
Analyzing your SAS log with user defined rules using an app or macro.

And the benefits this provides

Biography

Philip Mason

Philip Mason has been a SAS programmer for almost 40 years, having spent 30 years as an independent SAS Consultant. Starting in Australia he then moved on to Europe working in Belgium, Netherlands and the U.K. where he is now based. He has written 3 books about SAS, and also written several SAS courses, founded some user groups, helped run PharmaSUG, SAS Global Forum, VIEWS, SMUG, and so on.

He currently works with SAS and JavaScript, helping a bio-tech called **argenx** in Ghent, Belgium.



Agenda



Approach



Defining Rules



Using rules with a macro



Using rules with web application



Futures

Approach

What do we want from log analysis

What do we want?



Give me more than SAS does



Fast



Consistent source of truth



Analyze and find what I want



Run in batch or use interactively



Integrate with linking to it and from it

How do we get it?

- **Define each thing to look for**
- **Implement code to use those rules to analyze log**
 - Do this in SAS
 - Do this in JavaScript
- **Build a web application to provide interactive interface to results**

Defining Rules

A way to represent what to look for in a log



Simple rule to find Notes

"startswith": "NOTE:",

"prefix": "",

"suffix": "",

"anchor": false,

"linkColor": "blue",

"type": "NOTE",

Rule properties

Property	Description
ruleType	- “regex” indicates this rule uses a regular expression to search text - “startsWith” indicates this rule will simply check if a line starts with some text
startswith	Text used in searching line to see if that line starts with this
regex	Regular expression used to search line. Basic use is simply to see if a line matches. You can use capture groups which can be used in some more advanced instances.
prefix	Text (usually HTML code) that will be added to the beginning of line before it is displayed. Usually, you can use a tag with some formatting in style= to change colours, fonts, etc.
suffix	Text (usually HTML code) that will be added to the end of line before it is displayed.
anchor	- True will display line as a clickable link in the “contents” area of log viewer. - False won't display a link, but in the HTML of the log displayed there will be an id= tag with a unique reference, so that the line can be linked to from other functionality such as the input/output tables.
linkColor	Defines an HTML colour to use for links in the “contents” area

Rule properties

Property	Description
type	Defines text to categorise rule as, e.g. “ERROR”. You can create any categories, and checkboxes will appear for each category along with a count of how many lines match. null can be used as a value, which will not produce any category checkbox.
interesting	- true means that you can use the next/previous interesting thing arrows in the viewer for this line, allowing grouping of the things you are most interested in - false means this is not interesting
description	Description of this rule, useful as documentation and to help others especially if you are using more complex regular expressions
substitute	- false does nothing - true will do the following: 1 - use the regular expression to capture the text that matches 2 – replace {{matched}} in prefix and suffix with the matching text e.g. find URLs in the log and convert them into clickable links to those URLs

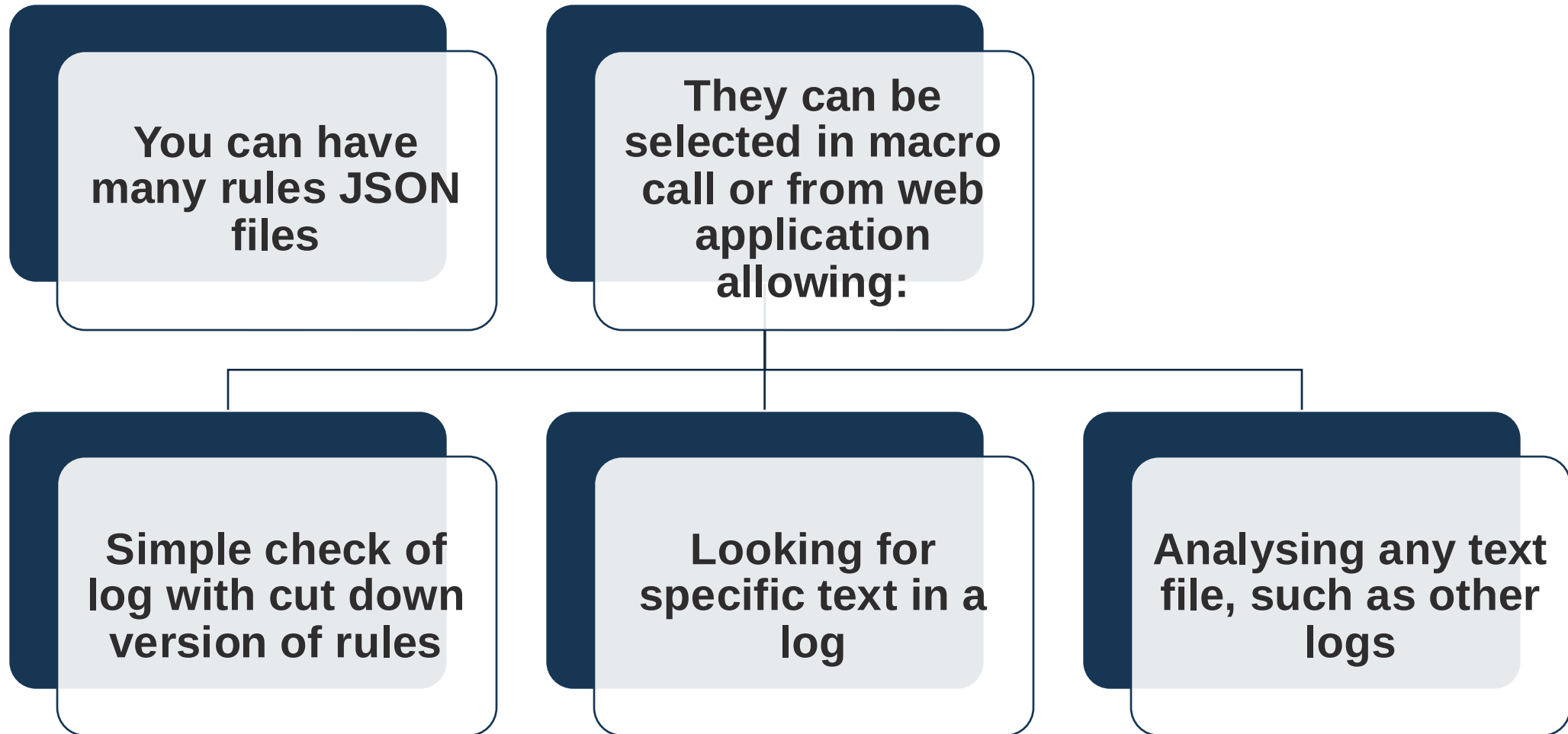
A rule to find Warnings

```
"ruleType": "regex",  
  "regex": "(?:^WARNING:|^WARNING\\s)(?!.*PRODUCT WITH  
WHICH|. *IS ASSOCIATED WILL  
EXPIRE|. *MPRINT|. *MLOGIC|. *SYMBOLGEN|. *IS ASSOCIATED IS  
SCHEDULED TO EXPIRE|. *YOUR SYSTEM IS SCHEDULED TO  
EXPIRE)",  
  "prefix": "<span style='color: green'>",  
  "suffix": "</span>",  
  "anchor": true,  
  "linkColor": "green",  
  "type": "WARN",
```

More complex rule making links to URLs

```
{  
  "description": "Create link to paths to a file or directory starting with http.",  
  "ruleType":    "regex",  
  "regex":       "http[s]?\\:([\\w+|\\-|.]*",  
  "prefix":      "<a href='{{matched}}' target='_blank'>",  
  "suffix":      "{{matched}}</a>",  
  "anchor":      false,  
  "linkColor":   "blue",  
  "type":        "FILE",  
  "interesting": false,  
  "substitute":  true  
}
```

Defining special rules files



Using Rules with a macro

Analyze log with a macro call to produce a report

Requirements for macro

What rules and log to use?

Types of rules to return?

e.g. just Errors? Warnings?

Produce a dataset with results

Macro variable with result of scan

e.g. Pass or Fail

fileCheckWithRules parameters

Parameter	Description
log	location of text file (log) that will be analysed. If you specify pcsas then it will grab the log if running on PC SAS
rules	location of JSON file containing rule definitions. By default uses file from logviewer directory.
returnTypes	List of pipe separated quoted types to return, e.g. ERROR WARN If blank, then everything found is returned.
print	- 0, produces no report, but does set macro variables and creates a dataset - 1, also produces a report
outputName	Optionally override the name of dataset that is created

What is returned by fileCheckWithRules

- Dataset is created containing results - fileCheckWithRules
- Macro variable returned with the number of rows in dataset
- Allows checking in a program if anything was found
- HTML report is generated

4 lines found in ___filecheckwithrulestemp.log

Line Number	Type	Line	Interesting
16	WARN	WARNING 1-322: Assuming the symbol SET was misspelled as sett.	
26	ERROR	ERROR: Libref SASHEL is not assigned.	Y
28	SERIOUS	NOTE: The SAS System stopped processing this step because of errors.	Y
29	WARN	WARNING: The data set WORK.ERR may be incomplete. When this step was stopped there were 0 observations and 0 variables.	

PC SAS example 1

```
dm 'log;clear';

data ok; set sashelp.class ; run ;
data warn ; sett sashelp.class ; run ;
data err ; set sashel.class ; run;

%filecheckwithrules(pcsas,rules=C:\github\logviewer\src\rules.json,print=1)
```

4 lines found in ___filecheckwithrulestemp.log

Line Number	Type	Line	Interesting
16	WARN	WARNING 1-322: Assuming the symbol SET was misspelled as sett.	
26	ERROR	ERROR: Libref SASHEL is not assigned.	Y
28	SERIOUS	NOTE: The SAS System stopped processing this step because of errors.	Y
29	WARN	WARNING: The data set WORK.ERR may be incomplete. When this step was stopped there were 0 observations and 0 variables.	

PC SAS example 2

```
%filecheckwithrules(C:\Users\pmason\SAS\logs\pp.log,  
    rules=C:\github\logviewer\src\rules.json,  
    print=1)
```

1 lines found in C:\Users\pmason\OneDrive - argenx BVBA\Documents\SAS\logs\pp.log

Line Number	Type	Line	Interesting
2489	ERROR	NOTE: Variable path is uninitialized.	Y

LSAF example

Code	 Log	Results	Submission Inputs	Submission Outputs
1	proc printto log="&_sasws_/temp.log" new ; run;			
2	data ok; set sashelp.class ; run ;			
3	data warn ; sett sashelp.class ; run ;			
4	data err ; set sashel.class ; run;			
5	proc printto ; run;			
6	%include "&_SASWS_/general/biostat/macros/_library/filecheckwithrules.sas";			
7	%filecheckwithrules(&_sasws_/temp.log,print=1)			

2 lines found in /lsafshared/SASWorkspaces/pmason/temp.log

Line Number	Type	Line	Interesting
29	ERROR	ERROR: Libref SASHEL is not assigned.	Y
34	WARN	WARNING: The data set WORK.ERR may be incomplete. When this step was stopped there were 0 observations and 0 variables.	Y

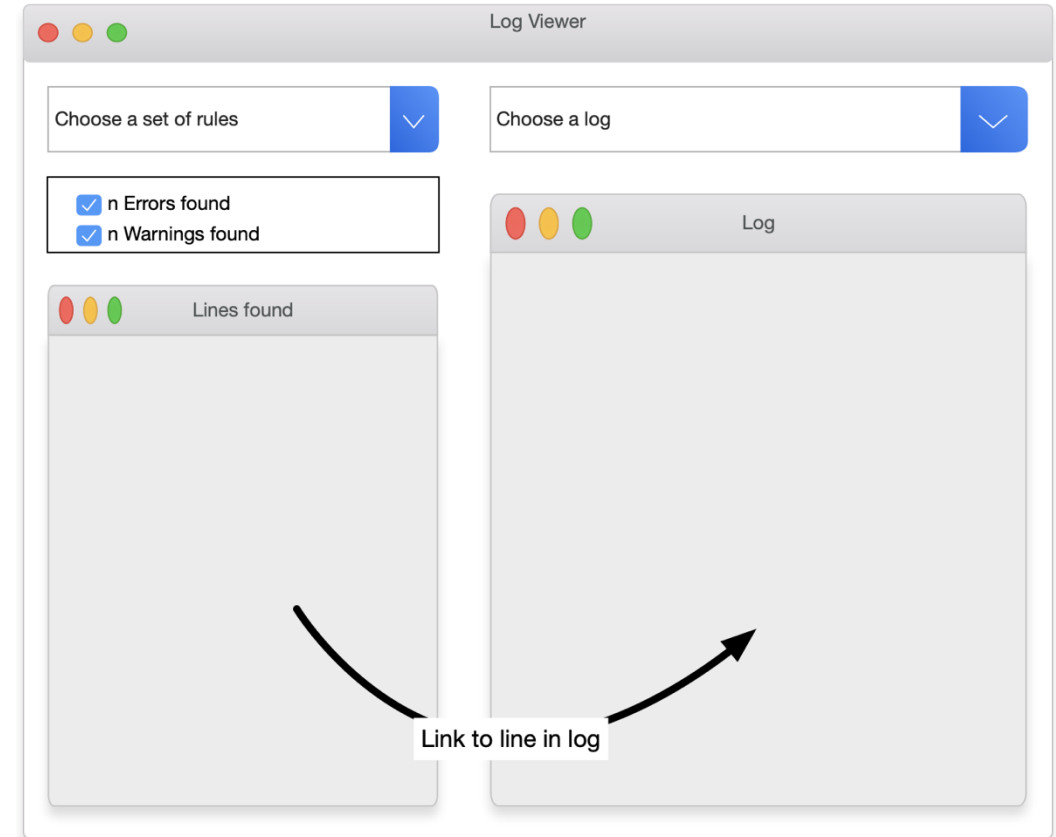
Using Rules with Web Application

Log Viewer Web Application

Requirements

- **Select rules and log to analyze**
- **Display log with styles applied**
- **Summarize results of analysis**
- **Jump from summary to related point in log**
- **Nice to haves**
 - Performance info
 - Macros used
 - Turn off mprint lines (reduce clutter)

Wireframe



Log Viewer Web app

Log Viewer

localhost:3000/sites/Biostatistics/Shared%2520Documents/Log%2520Viewer

Web Dev Programming SAS Business Radio Home Fun WSC Pharnasug CDISC WIKI Dataset-JSON v1.1

Directory containing logs
/Users/philipmason/github/logviewer/tests

READ

Log Name

Program: dashboard2.sas Submitted: Apr 11, 2023 02:36:04 GMT by mbusselen Ended: Apr 11, 2023 03:03:00 GMT Lines: 47,169

e (2.4mb).log

☒ ERROR ☒ WARN ☒ SERIOUS ☒ NOTICE ☒ OTHER ☒ JOB ☒ INFO

WARNING: Apparent symbolic reference ALLRESDB not resolved.

ERROR: The text expression &ALLRESDB contains a recursive reference to the macro variable ALLRESDB. The macro variable will be

ERROR: BY variable reporting_event_path is not on input data set WORK.PGMSUMM.

ERROR: BY variable reporting_event_path is not on input data set WORK.OUTPUTCATSUMM.

WARNING: The data set JOBOUT.METAPLUS may be incomplete. When this step was stopped there were 0 observations and 26 variables.

WARNING: Data set JOBOUT.METAPLUS was not replaced because this step was stopped.

ERROR: Variable RETYPE not found.

ERROR: Variable RETYPE not found.

ERROR: Variable RETYPE not found.

WARNING: The data set WORK.ALLSUMM may be incomplete. When this step was stopped there were 0 observations and 0 variables.

ERROR: Variable RETYPE not found.

ERROR: Variable retype is not on file WORK.ALLSUMM.

WARNING: The data set WORK.ALLSUMTOT may be incomplete. When this step was stopped there were 0 observations and 6 variables.

ERROR: The variable type of 'CREATEDPGM.SUM'n is invalid in this context.

ERROR: The variable type of 'CREATEDPGM.SUM'n is invalid in this context.

ERROR: The variable type of 'EXPECTEDPGM.SUM'n is invalid in this context.

ERROR: The variable type of 'ISSUEPGM.SUM'n is invalid in this context.

ERROR: The variable type of 'ISSUEPGM.SUM'n is invalid in this context.

ERROR: The variable type of 'CREATEDOUTPUT.SUM'n is invalid in this context.

ERROR: The variable type of 'CREATEDOUTPUT.SUM'n is invalid in this context.

ERROR: The variable type of 'EXPECTEDOUTPUT.SUM'n is invalid in this context.

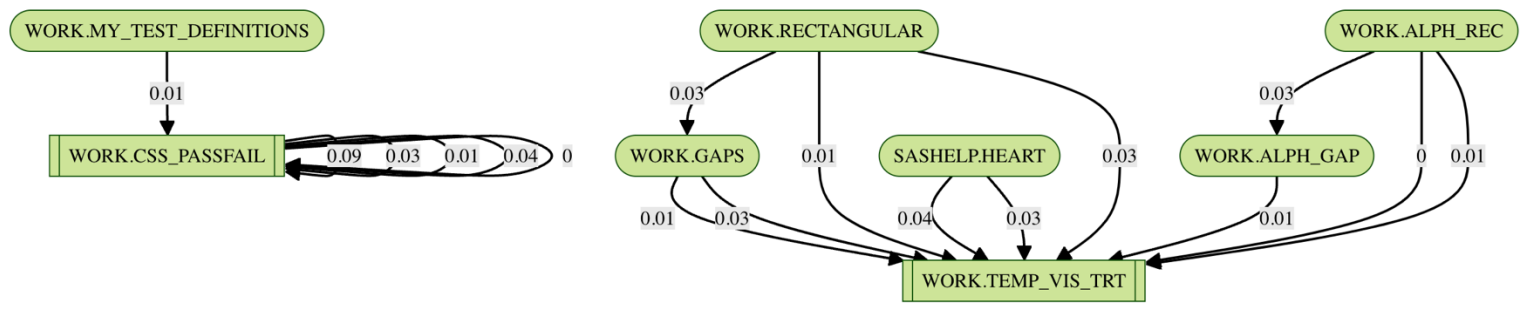
OUTPUTS	INPUTS	FILES	REAL TIME	CPU TIME	MPRINT	MLOGIC	SYMBOLGEN
Line	Libname	dataset	# obs	# variables			
168	WORK	GEN_SETENV	2	3			
317	WORK	REPEV_SETENV	0	3			
687	WORK	LSAFGETPROPERTIES	21	7			
728	WORK	META	383	5			
740	WORK	MFTA	383	5			

```
96 +macro setSasAutos();
97 + %local firstset last3 nb_dirs user_workspace_macros;
98 + %let firstset = 1;
99 + %if %symexist(runenvironment)=0 %then %local runenvironment;
100 +
101 + **-- Determine if WORKSPACE macros should be used (in case available) --*;
102 + %if %symexist(USE_SAS_WORKSPACE_MACROS)=0 %then %do;
103 + %global USE_SAS_WORKSPACE_MACROS;
104 + %end;
105 + %if %sysfunc(sysexist(USE_SAS_WORKSPACE_MACROS)) %then %do;
106 + %let USE_SAS_WORKSPACE_MACROS = %sysget(USE_SAS_WORKSPACE_MACROS);
107 + %end;
108 + %if %sysfunc(sysexist(USE_SAS_WORKSPACE_MACROS))=0 and %qsysfunc(upcase(%superq(USE_SAS_WORKSPACE_MACROS))), $1.)=Y
109 + !%then %do;
110 + **-- Set environment variable --*;
111 + %option set=USE_SAS_WORKSPACE_MACROS "Y"; /*- to unset environment variable use: option
112 + !set=USE_SAS_WORKSPACE_MACROS ""; --*/
113 + %end;
114 + %put 6=USE_SAS_WORKSPACE_MACROS;
115 + %put %str(Not)ice: Environment variable USE_SAS_WORKSPACE_MACROS is
116 + !%sysfunc(ifc(%sysfunc(sysexist(USE_SAS_WORKSPACE_MACROS)), DEFINED, NOT DEFINED)).;
117 +
118 + /*in LSAF there are 3 default directories in sasautos and these need to be kept*/
119 + %if &runenvironment=repository or &runenvironment=workspace %then %do;
120 + data _null_;
121 + length a b c dirs l3 $4000; /*- jbdart - 2022-12-08 - increase l3 size to avoid truncations --*/
122 + a = strip(getoption("sasautos"));
123 + put a=;
124 + len = length(a);
125 + nobrack = substr(a,2,len-2);
126 + b = tranwrd(nobrack,',','');
127 + c = compl(strip(tranwrd(b,',','')));
128 + call symputx("nb_dirs",strip(put(e,8.)));
129 + do i = 1 to e;
130 + l3cat3 = (a-i<2);
```

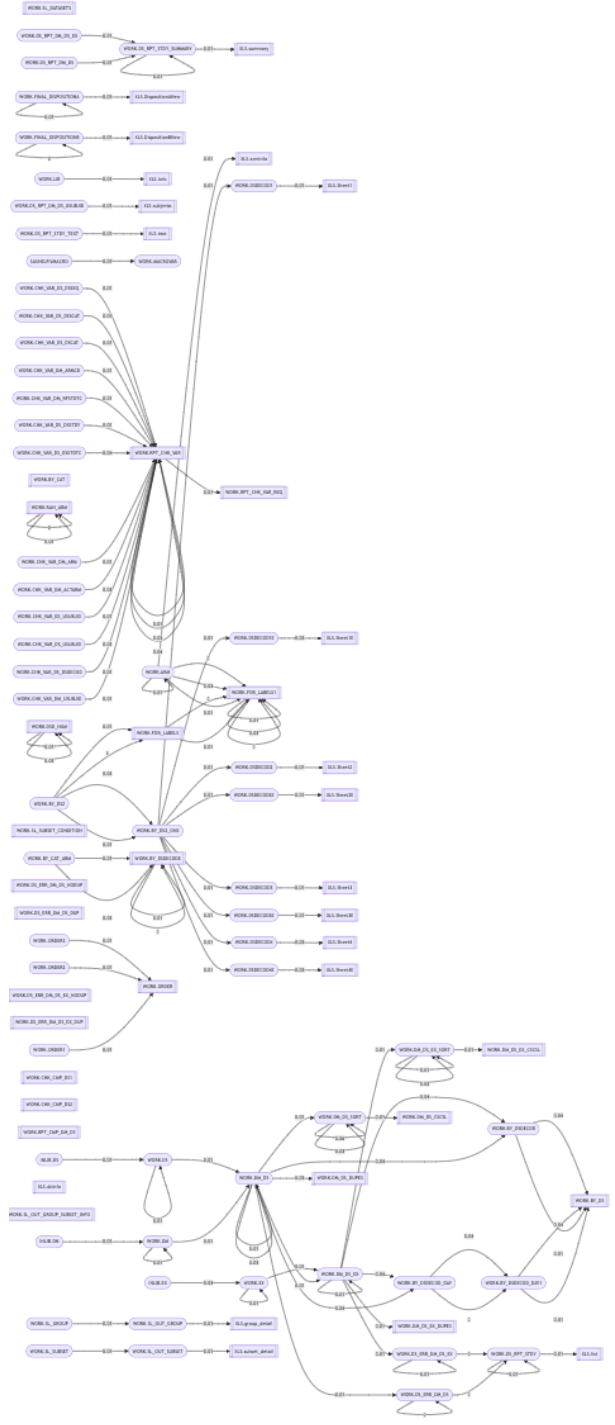
- Resize panels
- Resize fonts
- Show/Hide
 - Source code
 - Macro lines
 - Line numbers
- Data Flow Diagram
- Choose/View rules
- Copy source code
- Paste a log
- Email about this log
- Refresh
- Resize manually
- Next/Previous
 - Page
 - Interesting thing
 - Error
 - Warning
- Search, find next/previous
- Info & Documentation

Example Data Flow Diagrams

🔍 ↺ 🔍 CLOSE



1,775 characters 27 lines ✖



Futures

See me for a demo

Many enhancements are on the way

- Bookmarklet to automate from LSAF
- Tool definition to automate from PC SAS
- Bookmarklet to automate from SAS Studio
- Electron version, running natively on Windows, Mac and Linux
- VS Code extension

Name: Philip Mason
Affiliation: Wood Street Consultants Ltd.
Contact Number: (ask me)
E-mail: phil@woodstreet.org.uk or pmason@argenx.com
Website: <https://github.com/philipmason/Wood-Street-Consultants>

LinkedIn: <https://www.linkedin.com/in/philmason/>

