Paper 10723-2016

SAS-IO, a Browser-Based Tool to Automate Creation of Inputs/Outputs Lists for SAS® Programs

Mohammad Reza Rezai, Institute for Clinical Evaluative Sciences, Ontario, Canada

ABSTRACT

High-quality documentation of SAS® code is standard practice in multi-user environments for smoother group collaborations. One of the documentation items that facilitate program sharing and retrospective review is a header section at the beginning of a SAS® program highlighting the main features of the program such as the program's name, creation date, program's aims, the programmer's identification and project title. In this header section, one of helpful components is to keep a list of the inputs and outputs of the SAS® program (e.g. SAS® datasets and files the program used and created).

This paper introduces SAS-IO, a browser based HTML/JavaScript tool that can automate production of such an input/output list. This can save the programmers' time especially when working with long SAS® programs.

INTRODUCTION

In multi-user collaborative programming environments, good documentation of SAS® programs is a must. High-quality documentation of SAS® programs facilitates program sharing and collaboration. Among helpful documentation components of a SAS® program is creating a header section at the beginning of the program highlighting the main features of the program. A program header usually has items like the program's file name, creation date, program's aims, the programmer's identification, project title and so on. In this header section, some organizations as well as individual programmers may find it helpful to maintain a list of the inputs/outputs of a SAS® program. These input/output items may include but are not limited to SAS datasets used/created, non-SAS files read/wrote, the macros called and the libraries defined within the SAS® program.

Creating such a list usually demands searching the SAS® code or the corresponding log output for certain SAS® syntax to extract the list items. This may be time-consuming especially with lengthy SAS® programs having hundreds of lines of code, and in high-demand, high-throughput programming environments. This paper introduces a browser-based tool that can automate production of such an input/output list saving programmers' time.

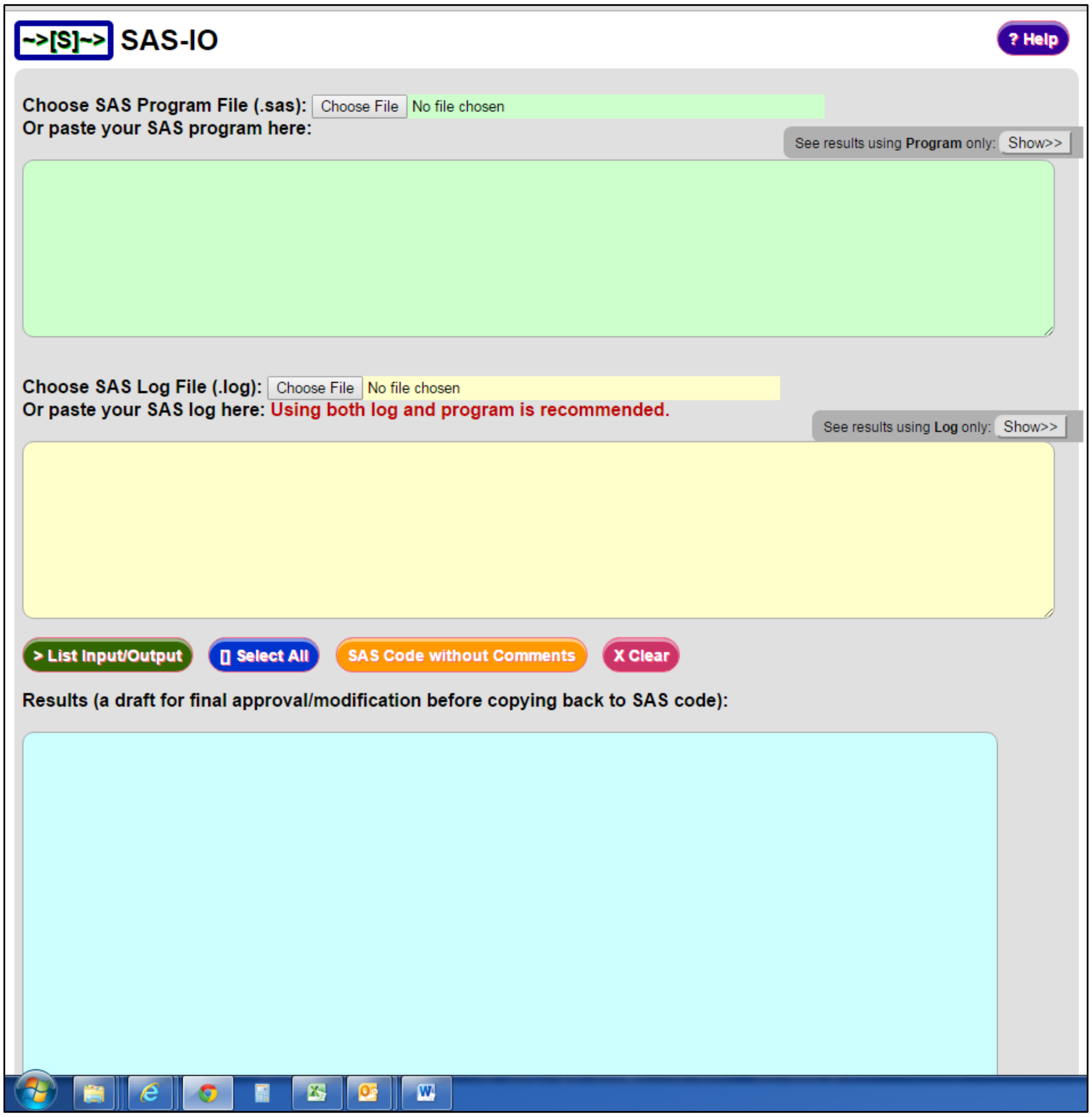
TOOL DESCRIPTION

SAS-IO was developed using HTML (Hypertext Markup Language), JavaScript, CSS (Cascading Style Sheets), and regular expressions programming languages. It is implemented in a single HTML page, needs no installation and can be simply run as a web page in most browsers supporting JavaScript and HTML5 (e.g. Microsoft Internet Explorer 10). No web server is needed as it can be run locally on the user's machine.

It is able to read a SAS® *Program* and/or its corresponding *Log* from the disk or process them as paste-in text. The result is a list of inputs/outputs appearing in a text box which the user can copy back to his/her SAS® program. The tool uses JavaScript and regular expressions to search for certain patterns in the text inputs. The algorithms in the tool first remove commented areas from the code to avoid listing the items mentioned in comments.

INTERFACE

The user interface is a web page. It gives the user the option to load a SAS® *Program* or *Log* (in text format) from the disk (using an HTML file object with a *Browse* or *Choose File* button) or alternatively pasting the SAS® *Program* (code) or related SAS® *Log* contents into text boxes on the user interface. The paste-in option is useful when SAS® *Program* or *Log* has not been saved as text files on the disk and are embedded, for instance, in SAS Enterprise Guide®. The SAS® *Program* text area has a green background and SAS® *Log* text area has a light-yellow background. The final results will turn up in the turquoise text area at the bottom **(Display 1)**. To aid the user, a built-in help section is provided via a **[Help]** button on top right. There is also tool-tip style help available.



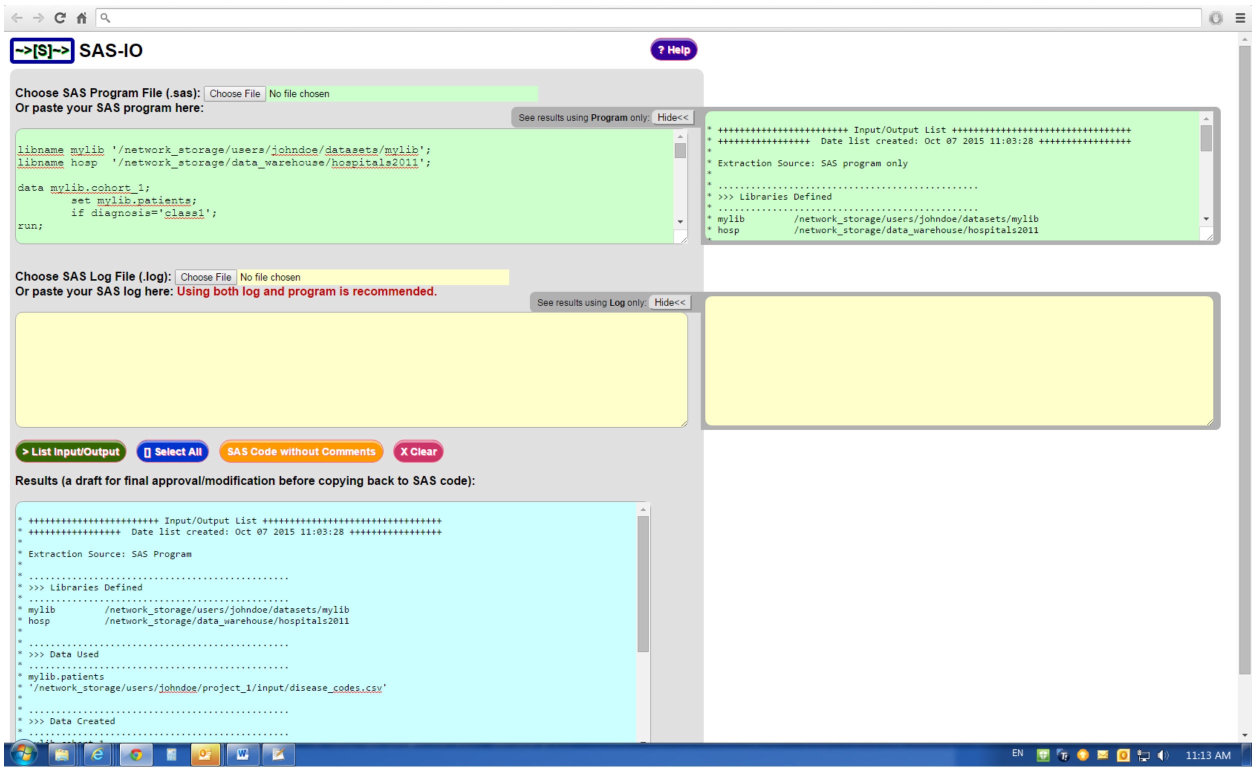
~>[S]~> SAS-IO
? Help
Choose SAS Program File (.sas): Choose File No file chosen
Or paste your SAS program here:
See results using Program only: Show>>
Choose SAS Log File (.log): Choose File No file chosen
Or paste your SAS log here: Using both log and program is recommended.
See results using Log only: Show>>
> List Input/Output
Select All
SAS Code without Comments
X Clear
Results (a draft for final approval/modification before copying back to SAS code):

Display 1. The screenshot of SAS-IO user interface.

The user can move the mouse over each part of the user interface to see informative tool-tips.

- A green **[List Input/Output]** button starts the processing of the SAS® *Program* and/or *Log* and shows the final input/output list in the text area below the button bar (turquoise background).

- A dark-blue **[Select All]** button helps the user to select all of the resulting text output and to copy it consequently.
- An orange **[SAS Code without Comments]** button opens a new window with the input SAS® code stripped of all the comments.
- A pink **[Clear]** button resets all of the form and prepares for the next task.
- Above each *Program* and *Log* text areas there is a [**Show>>**] button which reveals the (input-specific) inputs/outputs list using only *Program* or only *Log* as the input texts. These buttons work after *[List Input/Output]* button has been pressed and the text inputs have been processed.



Choose SAS Program File (.sas): No file chosen
Or paste your SAS program here:

See results using **Program** only:

```
libname mylib '/network_storage/users/johndoe/datasets/mylib';
libname hosp  '/network_storage/data_warehouse/hospitals2011';

data mylib.cohort_1;
        set mylib.patients;
        if diagnosis='class1';
run;
```

```
* ++++++++++++++++++++++++ Input/Output List ++++++++++++++++++++++++++++++++++
* +++++++++++++++++  Date list created: Oct 07 2015 11:03:28 +++++++++++++++++
*
* Extraction Source: SAS program only
*
* ................................................
* >>> Libraries Defined
* ................................................
* mylib         /network_storage/users/johndoe/datasets/mylib
* hosp          /network_storage/data_warehouse/hospitals2011
```

Choose SAS Log File (.log): No file chosen
Or paste your SAS log here: Using both log and program is recommended.

See results using **Log** only:

Results (a draft for final approval/modification before copying back to SAS code):

```
* ++++++++++++++++++++++++ Input/Output List ++++++++++++++++++++++++++++++++++
* +++++++++++++++++  Date list created: Oct 07 2015 11:03:28 +++++++++++++++++
*
* Extraction Source: SAS Program
*
* ................................................
* >>> Libraries Defined
* ................................................
* mylib         /network_storage/users/johndoe/datasets/mylib
* hosp          /network_storage/data_warehouse/hospitals2011
*
* ................................................
* >>> Data Used
* ................................................
* mylib.patients
* '/network_storage/users/johndoe/project_1/input/disease_codes.csv'
*
* ................................................
* >>> Data Created
* ................................................
```

Display 2. Screenshot of SAS-IO tool showing the inputs/outputs list from a SAS® Program only; note the input-specific hidden sub-panels are revealed.

CAPABILITIES

For the best results, it is recommended to use both SAS® *Program* and SAS® *Log*. In particular, when the programmer used macro variables to refer to dataset names or files, use of a SAS® *Log* is necessary to give desirable results. Otherwise, the tool can work with either SAS® *Log* or *Program* as text inputs.

The tool conducts some **preliminary checks** on the inputs when *[List Inputs/Outputs]* button is pressed. It first checks if appropriate text inputs (SAS® Program and /or Log) have been provided by the user. It also checks if the file extensions provided (.sas, .log or .txt) make sense and match their input destinations on the user interface. Further checks are performed to see if the SAS® Program and Log files (if provided) have the same names (disregarding their extensions). These are all to prevent accidental user errors like providing an unrelated log with a SAS® program. Conventionally the names of the SAS® program and its log are the same e.g. my_program.sas and my_program.log. The user may accidentally provide a program but a log output from a different program. If using the file input method reading files from the disk, SAS-IO tool can detect such errors. Also, it warns (using browser's built-in pop-up messages) if a user chose a log file in SAS® Program file input and vice versa.

Despite the warnings, the tool allows the user to proceed (by selecting OK in the pop-up messages) as some users may have saved their SAS® programs or logs with unconventional file extensions, etc. For instance a SAS® program may have been saved with .txt extension or no extension.

After the Program/Log inputs pass the preliminary checks, the tool produces 3 inputs/outputs lists: one from the SAS® *Program,* one from the SAS® *Log* text input, and a final output comparing the former lists. The final list loads in the turquoise text area at the bottom **(Display 1)**. The tool also puts the input-specific lists from SAS® Log and Program into two text areas beside the *Log* (light-yellow) and *Program* (green) text areas (hidden by default). If the user wishes to examine these input-specific lists, s/he can use corresponding *[Show>>]* buttons to show/hide these by-products too **(Display 2)**. A sample of input/output list is shown in **Output 1**.

```
* +++++++++++++++++++++++ Input/Output List ++++++++++++++++++++++++++++++++
* ++++++++++++++++  Date list created: Oct 06 2015 12:40:30 ++++++++++++++++
*
* Extraction Source: Both SAS Program and Log
*
* ................................................
* >>> Libraries Defined
* ................................................
* mylib     /network_storage/users/johndoe/datasets/mylib
* hosp      /network_storage/data_warehouse/hospitals2011
*
* ................................................
* >>> Data Used
* ................................................
* mylib.patients_2011
* hosp.hospitals_2011
* /network_storage/users/johndoe/project_1/input/disease_codes.csv
*
* ................................................
* >>> Data Created
* ................................................
* mylib.cohort_1--[n=34405--vars=37]
* mylib.cohort_2--[n=5315--vars=5]
*
* ................................................
* >>> Reports/Files/Outputs Produced
* ................................................
*
* /network_storage/users/johndoe/project_1/output/Report1.rtf
* /network_storage/users/johndoe/project_1/output/Table1.txt
* /network_storage/users/johndoe/project_1/output/Table2.txt
* /network_storage/users/johndoe/datasets/export2R.csv
*
* ................................................
* >>> Macros Called
* ................................................
* %MAKETABLE
* %MAKEREPORT
*
* ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
*
* ;
```

Output 1. Example output from SAS-IO tool by use of SAS® log and program as inputs to the tool.

The final input/output list contains the following items **(Output 1)**.

1. **Libraries Defined**: All libraries defined will be listed. This is useful to locate the listed SAS® datasets using the paths of their library names. This is performed by searching the pattern: libname [XXX] “[XXX]”

2. **Data Used:** Only datasets used from a non-WORK library are listed (e.g. `mylib.mydata1, temp.mydata3`).

   When using SAS® *Program* as input, the algorithms in the tool detect datasets used via *"set"* statement in data steps, and after *"from"* or *"inner/left/outer/full join"* statements in SQL procedure as well as *"datafile="* statement in *import procedure (proc import)*. The temporary datasets with no library prefix will not be listed. If a dataset has been both created and used (e.g. using *proc contents* or *proc means* for quality checks), it will not be listed in the data used list to avoid confusion and redundancy.

   In our organization, some large in-house data repositories are read via in-house macros. The tool also places such datasets in “Data used” list when it detects such data retrieval macros. For instance if a programmer uses %GETOHIP macro used to retrieve data from OHIP (Ontario Health Insurance Plan) data repository, the final “Data Used” list will contain the item “OHIP”.

   When using SAS® *Log* as input, the algorithms in the tool detect standard SAS® log syntax as the following example: ”`NOTE: There were [XXX] observations read from the data set [XXX].`”

3. **Data Created:** Only the SAS® datasets created into a non-WORK library e.g. `mylib.data1`, will be listed.

   When processing a SAS® *Program*, this tool detects *“data”* syntax in data steps, *"create table"* statement in *proc sql*, and *"out="* syntax in SAS® procedures and macros. If the programmer uses other statements (e.g. “outputFile=” in a macro), the tool cannot list it using *Program* only – i.e. providing the output *Log* is necessary.

   While processing a SAS® *Log*, the tool uses standard SAS® log text as in the following example: “`NOTE: The data set [XXX] has [XXX] observations and [XXX] variables.`” or from log text patterns relating SAS® *proc import* like this example: “`NOTE: [XXX] records were read from the infile [XXX]`”.

   The list of created SAS® datasets also reports the number of records and variables in each SAS® dataset only when using SAS® *Log* as input.

4. **Reports/Files/Outputs Created:** This mainly relates to the files created by the SAS® code (e.g. .rtf and .pdf files) using *ODS* (output delivery system) as well as exported non-SAS® datasets or outputs (e.g. .txt and .csv files).
   Using the SAS® Program as the input, all file paths after *"file="* (e.g. *ods rtf*) and *"outfile="* (e.g. in *proc export*) are listed.
   While processing the SAS® *Log* as input, the tool uses standard SAS® log text patterns associated with *ods html*, *ods rtf*, and *ods pdf.* For instance, the following pattern is searched for the files produced using *ods rtf*: “`NOTE: Writing RTF Body file: [XXX]`”

5. **Macros Called:** A list of the macros called is shown. The algorithm tries to avoid listing the SAS® macro functions (e.g. %sysfunc, %str), and common macro statements such as %macro, %mend, %include, %put and %let.

The algorithm in the tool performs final checks before outputting the final list. For example, it removes duplicate items from each category (e.g. when a dataset was used more than once). If one type of text input (i.e. only SAS® *Program* or only SAS® *Log)* is used, all items in the final list will be extracted from a single source. The source of extraction is also recorded at the top of the final output (**Output 1**).

In the cases where SAS® *Program* is the only input source for extraction of the list, and the programmer used macro variables (e.g. `data lib&i.dataset&j;`), the tool warns the user when detecting "&" character in the final output. The pop-up message recommends that the programmer would use the appropriate SAS® *Log* along with the SAS® *Program* containing macro variables.

If both SAS® Program and SAS® Log are used, the tool primarily extracts "Data Used" and "Data Created" lists as well as the "Reports/Files/Outputs" list from SAS® *Log* as it is more accurate - in particular, when the programmer used macro variables for such items. However, some sources of used data (e.g. in *proc sql*'s join statements) may only be extractable from *Program* and not *Log*. The "Macros Called" list is extracted from both *Log* and *Program* sources. Finally, in such cases, "Libraries Defined" list is only extracted from SAS® *Program* source.

BY-PRODUCT

A by-product of SAS-IO tool is a **Comment-Stripped SAS® Program.** The author is not clear how useful this feature is to SAS® programmers, but this capability is provided as a by-product since it had to be built in the tool anyway. The user can press *[SAS Code without Comments]* button **(Display 1)** to open a new window showing a text area containing a version of the input SAS® *Program* code with all comments stripped off.

LIMITATIONS

If only SAS® Program is used as the input, this tool cannot resolve macro generated dataset names or file outputs from a SAS® Program. For instance, if SAS® program contains the following code, the tool simply reports `mylib.newdata&i` as a dataset name in "Data Used" list.

```
data mydata;
  set mylib.newdata&i
run;
```

Fortunately, the tool warns the user in such cases in a pop-up message and recommends a *Log* output would be provided along with the SAS® Program.

The "Data Created" list reports the number of records and variables for SAS® datasets (**Output 1**) only when using SAS® *Log* as input. The number of records and variables are not reported in "Data Used" list because the user may use sub-setting statements (e.g. *where* or *if*) in data steps and such numbers will not be reliable. Also, some files are written and read in multiple steps and extracting the final or accurate numbers may be challenging. For accurate results, if a log file is to be used as the input, it should be the complete log after a successful run of the SAS program. Incomplete aborted log files will give incomplete information.

SAS-IO tool is still in testing phase and the final list of inputs/outputs should be considered a draft for review/approval by the programmer. SAS® is a comprehensive language and has numerous procedures. It is not unlikely that this tool might miss listing some of the items relating SAS® procedures the author is unfamiliar with. The author hopes to make SAS-IO a more robust tool by further testing and feedback by in-house users.

CONCLUSION

The main goal of SAS-IO tool is saving the programmers' time by creating a draft of inputs/outputs for programmer's final review - especially for very long SAS® codes. Programmers can use such a list to get a quick overview of new SAS® programs they are visiting for the first time without running it, or simply use the output lists to enhance their codes' documentation.

There is a lot of space for improving SAS-IO. One feature that can be added in future versions is allowing the users to provide a setting file to define their own user-specific SAS® syntax to be detected in the *Program* or *Log* input. The program can also be extended to contain more fields (e.g. programmer's and project's characteristics, etc.) to allow the programmer to create a full program header. These headers may vary across organizations.

Although this tool is implemented outside SAS® environment, there might be potential for it to turn into a SAS Enterprise Guide® plug-in using Microsoft .Net® platform. The browser-based implementation has the advantage that the tool works even without access to a SAS® installation. Similar capabilities may also be created via SAS® macros and use of regular expressions. However, the design and the user interface offered by this tool or a similar integrated SAS® plug-in would make it more user-friendly.

CONTACT INFORMATION

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

M. Reza Rezai
Institute for Clinical Evaluative Sciences
G1 06, 2075 Bayview Avenue
Toronto, Ontario M4N 3M5
Canada
Email: reza.rezai [at] ices.on.ca
LinkedIn: https://ca.linkedin.com/pub/mohammad-reza-rezai/2a/13a/637