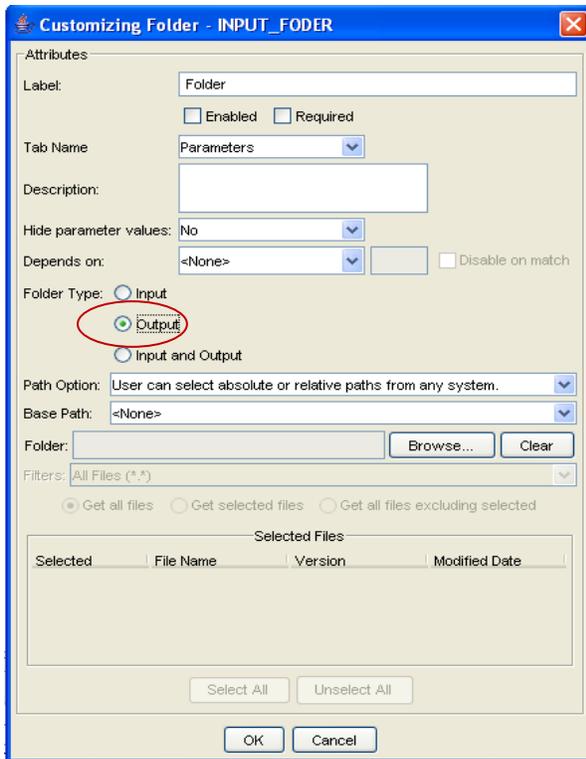
Since SDD is a web-based system, to execute programs on SDD requires sending and receiving information to the server. The more data and codes are involved, the longer it will take to execute a program. When executing a

program, eliminating useless data, macros and codes can improve SDD efficiency.

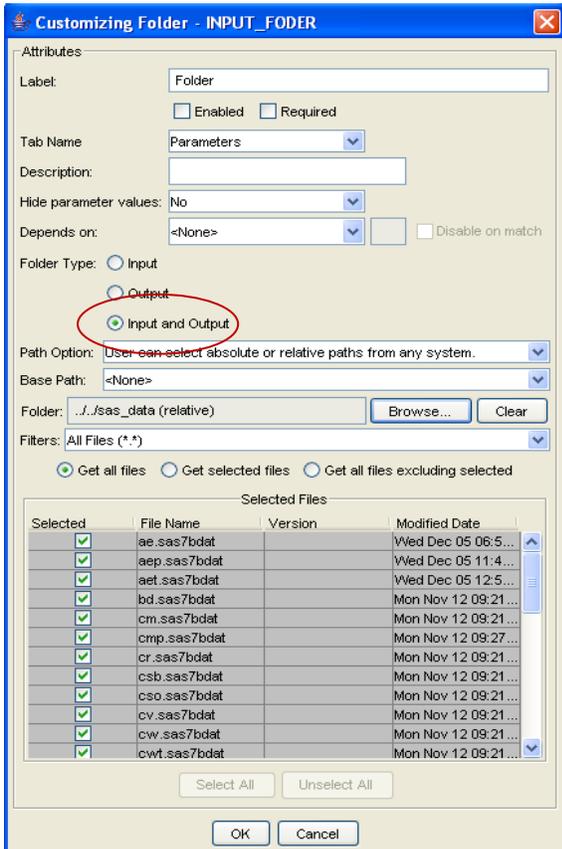
To define SAS data sets location in SDD parameter window, the location should be set as a folder. In SDD parameter window, there are three kinds of folder: "Input", "Output" and "Input and Output". The "Input" folder defines the location of a folder for data used in a program. One or more data sets from 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).



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



Display 3. Output folder



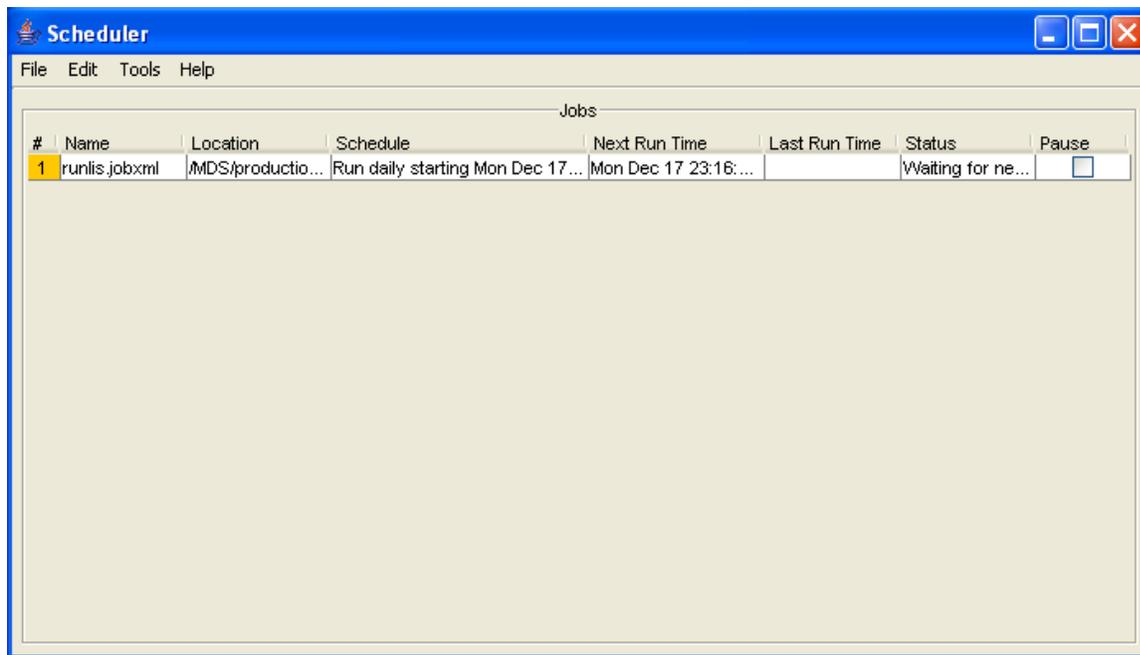
Display 4. Input and Output folder

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.



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

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.