

Introducing Interactive Data Step Debugger: What you can do with SAS Data Step Debugger (SAS® Enterprise Guide® 7.13)

Kevin Kim, Synteract, Inc.

ABSTRACT

Fall of 2016, the Interactive DATA step debugger has been released (SAS Enterprise Guide 7.13).

Debugging is the process of removing logic errors from a program. Unlike syntax errors, logic errors do not stop a program from running. Instead, they cause the program to produce unexpected results.

While the SAS log can help you identify data errors, the DATA step debugger offers you an easier, interactive way to identify logic errors, and sometimes data errors, in DATA steps. You want to have this debugging tool.

INTRODUCTION

Imagine you completed a program but you got a different results than what you have expected... What do you do? Well, starting investigation... aka, Debugging.

You will need to find logic errors in the program that caused the wrong results. This process sometimes can take time.

With the DATA Step Debugger, you can find logic errors in the program (DATA step). You can watch the variable values in a program change as the program runs. You can execute the program line by line, and you can also set specific breakpoints in the program that stops the process, so that you can observe the value you are watching. With every step, the watch window is updated with the latest values of the variables in your step. When a variable changes value, it's colored red. If you want to the DATA step to break processing when a certain variable changes value, check the Watch box for that variable.

By observing the results that are displayed, you can determine where the logic error lies. Because the debugger is interactive, you can repeat the process of issuing commands and observing the results as many times as needed in a single debugging session.

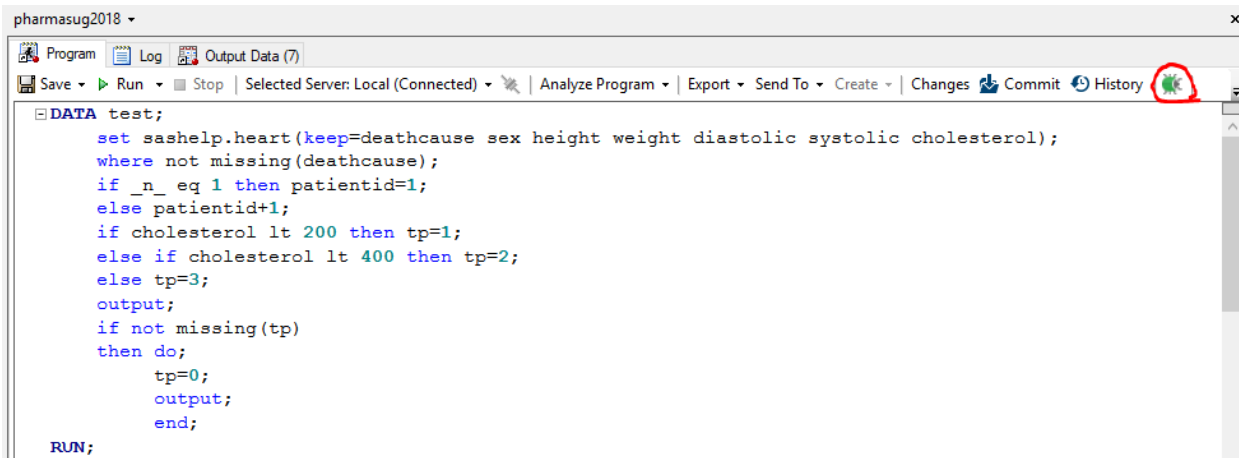
USING THE DATA STEP DEBUGGER IN SAS ENTERPRISE GUIDE

The new debugging environment, introduced in SAS Enterprise Guide 7.13, and these are added features.

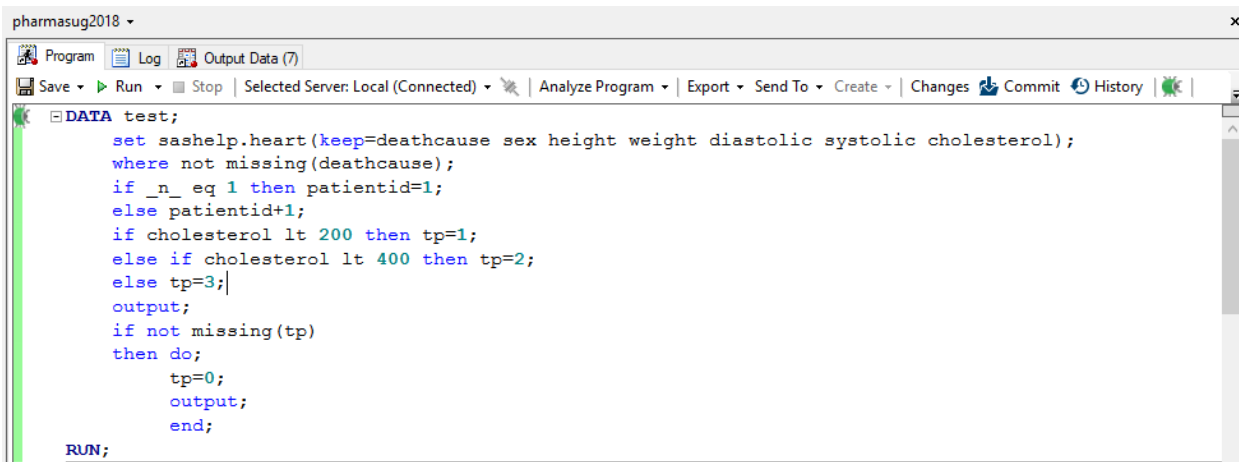
- The new DATA Step Debugger is a tool that enables you to find logic errors in a DATA step program. With the DATA Step Debugger, you can watch the variable values in a program change as the program runs. You can execute the program line by line, and you can also set specific breakpoints in the program.
- You can now transfer files from your local computer to a SAS server or from a SAS server to your local computer by using the Copy Files task. The Copy Files task works in a similar way to an FTP application. However, this task relies on the SAS protocols to complete the file transfers and does not require an FTP server.
- You can upload your existing SAS 9.4 data to SAS Cloud Analytic Services (CAS) by using the new Upload to CAS task.
- Starting with this release, the default graph format is PNG. Use the new Graph Format for Built-in Graph Tasks option to set the default format for the graph tasks, such as Bar Chart, Line Chart, and Pie Chart.
- You can specify whether to continue executing a process flow if an error is encountered.
- Server file navigation now defaults to your Documents folder on Windows servers. In addition, the file navigation now contains folder shortcuts.

DEBUGGING

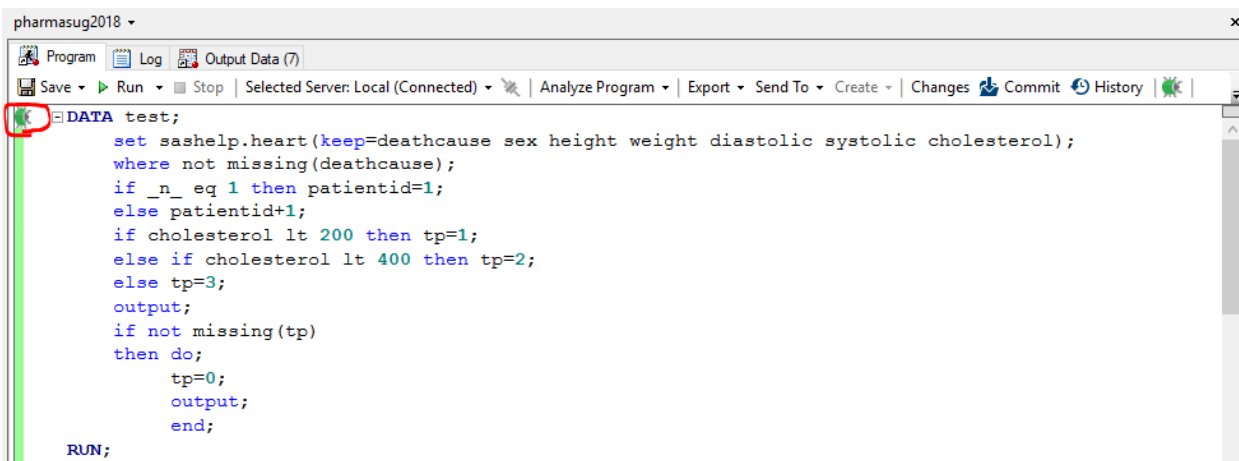
To activate the debugger, click the new "bug" toolbar icon in the program editor window.



Once activated, it shows 'green' vertical bar on the left side of the DATA Step in the program editor.

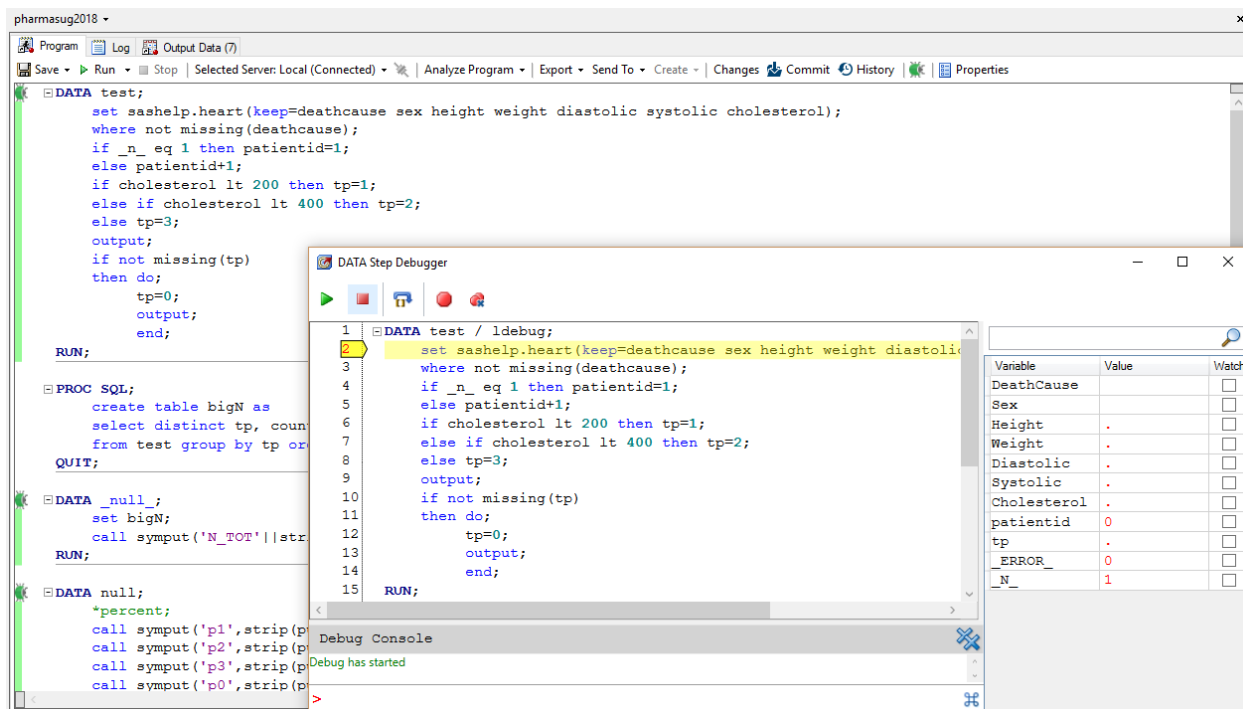


You can click the bug in the left "gutter" of the program editor to begin a debug session. (You can also press F5 to debug the active DATA step.)



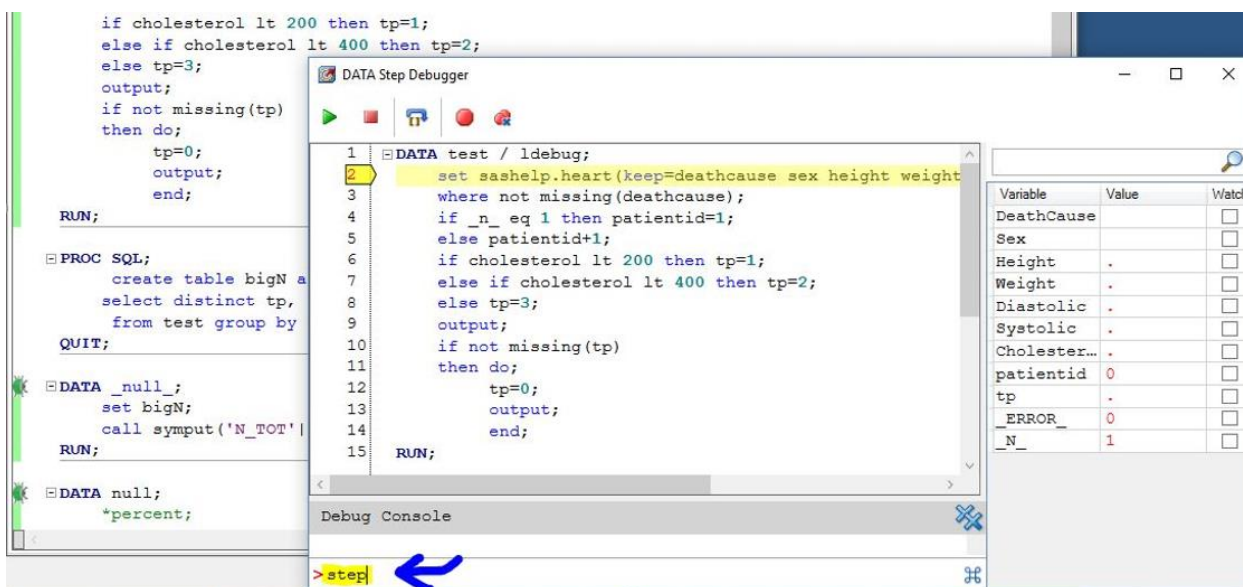
It opens 'DATA Step Debugger' window.

Examine the screenshot below. You see the source window on top and the console window at the bottom, plus a convenient "watch" window that shows much of the content in the program data vector (PDV). That's all of the variables defined in the DATA step, plus automatic variables like `_N_` and `_ERROR_`.



As you step through the DATA step, the line pointer in the source window advances to show the next line that will execute.

You can use keyboard shortcuts (F10), the toolbar, or type a command ("step") to execute that line and advance.



With every step, the watch window is updated with the latest values of the variables in your step. When a variable changes value, it's colored red.

The screenshot shows the SAS Data Step Debugger interface. The main window displays the SAS code for a DATA step named 'test'. The code includes a SET statement for 'sashelp.heart', a WHERE clause, and several IF-THEN statements for patient identification and treatment assignment. The Watch window on the right lists variables and their current values. A blue arrow points to the 'Value' column header.

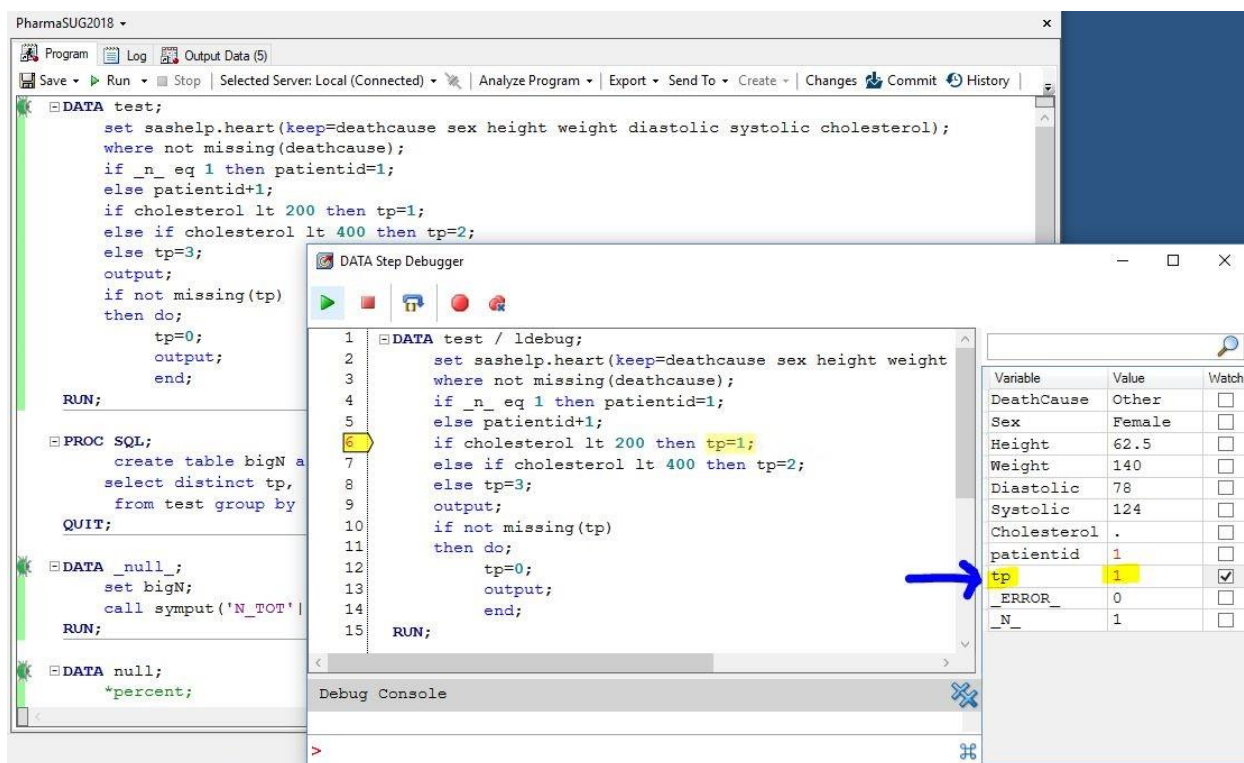
Variable	Value	Watch
DeathCause	Other	<input type="checkbox"/>
Sex	Female	<input type="checkbox"/>
Height	62.5	<input type="checkbox"/>
Weight	140	<input type="checkbox"/>
Diastolic	78	<input type="checkbox"/>
Systolic	124	<input type="checkbox"/>
Cholesterol	.	<input type="checkbox"/>
patientid	0	<input type="checkbox"/>
tp	.	<input type="checkbox"/>
ERROR_	0	<input type="checkbox"/>
N	1	<input type="checkbox"/>

If you want the DATA step to break processing when a certain variable changes value, check the 'Watch' box for that variable.

This screenshot shows the same SAS Data Step Debugger interface as the previous one, but with the 'Watch' checkbox checked for the 'tp' variable in the Watch window. The 'tp' variable's value is currently '.'.

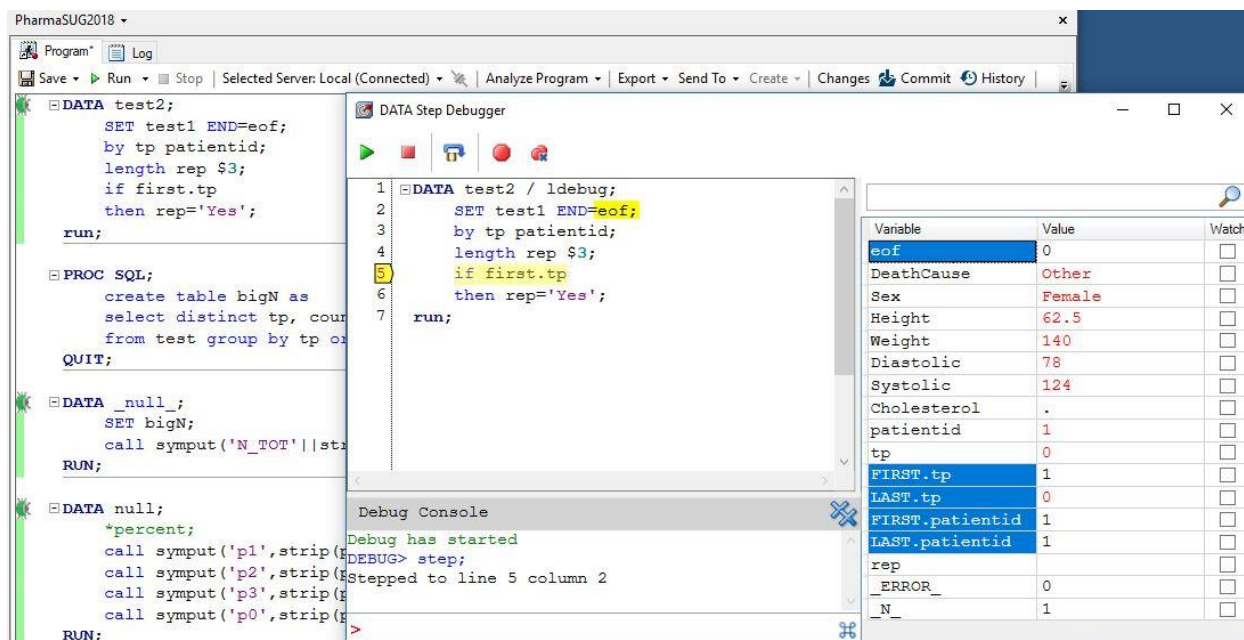
Variable	Value	Watch
DeathCause	Other	<input type="checkbox"/>
Sex	Female	<input type="checkbox"/>
Height	62.5	<input type="checkbox"/>
Weight	140	<input type="checkbox"/>
Diastolic	78	<input type="checkbox"/>
Systolic	124	<input type="checkbox"/>
Cholesterol	.	<input type="checkbox"/>
patientid	0	<input type="checkbox"/>
tp	.	<input checked="" type="checkbox"/>
ERROR_	0	<input type="checkbox"/>
N	1	<input type="checkbox"/>

It stops processing when the 'Watch' variables change values.



Here's another example of debugging a different DATA step program. This program uses a BY statement and FIRST.variable logic, and you can see the additional automatic variables (FIRST.tp and LAST.tp and FIRST.patientid and LAST.patientid) that the debugger is tracking. I also used END=eof on the SET statement; that adds the eof "flag" variable into the mix during run time.

Note: Debugging will start with the server in its current state. Make sure to run any dependent code prior to debugging your DATA step, if necessary

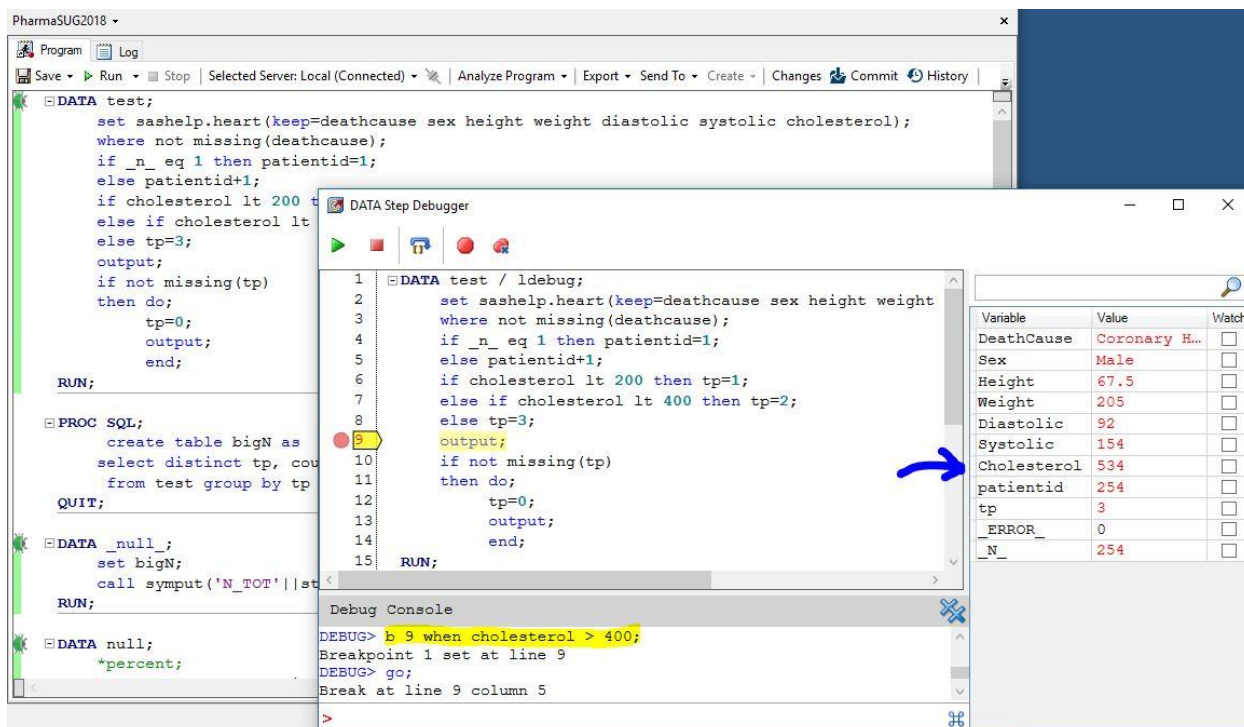
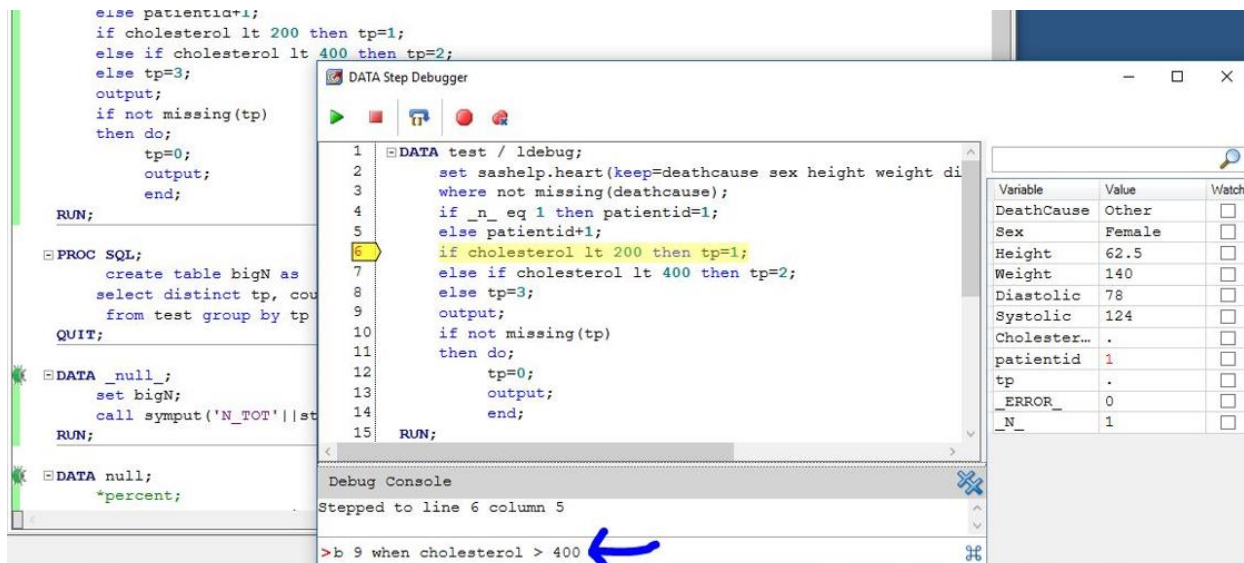


KEEP DEBUGGING

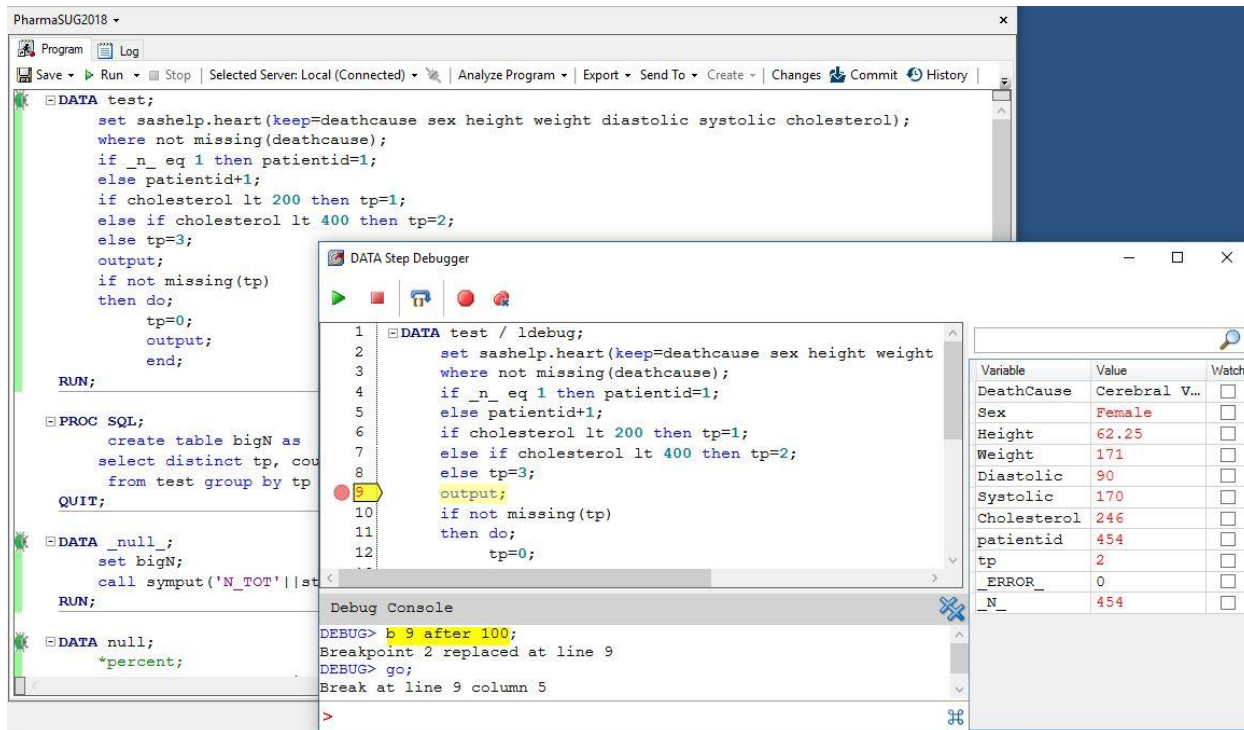
you can perform standard debugging operations by clicking buttons or pressing keyboard shortcuts. For example, you can step line-by-line through your code (F10), set breakpoints (F9), clear all breakpoints (Shift+F9), inspect variable values, and continue execution (F5).

You can also enter debugging commands in the Debug Console window.

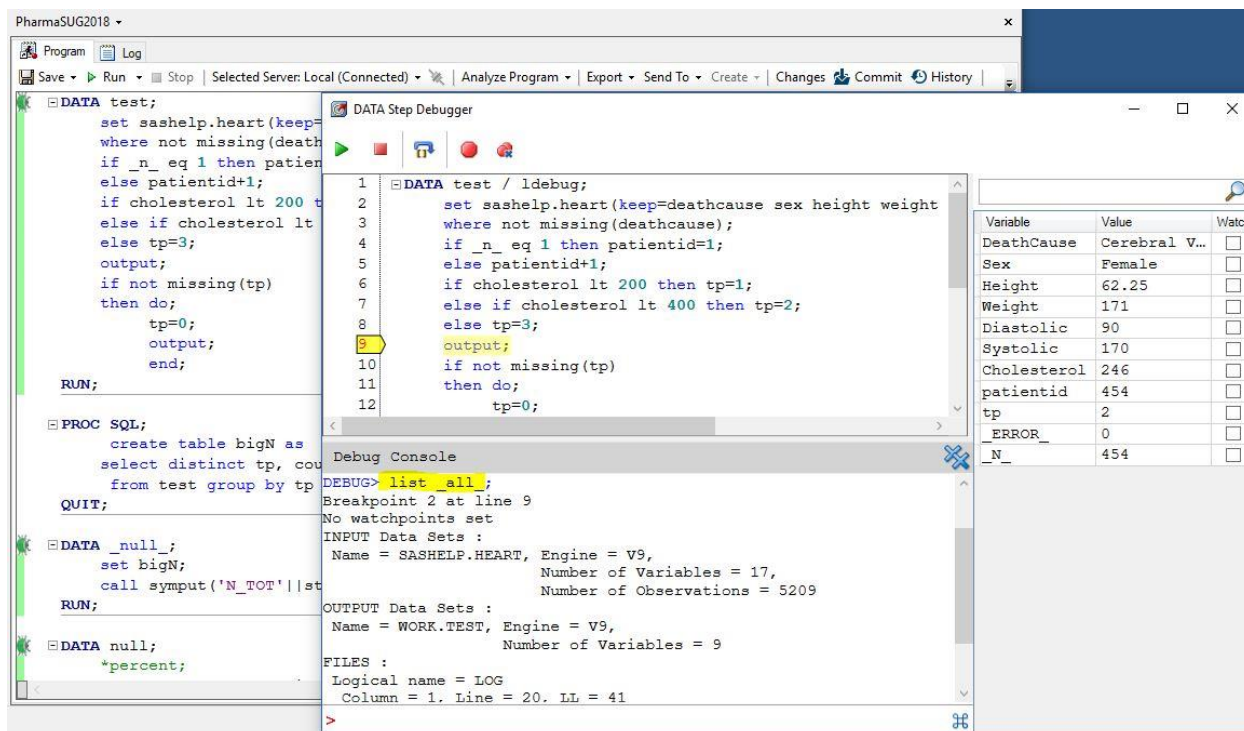
The DATA step debugger allows you to set breakpoints that trigger on specific conditions. For example, "b 9 when cholesterol > 400" will break on Line 9 when the value of cholesterol exceeds 400.




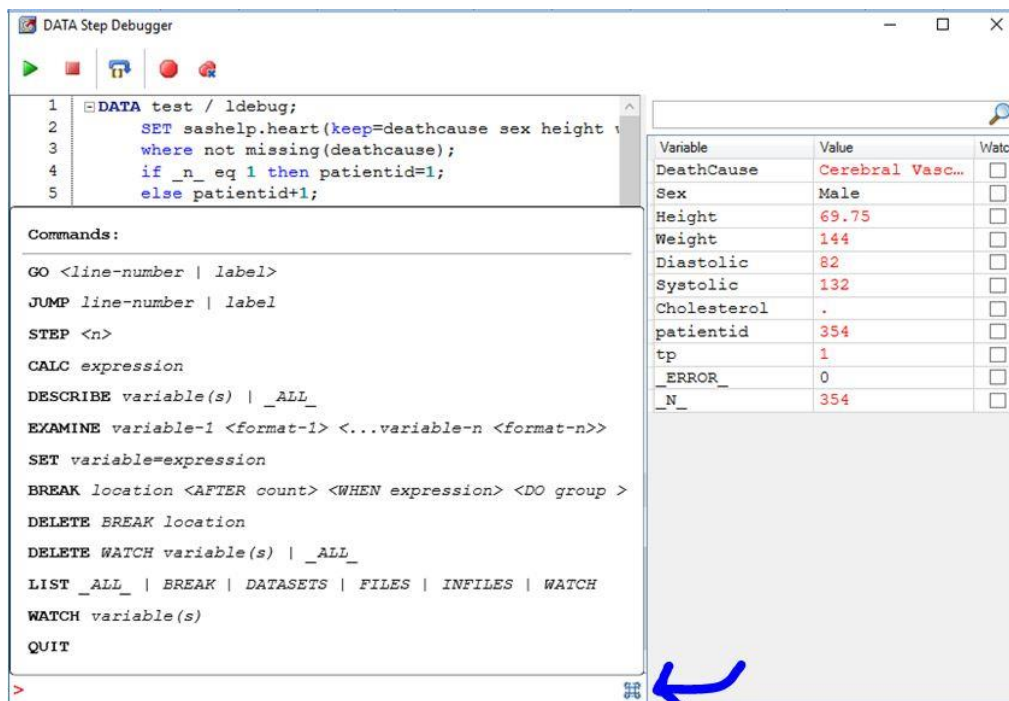
"b 9 after 100" will break on Line 9 after 100 loops through the DATA step. You can set and clear line-specific breakpoints by clicking in the "gutter" (that left-hand margin next to the line numbers).



The "list _all_" command reveals the details about your open data sets and files.



You can press the icon  to the right of the command line for a listing of the supported commands and their respective syntax.



The DATA step debugger in SAS Enterprise Guide supports almost all the same commands supported by the DATA step debugger in Base SAS. See the chapter, “Dictionary of DATA Step Debugger Commands” in *Base SAS® 9.4 Utilities: Reference* (SAS Institute Inc. 2013a), for a list of Commands (<http://support.sas.com/documentation/cdl/en/lebaseutilref/64791/HTML/default/viewer.htm#n1ulcr5494fc5on1gg1cdzuhez76.htm>).

CONCLUSION

DATA Step Debugger is a tool that enables you to find logic errors in a DATA step program. With the DATA Step Debugger, you can watch the variable values in a program change as the program runs. You can execute the program line by line, and you can also set specific breakpoints in the program.

One important limitation currently it has is that this tool is for debugging DATA step code. It can't be used to debug PROC SQL or PROC IML or SAS macro programs. It is expected that Debugging for PROC step or macro programs will be added in the future releases.

REFERENCES

Andy Bayliss, SAS Institute / Joe Flynn, SAS Institute Inc. “The SAS DATA Step Debugger in SAS® Enterprise Guide®” <http://www.phusewiki.org/docs/Conference%202017%20CT%20papers/CT06.pdf>
<http://www.phusewiki.org/docs/Conference%202017%20CT%20Presentations/CT06.pdf>

Casey Smith, SAS Institute Inc., “Increasing Your Productivity with New Features in SAS® Enterprise Guide®” <http://support.sas.com/resources/papers/proceedings17/SAS0562-2017.pdf>

Joe Flynn, SAS Institute Inc., “Step through Your DATA Step: Introducing the DATA Step Debugger in SAS® Enterprise Guide®” <https://support.sas.com/resources/papers/proceedings17/SAS0447-2017.pdf>

SAS Institute Inc., 1999. “SAS Institute Inc., SAS® Language Reference, Version 8”
<https://support.sas.com/documentation/cdl/en/lrdict/64316/PDF/default/lrdict.pdf>

RECOMMENDED READING

SAS Institute Inc. 2013. "Dictionary of DATA Step Debugger Commands." *Base SAS® 9.4 Utilities: Reference*. Cary, NC: SAS Institute Inc.

<http://support.sas.com/documentation/cdl/en/lebaseutilref/64791/PDF/default/lebaseutilref.pdf>

CONTACT INFORMATION

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

Kevin Kim
Synteract, Inc.
Kevin.kim@synteract.com
<https://www.synteract.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.