**PhUSE US Connect 2018**

**Paper CT15**

# ANALOG, an Interactive, Browser-Based and Business-Oriented SAS Log Checker

M. Reza Rezai, Hoffmann-La Roche, Mississauga, Canada

## ABSTRACT

SAS programmers have to interpret SAS logs every day. Multi-thousand-line SAS log files are not uncommon. SAS Base Software still lacks interactive tools providing a visual summary of SAS logs assisting in quicker and more insightful debugging. This paper introduces ANALOG, an interactive, browser-based tool (developed on HTML/JavaScript/CSS and regular expressions) that helps programmers interpret SAS logs more efficiently. Needless of a SAS installation, the tool summarizes SAS log in a colorized and succinct manner. It allows interactive toggling of log message categories, and click-to-see relevant log lines for each message. ANALOG also has advantages to SAS Enterprise Guide's log checking tool; it distinguishes significant NOTES which are NOTEs having more importance (e.g. uninitialized variable). The program also creates a list of inputs/outputs of the SAS program. ANALOG incorporates industry-specific log messages too. It is a great example of innovative use of web technology in everyday analytics.

## INTRODUCTION

SAS system is a widely used platform in different disciplines including the industry. A SAS program may run in different configurations, operating systems and via different user interfaces (GUI). Some well-known SAS user interfaces are *SAS Enterprise Guide*®, *SAS Studio*® (i.e. browser based) and *SAS Display Manager*® (Windowing Environment) which is probably the most widely used SAS GUI.

In contrast to these GUI environments, some programmers may use SAS in a pure command-line environment e.g. via a UNIX shell. There are also situations where proprietary software and their GUIs are the tier via which SAS programmers interact with SAS System's components like Base SAS® or other SAS add-on components.

Based on what programming setting and SAS configuration, SAS program logs may become available via SAS GUI windows (e.g. a log window in SAS Studio®) or be routinely saved in a log file in command-line settings.

The SAS log checking tools available in different SAS environments also vary. Whereas SAS Enterprise Guide® and SAS Studio® offer relatively interactive log checking and log summarizing windows, SAS Base and its Windowing Environment (Display Manager) offer almost none.

Most log checking tools (e.g. macros or system specific scripts) usually briefly report the number of errors and warnings after a program is run. While the errors and warnings are crucial parts of a SAS log, they may not tell the whole story that happened when a program ran. Important information about program and data issues may be ignored if only error and warnings are considered in a SAS log. In addition, such log checking macro/scripts usually used with SAS Base® are not interactive.

A proportion of statistical programming activities in industry are implemented via suites of SAS macros that provide a standard framework for generation of industry-specific generic analyses. In this paper, such macro suites are referred to as **Industry Specific SAS Macros (ISM)**. One feature of ISM is that they can have their own log messaging syntax. These custom **industry-specific SAS log messages** inform the user about the issues encountered by or information related to ISM macros. These messages may come in three major categories as the stereotypes below:

1) CRITICAL: A critical error happened, almost equivalent to 'ERROR' in SAS logs.
2) ALERT: An important issue merits programmer's attention; almost equivalent to 'WARNING' in SAS logs.
3) COMMENT: Usually provides extra information about ISM's options/automatic decisions etc., almost equivalent of 'NOTE' in SAS logs.

These ISM log messages may be written in the same SAS log as the program where such macros are used. Therefore, in a typical SAS log, the statistical programmer may face two classes of log messages: conventional SAS log messages and ISM log messages.

This tool was developed to provide an interactive log checking tool, providing enhanced and more efficient log summarization and visualization of both standard SAS and industry-specific log messages in one place taking the programmer's perspective beyond just errors and warnings. Although this tool demonstrates integration of ISM log

messages, it can be used for analysis of any SAS log. The tool was named ANALOG: a tool for **ANAL**ysis of SAS **LOG**s.

## DESCRIPTION OF THE TOOL

#### IMPLEMENTATION AND TECHNOLOGY

Although aimed to be a SAS log checking tool, ANALOG was developed based on web technology. There are already a number of SAS macros capable of log checking [3, 4]. Web technology has some advantages to SAS macros. First, a web interface (i.e. a web page) provides a more interactive and user friendly graphical user interface than a SAS macro. Second, a web-based tool runs independent of SAS installation; the user is not limited to having SAS System installed to be able to check a log (as in SAS macros). Last but not the least, these days, web browsers are available everywhere and provide a sophisticated platform for applications to run needless of requesting the user to install new software on their systems.

ANALOG was developed based on client-side web technology using HTML5, JavaScript, and Cascading Style Sheets (CSS). “Client-side” here means this tool can simply run independent of a web server and internet connection as a single HTML file residing on the user’s local drive – although it can be easily hosted on a web server and works fine online too. Regular Expressions (regex) language was used to detect text patterns for conventional SAS log and industry specific (i.e. ISM) log messages.

#### FEATURES AND CAPABILITIES

The user can input (read in) SAS log in two ways (**Figure 1**):

1) Opening a SAS log file, copying the log content and **pasting** in the text box provided in the tool. Using keyboard shortcuts (Ctrl+V) makes this even easier. At pasting, the tool automatically takes action and summarizes the log. Before pasting a new log content from clipboard, the user can reset the tool by pressing Ctrl+R (e.g. in Chrome, or Internet Explorer). Therefore, after the user comes to the browser window with ANALOG tool opened, a combination of Ctrl+R followed by Ctrl+V does the job in Chrome and Internet Explorer.
2) Using the ***Choose File*** button (in Chrome) or *Browse* button (as appears in Internet Explorer), to find the log file on a local or network drive. The tool automatically summarizes the log without pressing a “Go” or “Submit” button. After using this “Choose File” method, if the user needs to re-analyze the same log recently analyzed at the same file address (e.g. after re-running the SAS program and regenerating the log), there is no need to browse to the log file location again. Pressing **Re-Check Last Log** button re-reads the last log and analyzes it.

After the SAS log is analyzed, a summary appears in the teal-colored text area below the input text area (**Figure 1**). ANALOG summarizes log messages in 7 classes with each class colored in an intuitive way.

A. SAS log messages:
    a. **Error** (red): SAS log ERRORs.
    b. **Warning** (orange): SAS log WARNINGs.
    c. **Significant Note** (pink): The SAS log NOTEs considered of higher importance e.g. NOTE messages about “uninitialized variables” or “no observation in a dataset” [4].
    d. **Note** (teal): Less significant SAS NOTEs (i.e. not in “Significant Note” category above).

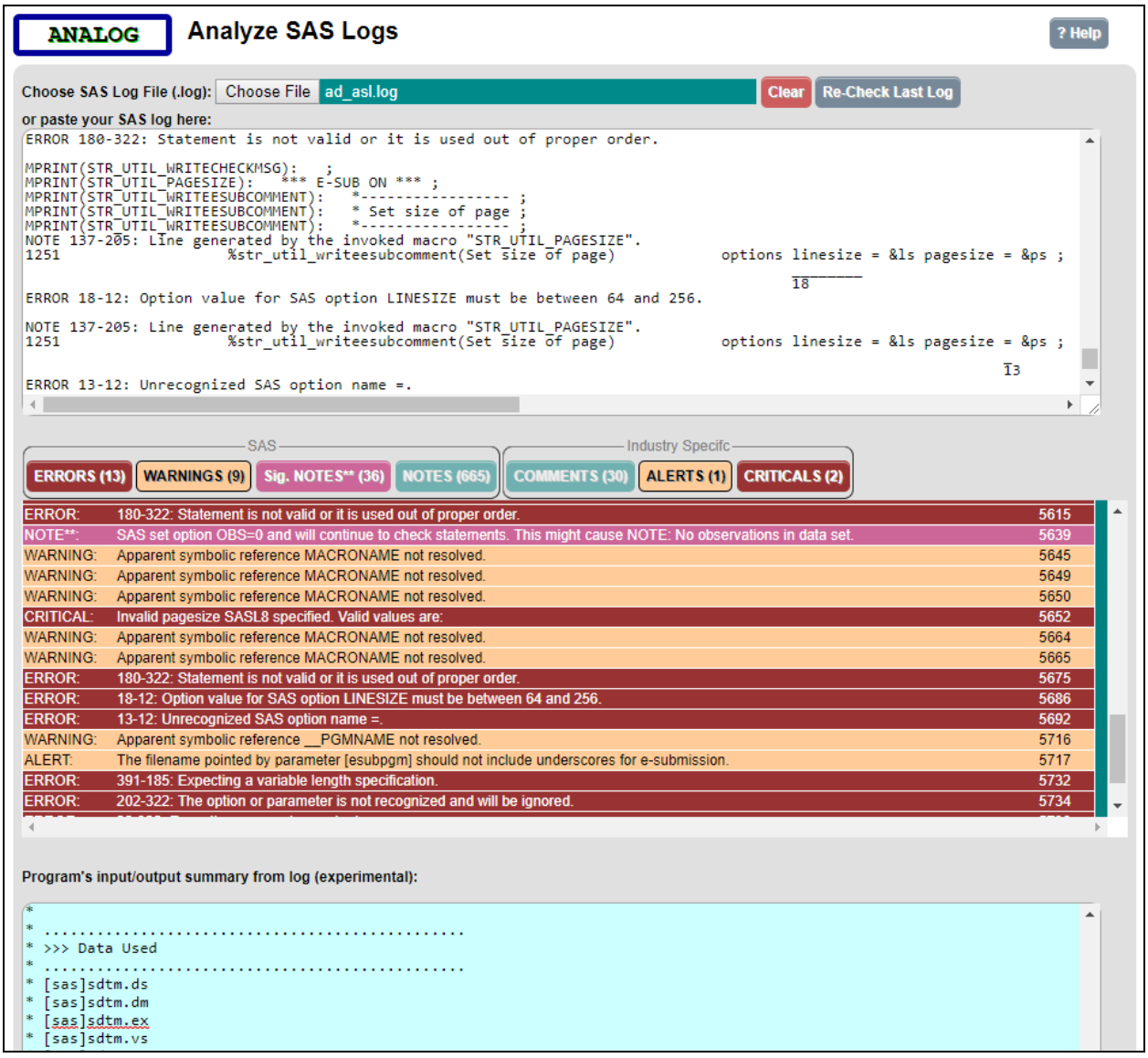
B. ISM (industry-specific) log messages (see above for details):
    a. **Critical** (red)
    b. **Alert** (orange)
    c. **Comment** (teal)

The **colors** of similar categories in SAS log messages and ISM log messages are the same. For instance, the SAS “Warning” and its ISM counterpart, “Alert”, messages have an orange background adding to intuitiveness of the log’s visualization.

One important feature of the tool is that, by default, it initially only shows the most important messages limiting to Errors/Criticals, Warnings/Alerts, and “Significant Notes”. It intentionally hides the SAS Notes and ISM Comments category. This saves the programmer’s time by providing a more relevant summary. However, the user can **interactively toggle** (show/hide) each log message category by pressing on the corresponding buttons. This provides a flexible way to show only the message categories of interest. Each button also shows the number of the messages found in each category. This feature resembles the log checker in *SAS Enterprise Guide*®. When a button

is toggled on to active mode showing the corresponding class of log messages the button and its text label looks sharper, and when toggled off to hide its corresponding log categories, the button and its text label becomes dimmer.

**Figure 1.** ANALOG's user interface. This is the user interface loaded as a web page in Chrome web browser.




A useful **interactive** feature of ANALOG is the user can click each colored message line in the report box to interactively scroll to relevant log area in the input log text area above. This saves user's time by providing a quick log browsing method. This feature works only if the user has not changed the viewing text size of the browser.

Another function implemented in ANALOG is **summarizing inputs to and outputs** from a SAS program. This feature is actually an extension of author's previous work [5] and is still in experimental stage.

This summary appears at the third text area on the user interface (cyan background) (**Figure 1**). Imagine the scenario that a programmer has a SAS log and wishes to have a summary of what happened during the run of that program – even without having SAS installed on their systems. This input/output list includes: libraries defined, datasets used/created in non-WORK libraries, macros and SAS procedures used in, and files created by the SAS program. Such a list can be useful, for different purposes:

1. Simply having a summary of inputs and outputs to check how things went during a program run.
2. Using this list to enrich a SAS program's documentation e.g. at program's header.
3. Reviewing the logs of a SAS program the programmer is visiting for the first time (e.g. shared in a collaboration) or getting a refresher summary when revisiting a SAS program one wrote a long time ago.

### PROS AND CONS

ANALOG has advantages compared with the log checkers in *SAS Studio*® and *SAS Enterprise Guide*®. ANALOG can summarize industry-specific (ISM) log messages. It can also distinguish significant NOTEs (e.g. uninitialized variable) from other NOTEs. The NOTE log messages are usually the most numerous SAS log messages. Although a NOTE message - as a non-critical message - may be ignored by most programmers, certain NOTEs may indicate a potentially serious issue in the code [4]. ANALOG helps the programmer by distinguishing more "significant" NOTE messages and only showing them along with ERRORs and WARNINGs at its initial log summary. This saves time considering that reviewing numerous SAS NOTE messages to find more significant ones can be time consuming. Providing the messages with message-specific intuitive colors is another advantage. Moreover, compared to *SAS Studio*®, ANALOG lists the log in a chronological order which is more natural and intuitive to follow. For instance, in ANALOG's log summary, one can find out a certain ERROR message happened after a certain NOTE message whereas in SAS Studio® the log message classes are provided separately.

Another advantage is ANALOG does not need a SAS installation to work, enabling log reviews even without having SAS system installed. As a web application, it is possible to open many windows of the tool in a browser and so comparing SAS log summaries in side-by side windows from different programs or different runs of the same program (i.e. loading two browser windows with ANALOG and setting them side by side).

As a disadvantage, one could argue that ANALOG is not integrated within SAS System, and a copy/paste or browse-to-file step is necessary to get a log input into the tool. Moreover, ANALOG works better in settings where a final SAS log file is ready after a complete run of a SAS program (i.e. command line settings) in contrast to programming in an interactive SAS session. It is still possible to copy log from an interactive session's log window. Finally, ANALOG at its current status cannot analyze many SAS logs at the same time in an automatic manner (e.g. analyzing all SAS logs in a directory). However, it has the potential to be extended to do so i.e. far more features can be added by use of server-based languages like PHP.

## CONCLUSION

In brief, ANALOG is an easy to use tool for summarizing/visualizing SAS logs needless of a SAS installation. It can save programmers' time by proving most relevant information by default in an interactive way and highlighting important points in the log including industry-specific log messages.

## REFERENCES


1. http://www.dummies.com/software/other-software/sas-for-dummies-cheat-sheet/
2. https://blogs.sas.com/content/sastraining/2017/04/12/whats-your-sas-interface/
3. Log Checks Made Easy. Pande Y. SAS Global Forum, 2012
4. SAS® Log Summarizer – Finding What's Most Important in the SAS® Log. Stojanovic M, SESUG, 2008
5. SAS-IO, a Browser-Based Tool to Automate Creation of Inputs/Outputs Lists for SAS® Programs. Rezai MR. SAS Global Forum 2016.


## CONTACT INFORMATION

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


M. Reza Rezai,
Hoffman La-Roche, Canada
Email: reza.rezai.rr1[at]roche.com
Web: https://ca.linkedin.com/in/mrezarezai