

## CC08

# Creating a Stored Macro Facility in 10 Minutes

Erik S. Larsen, Independent Consultant, Charleston, SC

### Abstract

Every SAS® macro developer from time to time has spent countless hours writing, testing and perfecting code so that it can be used over and over by different users on different platforms. There are certain situations where one needs to enhance the speed of the processing of SAS code, such as in a production environment or if there is a need to process large volumes of data. A SAS Stored Macro Facility consists of a library of SAS macros which are already compiled, that are available to users or processes at any time. These pre-compiled macros run immediately when called and do not use valuable CPU time while waiting for compilation. Some examples of where a stored macro facility would be useful are in production environments, where jobs may be scheduled to run on a certain day of the day, week, or month. Another would be a standard report which is being used for an FDA submission and needs to be in an exact format. Creating a stored macro facility can be a beneficial tool for a developer or end-user and it is fairly easy to set up on any platform. There are also options to store the source code along with the compiled macro. This paper will give a brief introduction to the SAS Stored Macro Facility and explain how to set it up and use it to its full potential.

### The Basics

First of all, a clarification on the definition of a SAS stored compiled macro is not to be confused with a SAS autocall library. A SAS autocall library is simply a location on a LAN, directory in UNIX or a partitioned dataset on MVS which stores macros which can be called by programs if the autocall library is specified in the OPTIONS statement. These macros are not compiled and typically can be viewed in their entirety in their appropriate directory (unless these files are protected by some form of security). The SAS stored compiled macro library contains permanently **compiled** macros, and may or may not have source code stored in the same directory. The compiled macros are not readable to the user, as with source code but can be seen in a SAS macro catalog called SASMACR. If the user desires to allow access to the source code, an option can be set to allow the user to view the source code.

### Why Compiled?

There are many reasons to store compiled macros in SAS. The reason most frequently used is to decrease the overhead processing time. When a typical macro is written in SAS and submitted in batch or through the SAS program editor, CPU time is needed to check the macro for syntax, set up temporary macro facility variables and to compile the macro. With a SAS stored compiled macro, these steps have already been executed to give the CPU a jump-start on running the macro. Another reason for using the stored compiled macro facility is for uniformity. This allows all users to run the same macro so that results are consistent, such as an FDA statistical appendix or report. These macros are compiled once and when multiple users are accessing the macro, there is no duplication of compilation necessary. The final advantage of the stored compiled macro is the security of the facility. Users can be 'locked out' so that they are not able to modify the macro and thus produce reports or analyses which are not consistent with others. Cases of "versionitis" have caused pain in many SAS developers' lives. This technique can help control the suffering that slightly different copies of code can cause in a department.

### OK...enough background, how does it happen?

Set up of a SAS stored macro library is simple. If users understand macros, they can easily set up a stored macro library. An example below shows what is needed to set up the SAS stored macro library.

```

OPTIONS MSTORED SASMSTORE=StoreMac;
LIBNAME StoreMac ‘/user/id/sasmac/stored/’;

%MACRO ANALYSIS (DATA=, DEPVAR=, etc.) / STORE;

    PROC GLM DATA=<data>;
        . . .
    RUN;

    /* More code, etc. */

%MEND ANALYSIS;

```

In the above example, the SAS **OPTIONS** statement has the keywords **MSTORED** and **SASMSTORE**, which tell the SAS system to create a SAS stored macro library in the library referenced by the **LIBNAME StoreMac**. In the macro definition for **ANALYSIS**, there is a **/STORE** at the end of the macro which tells SAS to save the compiled macro in the above referenced **LIBNAME Store**. A **SASMCR** subdirectory is created in the **StoreMac** library reference and the compiled macro **ANALYSIS** is stored there. This directory can be viewed in the SAS Explorer window however, the user will not be able to see any code in this location unless the **SOURCE** keyword is used, as in the example below:

```

%MACRO ANALYSIS (DATA=, DEPVAR=, etc.) / STORE SOURCE;

```

This will store a copy of the source code in the **StoreMac** library reference. This code will be viewable to the user unless steps are taken with security to prevent the user from seeing it. It is useful to allow users to see the code so that they can see how it works, which parameters are available, and how to use the macro. Often times it is not desirable to allow users to see the code if it is proprietary or it contains information that it is not desired for outside users to see.

Another useful keyword is **DES=** which allows the developer to specify a detailed description of the macro. Another example is below:

```

%MACRO ANALYSIS (DATA=, DEPVAR=, etc.) / STORE DES='Table 1.1 Analysis';

```

The label “**Table 1.1 Analysis**” will be visible in the **SASMCR** subdirectory of the SAS stored macro library. This is useful when there are several stored macros in the library. This label will only identify the macro and unless the **SOURCE** option is present, the user will not be able to view the macro code.

## Accessing the Library

It is simple to access a SAS stored compiled macro library. It can be accessed by simply submitting the appropriate libname and options statement that tells SAS where the compiled macros are stored.

```

OPTIONS MSTORED SASMSTORE=StoreMac;
LIBNAME StoreMac ‘/user/id/sasmac/stored/’;

```

Once this code is submitted, the user can simply call a stored macro as if it had been already submitted (or included) in the session or batch job.

## Conclusion

The above examples have shown that it does not take much effort to set up a SAS stored compiled macro facility. It can be quite useful in a production environment and any repeated reporting and analyses that are required in a work environment. The SAS stored compiled macro facility differs from an autocall macro library in that the code is already compiled and can save a significant amount of time when processing large volumes of data. The SAS stored compiled macro facility can also be easily ported from different operating systems.

## References

Carpenter, Arthur L. (2004), *Carpenter's Complete Guide to the SAS Macro Language, Second Edition*, SAS Institute Inc., Cary NC.

SAS Institute, Inc. (2007), *Online Documentation, Version 9.1.3*, SAS Institute, Inc., Cary, NC.

Stojanovic, Mirjana, Watson, Dorothy, and Hollis, Donna, "A Practical Approach to the Stored Compiled Macro Facility in a Clinical Trial Environment," *Proceedings from SAS Global Forum 2006*, paper AP10\_06.

## Author Contact Information

Erik S. Larsen  
Independent Consultant  
Post Office Box 505  
Charleston, South Carolina 29402-0505

A SAS Alliance Consulting Partner, Affiliate Member  
A SAS Certified Professional V8

(703) 568-3083 (work)  
Larsen770@aol.com.

SAS® is a registered trademark of SAS Institute, Inc., in the USA and other countries.

® indicates USA registration.