

Uncover the Most Common SAS® Stored Process Errors

Tricia Aanderud, And Data Inc

ABSTRACT

You don't have to be with the CIA to discover why your SAS® stored process is producing clandestine results. In this talk, you will learn how to use prompts to get the results you want, work with the metadata to ensure correct results, and even pick up simple coding tricks to improve performance. You will walk away with a new decoder ring that allows you to discover the secrets of the SAS logs!

INTRODUCTION

There are few things more ego deflating than running your beautiful stored process only to see it fail. Either the stored process complains it cannot complete the process or even worse – no results at all! If you are used to writing SAS programs that only you run, it is a little easier to debug issues or instinctively know how to avoid them. If you have never written a stored process, refer to *Creating Your First Stored Process* in the Reference section.

This paper explains some ways to prevent issues, how to find issues when they do occur, and some help with ensuring a speedy stored process.

PREVENTING ISSUES

Obviously, the best way to ensure you stored process works is to prevent the user from experiencing any issues. When using prompts there are some easy ways to help the user make the right selections.

DECODER TRICK #1: REQUIRE AN ANSWER

If your stored process code will break unless it gets a prompt value, require the user to provide a value. If your code is similar to the following example, it will fail if the user does not provide the &StatusPrompt value:

```
proc print data=OPS.orders;  
  where status = "&StatusPrompt";  
run;
```

If the stored process requires a prompt value – problem solved. When you create the stored process, select the **Requires a non-blank value** check box on the General pane. With this selection, SAS will not let the user move forward without an answer. SAS adds an asterisk to the prompt and if the user selects Run without providing an answer, an error message is generated. Very nice built-in functionality that will save you hours of coding.

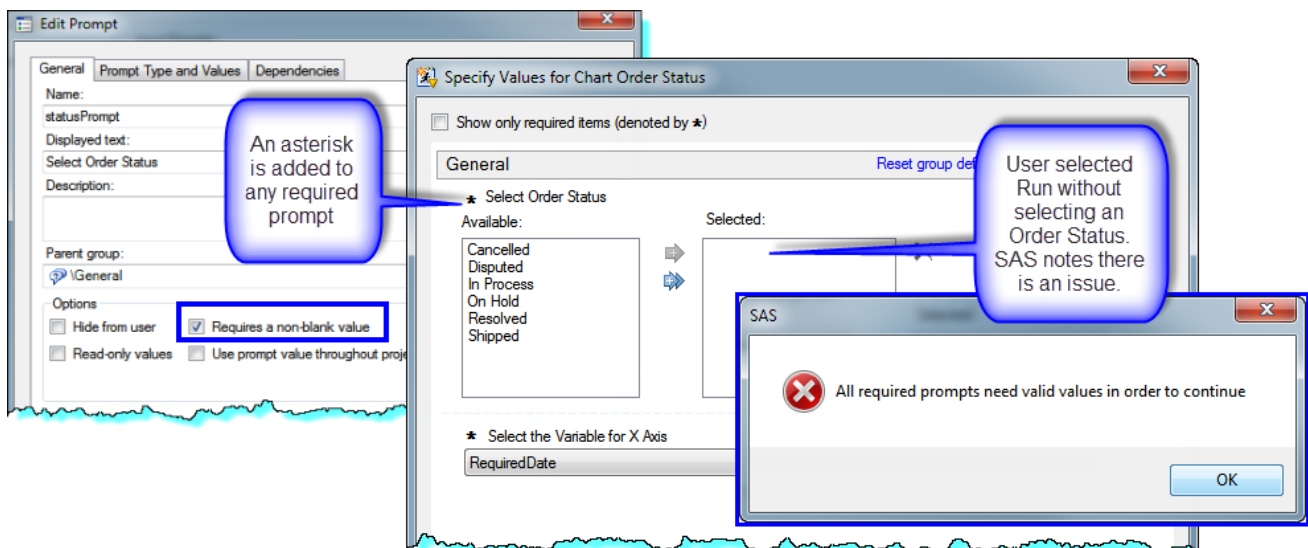


Figure 1 Require a non-blank value

DECODER TRICK #2: PROVIDE A DEFAULT VALUE

Tip #1 can be annoying and you might want to make sure the prompt situation warrants it. As an alternative, you can also set a default value for the prompt. Based on the prompt type, there are many ways to indicate the value. This does allow a user to go forward with the defaults so the stored process does not generate an error message. The only downside I see is if the user doesn't understand they can make changes – but most people are computer-savvy enough that they can understand it.

Hint: Set the default value when testing your stored process so you don't have to fill out the values each time.

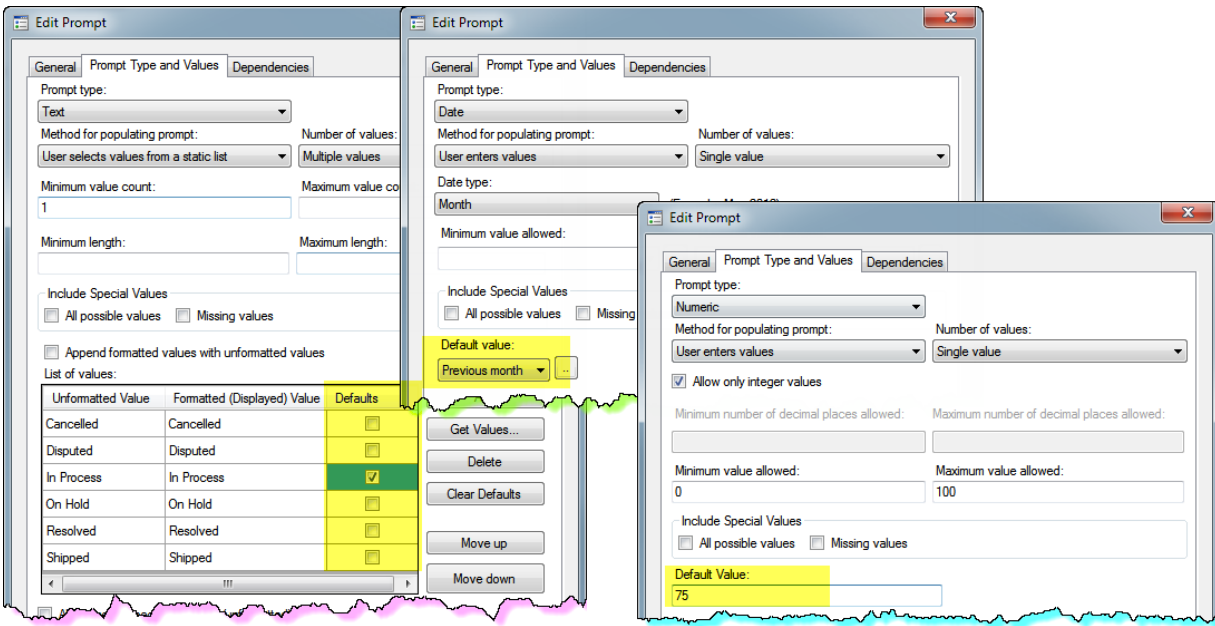


Figure 2 Provide default values

DECODER TRICK #3: PROVIDE MINIMUM AND MAXIMUM PROMPT VALUES

If you know, the data starts and ends in a certain period, then don't allow the user to select past those values. Here's some examples to give you some other ideas. For numeric prompts, if you know the user cannot select a value greater than 100, and then limit the value to 100. Likewise, if you are allowing the user to type a value, then set the minimum value to a length of three or five because it is more likely to be a word. You have to decide based on your data – but these examples might generate some thoughts.

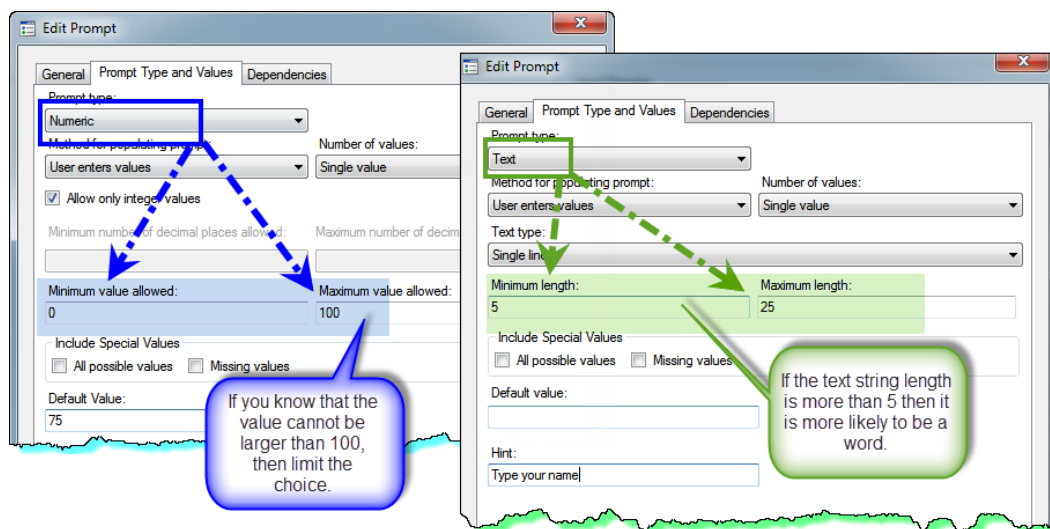


Figure 3 Set minimum and maximum values

FINDING THE STORED PROCESS LOG

The first statement you will hear from an experienced programmer when you report is an issue is “What did the log show?” Some may not realize that there are several ways to see the log file besides asking the SAS Administrator to retrieve the results.

DECODER TRICK #4: ADD A SUPPLIED SHARED PROMPT

Shared prompts are prompts that someone has already created that can be used with multiple stored processes. SAS ships about 15 shared prompts that allow you to select styles, logs, packages, and so on. Many of the tasks you may routinely need when creating a stored process! These prompts are typically found in the **SAS Folders > Products > SAS Intelligence Platform > Samples** location.

You can add a shared prompt as you create your stored process. From the Prompts pane, select Sharing > Add Shared. Navigate to the shared prompt location and select the Show SAS Log shared prompt.

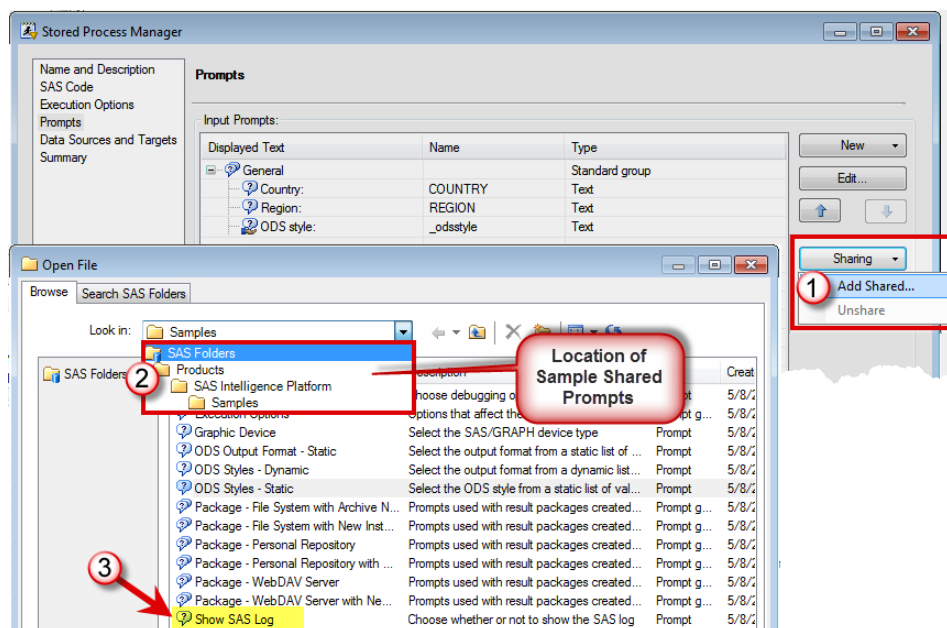


Figure 4 Adding a Shared Prompt

If you answer the prompt as Yes, then the log displays after the results.

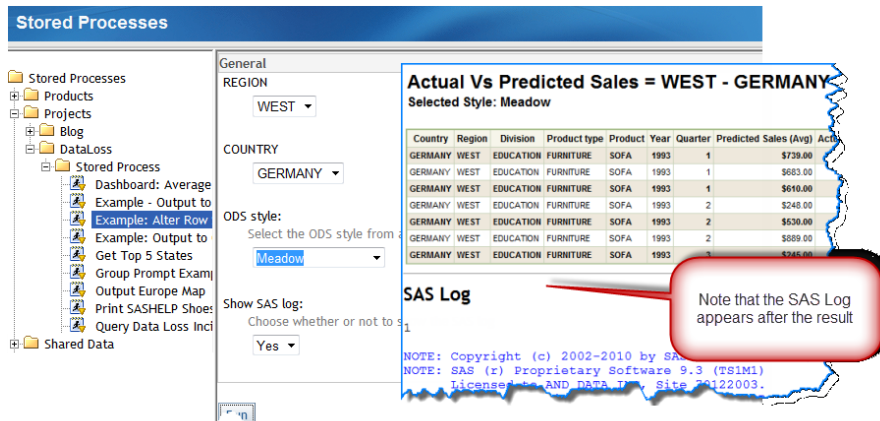


Figure 5 Log appears after the results

DECODER TRICK #5: SKILLFULLY APPLY THE URL

When the stored process fails, a standard error message appears with a button to open the SAS log. If stored process results are not returned as expected, the log may not appear either. You do not know what caused your stored process to visit La-La land. Rather than asking the SAS administrator to get the most recent log off the server, you can add a parameter to the URL path to see the log quickly.

From the SAS Stored Process Web Application, add the `&_DEBUG=131` option to the end of the stored process path and press Enter, as shown in the following figure. After the stored process runs, the log appears in the Web browser as shown below. Now you can locate the errors in your mis-behaving stored process.

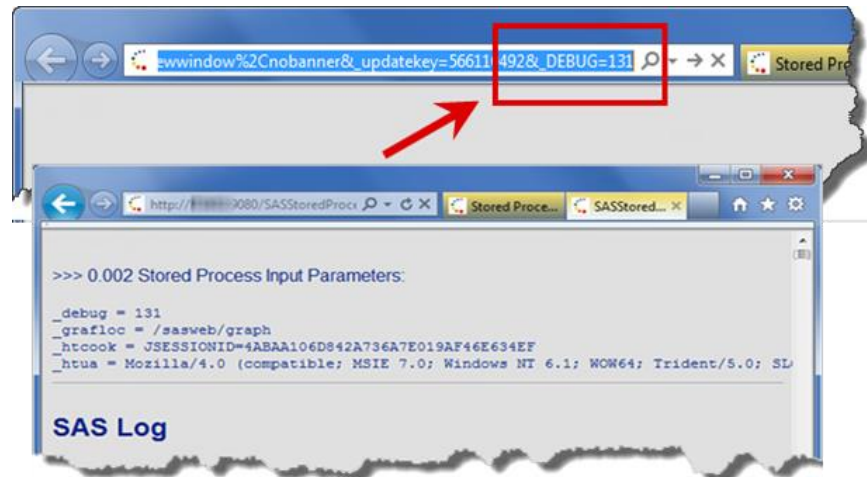


Figure 6 Add the Debug Option

REVIEWING COMMON STORED PROCESS ERRORS

Here are the most common errors that new developers make to help you as you get started.

DECODER TRICK #6: AVOIDING LIBNAME NOT FOUND ERRORS

If you develop your stored process code in SAS Enterprise Guide, then the code uses your connection profile. Thus, it has access to all of your libraries. When the same code runs from the stored process, you might find it generates a "LIBNAME not found error". A best practice is to add the LIBNAME statement to your stored process.

You can use a metadata library or use a system library. If you use a system library, it needs to be one that the target user group can access. For instance, you wouldn't want to use "C:/My Secret Folders" as a path since only you can access the location.

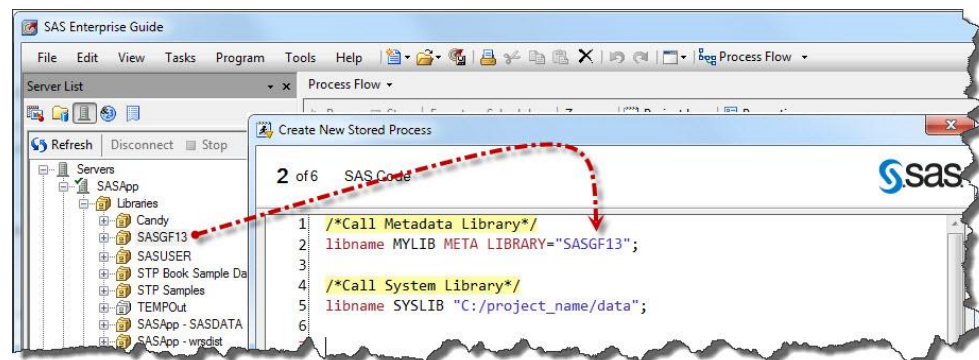


Figure 7 Add a LIBNAME statement

DECODER TRICK #7: AVOIDING UNEXPECTED RESULTS

If you have added the `%STPBegin` and `%STPEnd` macros in your stored process, then you must ensure the Stored Process Wizard does not add it back. If you have added the macros and the wizard has added the macros, then you

see where some interesting results might occur. The worse part of troubleshooting the issue is that it can take several hours with the log to determine the issue.

DECODER TRICK #8: AVOID THE PROMPT ALWAYS USING THE SAME VALUES

Did your prompt work but it keeps making the same choice no matter what you select? For instance, you selected the East region but it returns the West region – no matter how many times you select it? You might want to double-check the code to see if you left the LET statements in it. Many programmers test the prompts to ensure they work or create errors to confirm the prompt will fail.

Tip: Use %Let statements to test your prompts – but be sure to remove them!

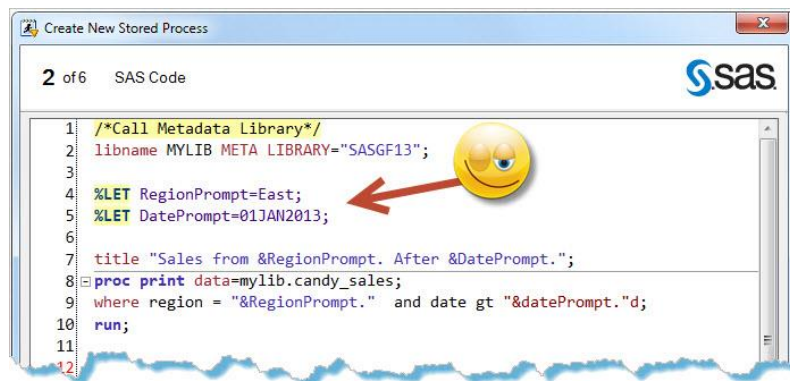


Figure 8 Look for test code left behind

TUNING A STORED PROCESS

The same tricks you use to tune other SAS code applies equally to your stored process code. Here's a few examples:

- Test if the prompts have needed values and if they do not, then end the process early.
- Avoid making multiple trips to the database if you can extract the needed data all at once.
- Try to minimize the data by using Keep and Drop statements.
- If it's a popular stored process, it might make more sense to extract the data during the batch process. This way the data could be prepared earlier in the day and be ready for queries.
- Index the key variables in larger datasets to improve the retrieval speed.

REFERENCES

Aanderud, Tricia and Hall, Angela. 2013. Creating Your First SAS Stored Process. SAS Global Forum 2013 Proceedings. Available at <http://support.sas.com/resources/papers/proceedings13/148-2013.pdf>

Aanderud, Tricia and Hall, Angela. 2012. The 50 Keys to Learning SAS Stored Processes. Raleigh, NC: Siamese Publishing.

RECOMMENDED READING

Aanderud, Tricia and Hall, Angela. 2012. Building Business Intelligence with SAS. Cary, NC: SAS Press.

BI Notes for SAS Users, Aanderud, Tricia. Available at <http://www.bi-notes.com>

Real BI for SAS Users, Hall, Angela. Available at <http://blogs.sas.com/content/bi/>

Lafler, Kirk Paul, 2012. Top Ten SAS® Performance Tuning Techniques. Available at: <http://support.sas.com/resources/papers/proceedings12/357-2012.pdf>

CONTACT INFORMATION

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

Tricia Aanderud
And Data Inc
tricia.aanderud@and-data.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.