

## Creating MACRO Catalogs in SAS/EG or SAS STUDIO Without an Unusual ERROR in the LOG

Updated on 2/26/2020,

I typically save my SAS® Macros in compiled MACRO CATALOGS stored in one or more directory locations. I generate the compiled macros using SAS code such as this, for example:

```
options nocenter mprint symbolgen compress=binary fullstimer
      mstored sasmstore=macin source2;

libname macin ('/your location/MACROS/');

/* Macro to use in ODS EXCEL to close the current spreadsheet tab and/or move onto another tab */
%macro BREAK / store source des="New Tab in xlsx file";
ods EXCEL options(sheet_interval="output");
ods exclude all;
data _null;
  declare odsout obj();
run;
ods select all;
%mend;

proc catalog catalog=macin.sasmacr;
  contents;
run;
```

The **MSTORED** option in the **OPTIONS** statement enables the developer to store the compiled macro in the directory specified by the **SASMSTORE=** option and the **LIBNAME** statement that defines where the catalog exists.

I have each macro in a separate code that compiles the macro via the **STORE** macro option and stores it in a single catalog called **sasmacr.sas7cat** in the specified libname. If I am using an interactive SAS platform such as SAS/EG or SAS STUDIO and run more than one code that contains the **SASMSTORE=** option and specified **LIBNAME** I get the program to work but I do get the following **ERROR** in the LOG.

```
ERROR: Unable to clear or re-assign the library MACIN because it is still in use.
ERROR: Error in the LIBNAME statement.
```

I don't like having **ERROR** statements in my code, even if it works. With SAS Studio and SAS/EG you can hit the Function Key "f9" which can be annoying. I finally found the solution from <https://support.sas.com/en/support-home.html> which pointed me to the solution which requires me to add the built in SAS MACRO **SYSMSTORECLEAR** before the **LIBNAME** as shown here:

```
%SYSMSTORECLEAR;
libname macin ('/your location /MACROS/');
```

### **More info on the full code on page 1:**

- **STORE** complies the macro.
- The **SOURCE** option allows the user to recover the code if the code is accidentally deleted. To recover the code. Use this code

```
%copy BREAK / SOURCE OUT='/ your location /MACROS/BREAK_recoverd.sas';
```

- The **DES** option for the macro compiling code adds a description to the compile macro
- The PROC CATALOG code with the option CONTENTS prints out the list of all macros in the specific catalog.

<b>Contents of Catalog MACIN.SASMACR</b>					
<b>#</b>	<b>Name</b>	<b>Type</b>	<b>Create Date</b>	<b>Modified Date</b>	<b>Description</b>
<b>1</b>	BREAK	MACRO	10/28/2019 17:12:39	10/28/2019 17:12:39	New Tab in xlsx file
<b>2</b>	IMPORT	MACRO	10/28/2019 16:04:24	10/28/2019 16:04:24	Import an xls file

When you run other code that you want to call the macros in the catalog without having to the hard code the macro in the program, use these options:

```
options nocenter mprint symbolgen compress=binary fullstimer  
mstored sasstore=macin source2;
```

```
%SYSSTORECLEAR;  
libname macin ('/your location/MACROS/') ACCESS=READONLY ;
```

You can include multiple LIBNAME locations if you have compiled macros in different directories. Here is an example:

```
libname macin ('/your location/MACROS/',  
              '/your location/MODELS/') ACCESS=READONLY ;
```

### **References:**

- **Carpenter, A.L.** Building and Using Macro Libraries.  
<https://support.sas.com/resources/papers/proceedings/proceedings/sugi27/p017-27.pdf>
- **Haden, L.** PROC CATALOG, the Wish Book SAS® Procedure.  
<https://support.sas.com/resources/papers/proceedings15/3459-2015.pdf>
- **Weaver, G. (2014)** Macros to Manage your Macros?  
[https://www.lexjansen.com/wuss/2014/72\\_Final\\_Paper\\_PDF.pdf](https://www.lexjansen.com/wuss/2014/72_Final_Paper_PDF.pdf)