GLOBAL attempt a name macro variable that exists in a local environment

Asked 2 years, 4 months ago Active 2 years, 4 months ago Viewed 696 times

0

Good morning everyone. I am trying to run the following code:

```
%macro clearALL;
/*delete macro variables*/

data delete;
set sashelp.vmacro;
where scope eq: 'F' and name ne: 'SYS';
run;

data _null_;
set delete;
call symdel(name) ;
run;
%mend clearALL;

%Macro Test;
%do i = 1 %to 18;
%put &= Iteration : &i;
    data _null_;
    set work.input;
         if _n_ eq &i then do;
         call symput('TipoDocumento',Tipo_Id);
         call symput('NroDocumento',Numero_Id);
         call symput('Fecha_nacimiento',FechaNacimiento);
         call symput('TipoEvaluacion',TipoEvaluacion);
         call symput('ESP',ESP);
         call symput('NIE',NIE);
         call symput('NIE_Asociado',NIE_ASOCIADO);
         call symput('Estrato',ESTRATO);
         call symput('IdConsulta',IdConsulta);
         call symput('TipoRespuesta',TIPO_RESPUESTA);
    end;
    run;

 %include "D:/Data_SAS/Sguerrero/TestMomento1.sas";
 %clearALL;
 %end;
%Mend;
%Test;
```

The data test contains the variables that will be converted into macro variables to execute code in the script included with include sentence. This code uses these variables and creates global and local variables. The% clearALL macro eliminates the macro variables created in the previous iteration. But I receive the following error message:

ERROR:% GLOBAL attempt a name (CAUSAL) that exists in a local environment.

Why the error message if the macro variables were removed with the% clearALL macro?

Thaks for helping me

macros sas

asked Mar 27 '18 at 14:22
Sandra Guerrero
171 1 8

1 Are you sure CLEARALL is working where scope eq: 'F' and name ne: 'SYS'; I don't see any macro with names that begin with F. – data _null_ Mar 27 '18 at 14:51

@data_null_G specifies that the macro variable is stored in the global symbol table, even if the local symbol table exists. L specifies that the macro variable is stored in the most local symbol table that exists, which will be the global symbol table, if used outside a macro. F specifies that if the macro variable exists in any symbol table, CALL SYMPUTX uses the version in the most local symbol table in which it exists. If the macro variable does not exist, CALL SYMPUTX stores the variable in the most local symbol table. – Sandra Guerrero Mar 27 '18 at 15:14

I'm very confused. Can you update your code so that it actually generates the error? I think the part that is generating the error must be in the TestMomento1.sas, so we can't see it. I'm confused by the intent of %ClearALL. If it is attempting to delete global macro variables it should be scope="GLOBAL". The only way scope eq: "F" could be true is if you have local macro variables, in a macro that starts with F. But you can't delete local macro variables. The CALL SYMPUTX arguments "F" "G" L" are not stored in the sashelp.vmacro. – Quentin Mar 27 '18 at 16:09

3 Answers

Active Oldest Votes

2

Be explicit when creating the macro variables whether they are Local or Global, Replace all your `Symput()` with `Symputx()` .

Example for Local:

```
call symputx('Estrato',ESTRATO,'L');
```

Example for Global:

```
call symputx('Estrato',ESTRATO,'G');
```

SymputX Documentation

answered Mar 27 '18 at 15:00
momo1644
1,697 6 24

0

Try adding `%SYSMSTORECLEAR;` to the beginning of your code. This will close the stored compiled macro catalog in case that was causing the error.

answered Mar 27 '18 at 14:55
momo1644
1,697 6 24

1 The `%sysmstoreclear;` macro statement is for when you are using a stored macro catalog. It has nothing to do with macro variables. – Tom Mar 27 '18 at 17:25

0

`call symput()` will create a new macro variable if the one you specified doesn't already exist. If you don't tell SAS where to create the macro variable (which requires using `call symputx()` instead of `call symput()`) then it will make the new macro variables in the current macro's scope. If you then later attempt to create a global macro variable with the same name using the `%global` statement you will get that error message.

You could just define the macro variables as global before trying to assign a value to the it.

```
%global TipoDocumento ;
```

Note that if you switch to using `call symputx()` you can tell SAS to make the macro variable in the global symbol table.

```
call symputx('TipoDocumento',Tipo_Id,'g');
```

answered Mar 27 '18 at 17:34
Tom
31.9k 2 10 22