



# SAS Enterprise Guide for Programmers

Going deep

If you're writing code in SAS Enterprise Guide,  
you're doing it wrong.

**BUSTED**



## Topics for programmers

- Make yourself at home (customize)
- Program editor tips
- Creating an all-code project
- Building a production-ready SAS program
- Using built-in version control
- Using external version control
- The debugger is your friend
- Checking on your macro vars

# Customize your options

Managing Results

Replace results:

Show generated wrapper code in SAS log

Recent Files Menu

Number of items to display in recent projects/programs menus:

Result Formats

<input type="checkbox"/> SAS Report	<input checked="" type="checkbox"/> HTML	<input type="checkbox"/> PDF
<input type="checkbox"/> RTE	<input type="checkbox"/> Text output	<input type="checkbox"/> PowerPoint
<input type="checkbox"/> Excel		

Default:

# Program editor tricks

- Turn on line numbers
- Get the tabs out
- Use abbreviations
- Let the editor format your code
- Zoom in and out
- Change the program editor font
- Column select with block select feature
- Use regex for find/replace
- Split pane viewing
- Break out to another editor

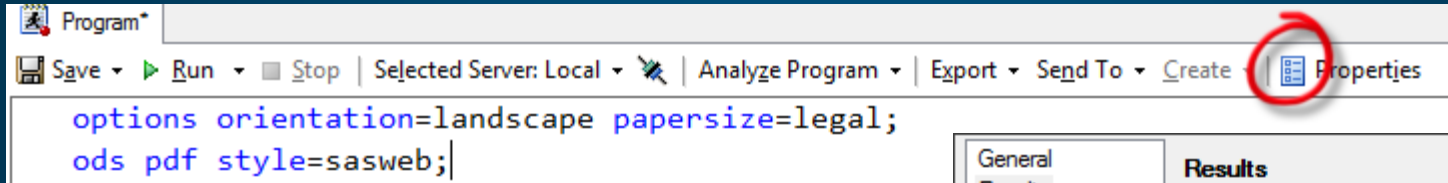
```
42 data _contents&i.(keep=zip memname);
43 length zip $200 memname $200; zip="%targdir/&&zname&i"; fid=dopen("targzip");
44 if fid=0 then stop; memcount=dnum(fid);
45 do i=1 to memcount; memname=dread(fid,i);
46     /* save only full file names, not directory names */
47     if (first(reverse(trim(memname))) ^= '/') then output; end; rc=dclose(fid);
48 run;
49
50 filename targzip clear;
51 %end;
52
53 /* Combine the member names into a single data set */
54 /* the colon notation matches all files with "_contents" prefix */
55 data &outlist.;
56     set _contents;
57 run;
```

<https://blogs.sas.com/content/sasdummy/sas-program-editor-tricks/>

# Organize a Code-only Project

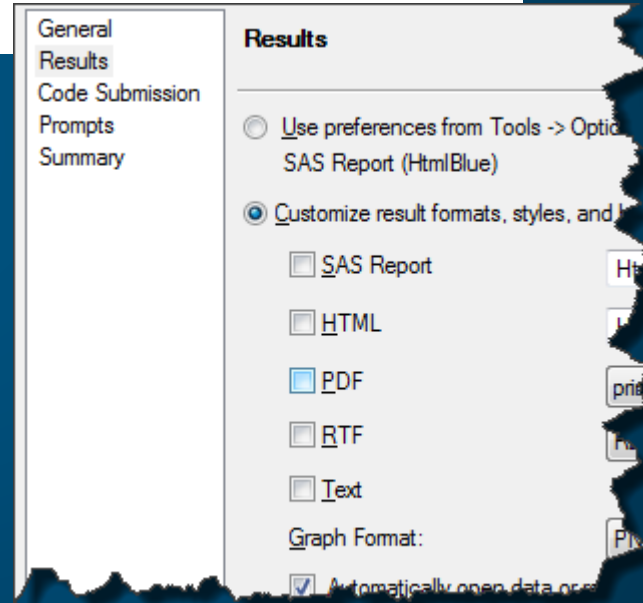
The screenshot displays the SAS Enterprise Guide interface for a project named "CodeProject - SAS Enterprise Guide". The "Project Tree" on the left shows a hierarchy: "Autoexec" (containing "Programs" and "AssignLibs"), "GitHub reporting" (containing "Note" and "Programs"), and "Programs" (containing "githubstats" and "githubemail"). The main workspace shows a workflow diagram with two "Run" icons labeled "githubstats" and "githubemail", connected by a dashed arrow. A yellow warning box is overlaid on the diagram, stating: "These programs rely on a stored /auth file, which must reside in your user profile." The interface includes a menu bar (File, Edit, View, Tasks, Favorites, Program, Tools, Help) and a toolbar with various icons.

# Run your code like it will run in prod



The screenshot shows the SAS Studio interface. The top menu bar includes 'Save', 'Run', 'Stop', 'Selected Server: Local', 'Analyze Program', 'Export', 'Send To', 'Create', and 'Properties'. The 'Properties' button is circled in red. Below the menu bar, a code editor contains the following SAS code:

```
options orientation=landscape papersize=legal;  
ods pdf style=sasweb;
```



The screenshot shows the 'Results' tab of the Properties dialog box. The 'Results' section is active, and the 'Customize result formats, styles, and...' option is selected. The following options are visible:

- Use preferences from Tools -> Options -> SAS Report (HtmlBlue)
- Customize result formats, styles, and...

Under the 'Customize result formats, styles, and...' option, the following checkboxes are visible:

- SAS Report
- HTML
- PDF
- RTF
- Text

The 'Graph Format:' section is also visible, with the 'Automatically open data or...' checkbox checked.

# Program history: version control built in

Commit Changes

filenameZipExample

```
1 ----- Difference (@@ -52,6 +5)
2       RATWeight;
3       run;
4
5 +title "Summary stats for the Hazmat data";
6 /* Summary stats for this data */
7 proc means data=hazmat;
8       run;
9
```

Description of changes:  
Added a title

Commit Reset Changes

History for filenameZipExample (Process Flow)

Compare Compare with Editor Edit Description Revert Help (F1)...

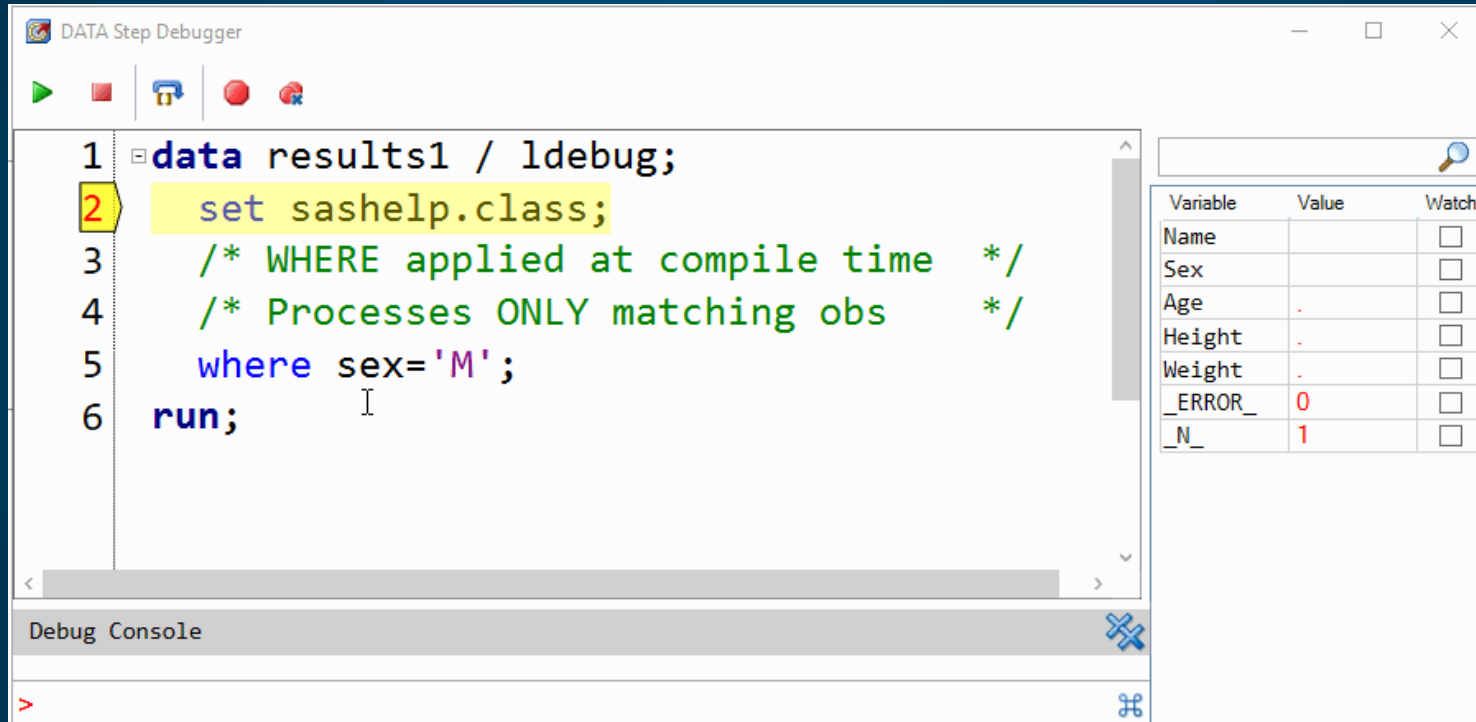
Version	Date	Author	Description
2	10/12/2014 11:35:34 AM	PUBLIC	Added a title
1	10/9/2014 2:24:39 PM	PUBLIC	add a useful PROC MEANS

Version 2 | Version 2 compared with version 1 | Blame

Line	Author	Version	Content
51	PUBLIC	1	PSUStrata
52	PUBLIC	1	RATWeight;
53	PUBLIC	1	run;
54	PUBLIC	1	
55	PUBLIC	2	title "Summary stats for the Hazmat data";
56	PUBLIC	1	/* Summary stats for this data */
57	PUBLIC	1	proc means data=hazmat;
58	PUBLIC	1	run;
59	PUBLIC	1	
60	PUBLIC	1	title "Box plot of Vehicles # per incident";
61	PUBLIC	1	ods graphics / height=200 width=450;
62	PUBLIC	1	proc sgplot data=hazmat;
63	PUBLIC	1	hbox vehiclenumber;
64	PUBLIC	1	label VehicleNumber="# of vehicles";
65	PUBLIC	1	xaxis labelattrs=(size=12) valueattrs=(size=12);
66	PUBLIC	1	run;



# DATA step debugger



The screenshot displays the SAS DATA Step Debugger interface. The main window shows the following SAS code:

```
1 data results1 / ldebug;  
2 set sashelp.class;  
3 /* WHERE applied at compile time */  
4 /* Processes ONLY matching obs */  
5 where sex='M';  
6 run;
```

The second line, `set sashelp.class;`, is highlighted in yellow, and a yellow arrow points to it. The right panel shows a table of variables and their values:

Variable	Value	Watch
Name		<input type="checkbox"/>
Sex		<input type="checkbox"/>
Age	.	<input type="checkbox"/>
Height	.	<input type="checkbox"/>
Weight	.	<input type="checkbox"/>
_ERROR_	0	<input type="checkbox"/>
N_	1	<input type="checkbox"/>

The bottom panel shows the Debug Console, which is currently empty.



Find more at [the SAS Dummy blog](#)  
merci beaucoup

sas.com