

Step through Your DATA Step: Introducing the DATA Step
Debugger in SAS® Enterprise Guide®

The SAS DATA Step Debugger in SAS® Enterprise Guide®

Andy Bayliss, SAS Institute

Joe Flynn, SAS Institute Inc.

Introduction

- What is a debugger?
 - Control DATA step execution
 - Examine and manipulate variables
- Why do I need a debugger?
 - Fix bugs
 - Understand legacy code

Introduction

- Basic Debugging Concepts
- Enterprise Guide DATA step debugger
- Advanced Debugger Commands
- Base SAS DATA step debugger

Basic Debugging Concepts

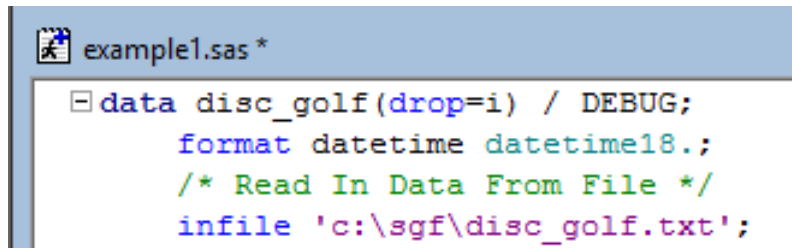
Introduction

- Initializing the debugger
- Controlling Program Execution
- Suspending Program Execution

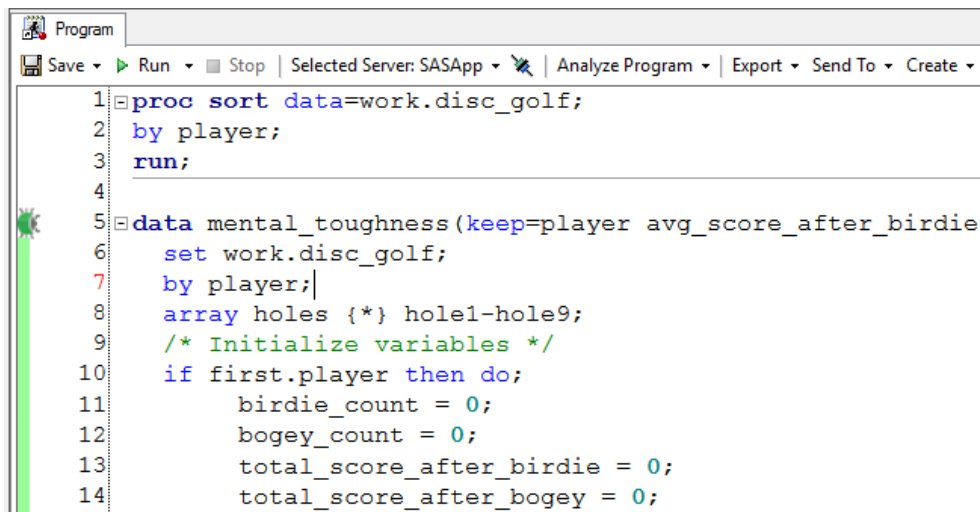
Basic Debugging Concepts

Initializing the debugger

- Base SAS
 - Add / DEBUG option to DATA step
- SAS Enterprise Guide
 - Highlight DATA step regions
 - Injects LDEBUG option



```
example1.sas *  
data disc_golf(drop=i) / DEBUG;  
  format datetime datetime18.;  
  /* Read In Data From File */  
  infile 'c:\sgf\disc_golf.txt';
```



```
Program  
Save Run Stop Selected Server: SASApp Analyze Program Export Send To Create  
1 proc sort data=work.disc_golf;  
2 by player;  
3 run;  
4  
5 data mental_toughness(keep=player avg_score_after_birdie  
6 set work.disc_golf;  
7 by player;  
8 array holes {*} hole1-hole9;  
9 /* Initialize variables */  
10 if first.player then do;  
11 birdie_count = 0;  
12 bogey_count = 0;  
13 total_score_after_birdie = 0;  
14 total_score_after_bogey = 0;
```

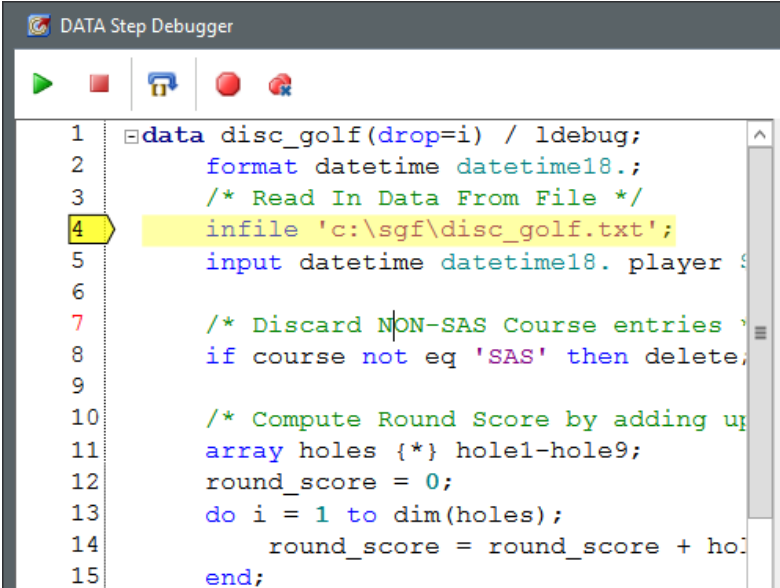
Basic Debugging Concepts

Initializing the Debugger

- Compile the step
- Display debugger window
- Pause execution

```
DEBUGGER LOG
DATA STEP Source Level Debugger
Stopped at line 4 column 5

PROC DEBUGGER SOURCE
1 data disc_golf(drop=i) / debug;
2   format datetime datetime18.;
3   /* Read In Data From File */
4   infile 'c:\sgf\disc_golf.txt';
5   input datetime datetime18. player $26. cours
6
7   /* Discard NON-SAS Course entries */
8   if course not eq 'SAS' then delete;
9
10  /* Compute Round Score by adding up hole 1-
11   array holes {*} hole1-hole9;
12   round_score = 0;
13   do i = 1 to dim(holes);
14     round_score = round_score + holes[i];
15   end;
16 run;
```



The screenshot shows the 'DATA Step Debugger' window. The title bar reads 'DATA Step Debugger'. Below the title bar is a toolbar with icons for running, stopping, stepping, and other debugging actions. The main area displays the SAS source code for a data step named 'disc_golf'. Line 4, 'infile 'c:\sgf\disc_golf.txt';', is highlighted in yellow, and a yellow arrow points to it from the left margin. The code continues with input statements, conditional deletions, and array processing for calculating round scores.

```
1 data disc_golf(drop=i) / ldebug;
2   format datetime datetime18.;
3   /* Read In Data From File */
4   infile 'c:\sgf\disc_golf.txt';
5   input datetime datetime18. player $
6
7   /* Discard NON-SAS Course entries */
8   if course not eq 'SAS' then delete;
9
10  /* Compute Round Score by adding up
11   array holes {*} hole1-hole9;
12   round_score = 0;
13   do i = 1 to dim(holes);
14     round_score = round_score + ho
15   end;
```

Basic Debugging Concepts

Controlling Program Execution

- Step
 - Execute one statement in DATA step (by default)
 - Return control to the user
- Jump
 - Move execution to specified location
 - Does not execute intervening statements
 - Return Control To User
- Go
 - Execute until breakpoint, watch variable changes, or DATA step complete

Basic Debugging Concepts

Suspending Program Execution

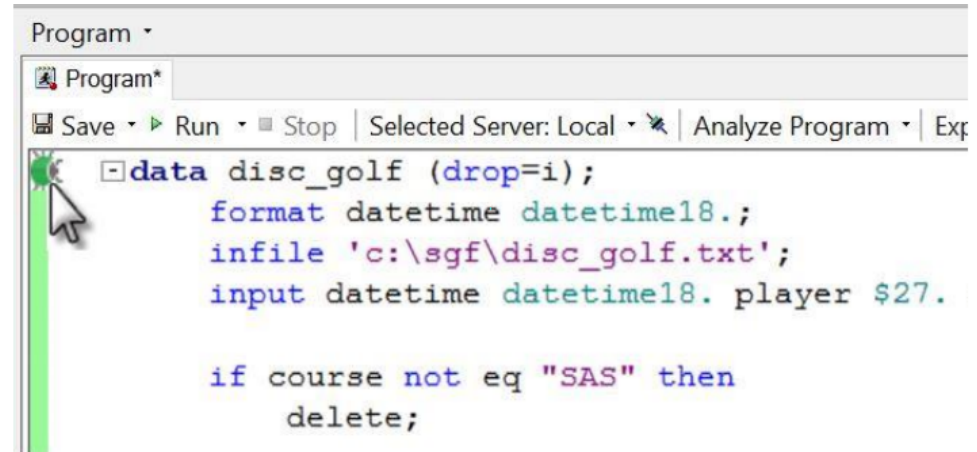
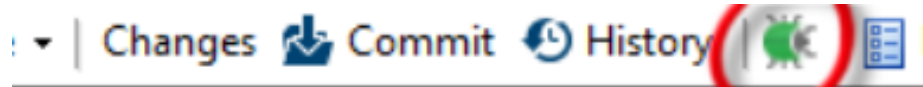
- Break
 - Suspends execution at specified statement
 - Only suspends when that line is about to be executed
- Watch
 - Suspends execution when value of a variable changes

Enterprise Guide Debugger

- Enable/Launch Debugger
- Tour Debugger Window
- Demo – At the SAS booth

Enterprise Guide Debugger

- Enabling the debugger
- Launching the debugger



Enterprise Guide Debugger

Debug Window Tour

DATA Step Debugger

Toolbar

Debug Source Window

```
3 infile 'c:\sgf\disc_golf.txt';
4 input datetime datetime18. player $27. course $CHAR15. hole1-hc
5
6 if course not eq "SAS" then
7     delete;
8
9 /* Goal 1: Find the lowest round score for each course */
10 array holes{1-9} hole1-hole9; do i = 1 to dim(holes);
10 2: Keep track of lowest round */
11
12 /* Initialize the low */
13 if _n_ eq 1 then
14     do; course_record = round_score; record_holder = player; er
15     /* Update All Time Low */
```

Debug Console

Break at line 6 column 2
DEBUG> go;
Break at line 6 column 2
>

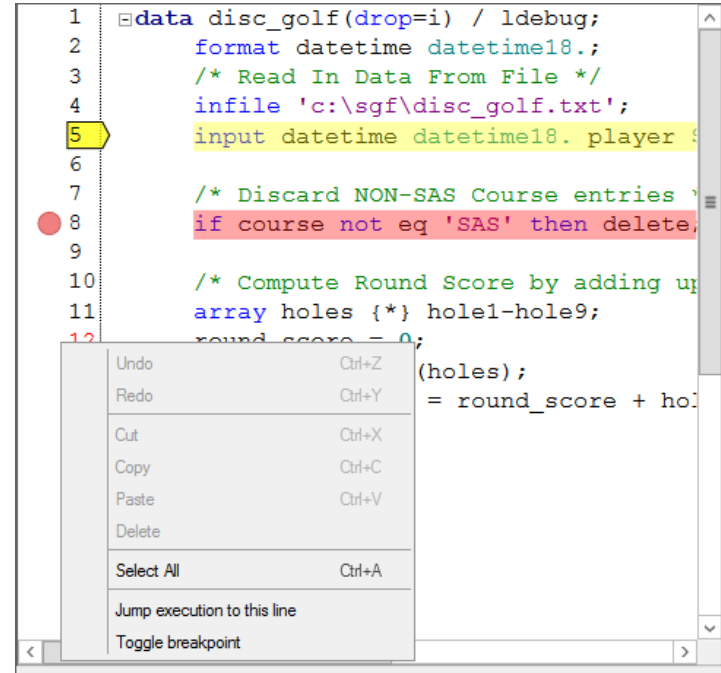
Watch Window

Variable	Value	Watch
datetime	15NOV13:11:...	☐
player	Joe Flynn	☐
course	SAS	☐
hole1	4	☐
hole2	3	☐
hole3	.	☐
hole4	.	☐
hole5	3	☐
hole6	3	☐
hole7	4	☐
hole8	5	☐
hole9	3	☐
round_score	.	☐
i	.	☐
course_reco...	.	☐
record_hold...	.	☐
ERROR	0	☐
N	16	☐

Enterprise Guide Debugger

Debug Source Window

- Highlight Executing Line Number and Source
- Highlight Break Points
- Toggle Break Points
- Jump To Source Line



The screenshot displays the 'Debug Source Window' in the Enterprise Guide Debugger. It shows a SAS program with the following code:

```
1 data disc_golf(drop=i) / ldebug;  
2   format datetime datetime18.;  
3   /* Read In Data From File */  
4   infile 'c:\sgf\disc_golf.txt';  
5   input datetime datetime18. player $;  
6  
7   /* Discard NON-SAS Course entries */  
8   if course not eq 'SAS' then delete;  
9  
10  /* Compute Round Score by adding up  
11  array holes {*} hole1-hole9;  
12  round_score = 0;  
    (holes);  
    = round_score + hol
```

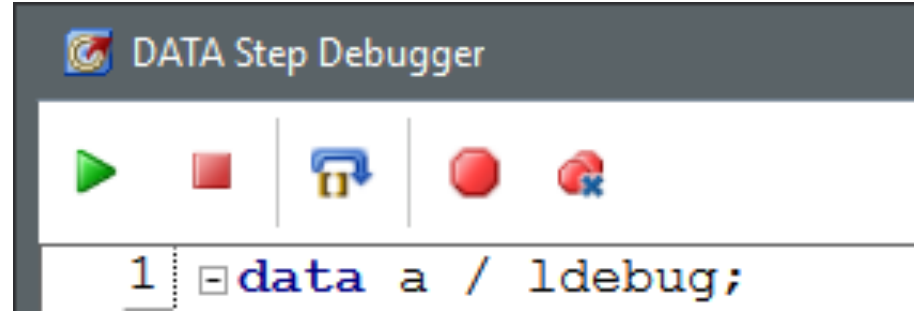
A yellow highlight is on line 5. A red circle on line 8 indicates a breakpoint. A context menu is open over line 12, showing the following options:

- Undo (Ctrl+Z)
- Redo (Ctrl+Y)
- Cut (Ctrl+X)
- Copy (Ctrl+C)
- Paste (Ctrl+V)
- Delete
- Select All (Ctrl+A)
- Jump execution to this line
- Toggle breakpoint

Enterprise Guide Debugger

Toolbar

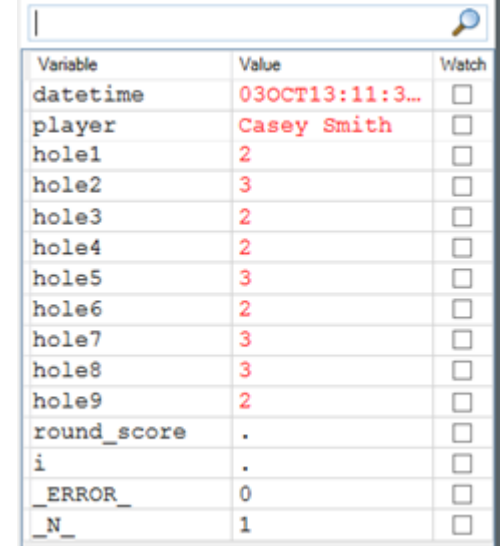
- GO 
- Stop 
- STEP 
- Toggle Breakpoint 
- Clear All Breakpoints 



Enterprise Guide Debugger

Watch Window

- Searchable list of Variables in PDV
 - Includes `_N_`, `_ERROR_`, `FIRST.` And `LAST.` and any automatic variables you may use
- Monospace font for easy space detection
- Easily toggle a WATCH on specific variables
- Update the value of variables during execution
- Highlight variables that have changed

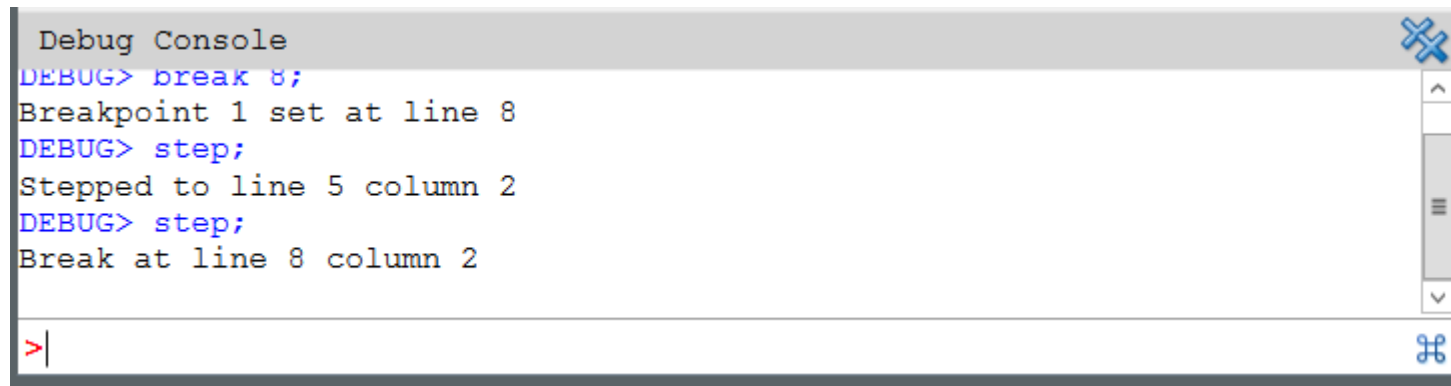


Variable	Value	Watch
datetime	03OCT13:11:3...	☐
player	Casey Smith	☐
hole1	2	☐
hole2	3	☐
hole3	2	☐
hole4	2	☐
hole5	3	☐
hole6	2	☐
hole7	3	☐
hole8	3	☐
hole9	2	☐
round_score	.	☐
i	.	☐
ERROR	0	☐
N	1	☐

Enterprise Guide Debugger

Debug Console

- Running Log of Session
- Issue Commands not surfaced by UI

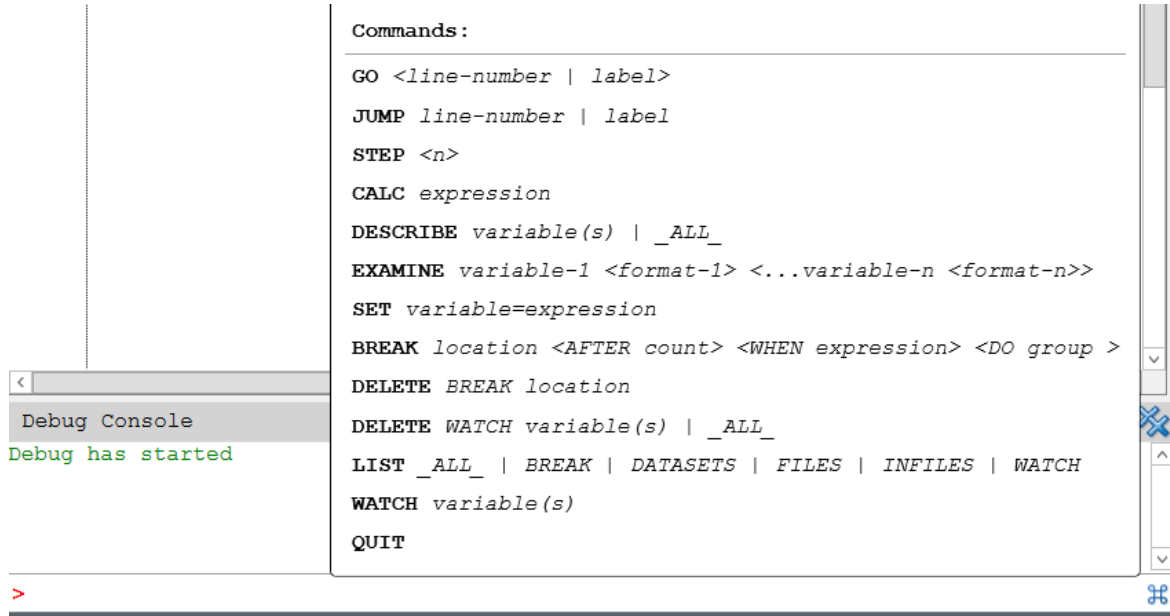
A screenshot of the 'Debug Console' window in the Enterprise Guide Debugger. The window has a title bar with the text 'Debug Console' and a close button (X) on the right. The main area contains a log of commands and responses. The commands are in blue text, and the responses are in black text. The log shows: 'DEBUG> break 8;' followed by 'Breakpoint 1 set at line 8', 'DEBUG> step;' followed by 'Stepped to line 5 column 2', and 'DEBUG> step;' followed by 'Break at line 8 column 2'. At the bottom, there is a prompt '>' followed by a vertical bar, indicating the command input area. On the right side of the console, there are standard UI controls: an up arrow, a scroll bar, a menu icon (three horizontal lines), and a down arrow. At the bottom right corner of the console, there is a maximize button icon (two overlapping squares).

```
Debug Console
DEBUG> break 8;
Breakpoint 1 set at line 8
DEBUG> step;
Stepped to line 5 column 2
DEBUG> step;
Break at line 8 column 2
>|
```

Enterprise Guide

Debug Console

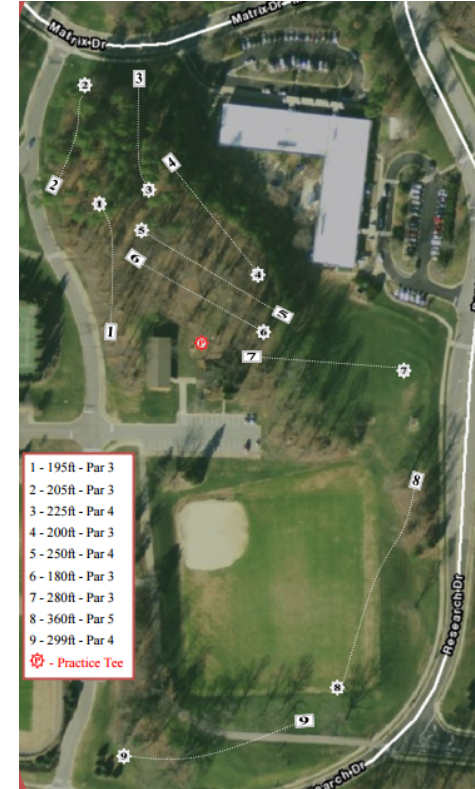
- Quick reference on debug command syntax



Enterprise Guide Debugger

Demo

- Example 1: Import Data & Calculate Round Score
- Example 2: Calculate course record
- Example 3: Calculate mental toughness





SAS Enterprise Guide Community

Online. Everyday.

[Communities.sas.com/enterprise-guide](https://communities.sas.com/enterprise-guide)

