

Capturing SAS Macro Code into an Executable SAS Program

David J. Jemiolo, DataCeutics, Inc., Pottstown, PA

ABSTRACT

Every day SAS users utilize SAS macros developed by other programmers or statisticians. There are often times when it is useful to store the resolved executable code produced by a macro, either for reference purposes or for later re-execution. For users in the pharmaceuticals industry it may be necessary to deliver 'standalone' executable SAS programs to regulatory agencies. Another situation is where a SAS program, utilizing several SAS macros, developed years earlier, needs to be updated. Storing and modifying the resolved SAS code may be an easier process than updating the original macro. This paper describes a method for capturing, formatting and storing resolved 'mprint' SAS code from a program invoking SAS macros.

INTRODUCTION

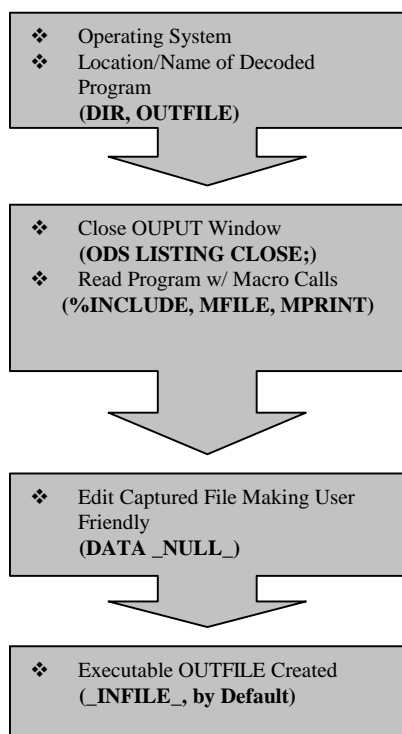
The purpose of this paper is to describe the SAS macro tool called `GENERATE_MACRO_CODE`, as well as the approach taken in its development. This SAS macro creates a fully executable, user-friendly, SAS program that is based upon an original SAS program containing one or many SAS Macro calls. This SAS macro is simple to use; the user does not need to know the details of the SAS program or the SAS macros that are referenced in the SAS program.

CONSIDERATIONS

In order for `GENERATE_MACRO_CODE` to be a successful tool, it needed to be flexible, simple to use, and of course accurate. There were a number of factors addressed during the development of `GENERATE_MACRO_CODE`. These factors included:

- ❖ Determine Operating System
- ❖ Assign Location, Name of Decoded Program
- ❖ Utilities to Capture Code, Control Output
- ❖ Format to a User Friendly Appearance

GENERATE_MACRO_CODE Process Flow



DETERMINE OPERATING SYSTEM

The first step executed in this macro is the determination of the operating system. The operating system is determined by implementing the SAS system value `&sysscp`:

```
%let del=/; ***UNIX;
%if &sysscp=WIN %then %let del=\; ***Windows;
```

ASSIGN LOCATION, NAME OF DECODED PROGRAM

The next step executed in this macro is the assignment of the name and directory location of the soon to be created decoded program file:

```
%let noval=;
%if &outfile = &noval %then
  %let outfile = &_amp;infile._;
%if &dir = &noval %then
  %let dir = &del.users&del.jemiolod;
***location predefined by user;
```

If an output file name is not passed as a parameter value into `GENERATE_MACRO_CODE`, the default will be the original SAS file name with "underscores" on either side. Similarly, if an output directory name is not passed as a parameter value into `GENERATE_MACRO_CODE`, the default will be the location predefined by the user.

UTILITIES TO CAPTURE CODE, CONTROL OUTPUT

The main tools that will be used in this code capture are the `MFILE` and `MPRINT` utilities in conjunction with the `%INCLUDE` command. But before we implement these tools, it is very important to close the ODS listing. This is necessary so that any output that may be generated by the original SAS program is not sent to the output window.

Although `MPRINT` and `MFILE` are very useful in capturing SAS Macro code, they still lack the ability to directly capture non-macro SAS code. The "trick" in `GENERATE_MACRO_CODE`'s approach is to enclose the `MPRINT`, `MFILE` option, and `%INCLUDE` statements within a dummy macro called `%_RUNIT_`. When `%_RUNIT_` is called, not only will the decoded SAS Macro calls be captured, but any other non-macro code that may exist in the original SAS program, will also be retained. All captured code is written to a text file named `_temp_`.

Once `%_RUNIT_` has been called, the ODS listing must be opened, else all subsequent tables, listings, etc., will never show up in the output window:

```
ods listing close;
%macro _runit_

  filename incode "&dir.&del.&infile..sas"
  lrecl=2048;
  filename mprint "_temp_" lrecl=2048;

  options mfile mprint;

  %include incode;

  options nomprint;

%mend _runit_;

%_runit_;
ods listing;
```

FORMAT TO A USER FRIENDLY APPEARANCE

Now that a text file (i.e., `_temp_`) containing all the decoded SAS Macro calls and non-Macro code from the original SAS program has been created, `_temp_` must be read in and formatted into a user-friendly presentation, and outputted to a new SAS program ready for execution.

The text file produced is all left aligned with a lack of hard returns between PROCs and DATA steps. To resolve these issues, a series of key words (i.e., PROC, DATA, RUN, and QUIT) are searched for. If these key words are encountered at the beginning of a line of text (not beginning with an `"**"`), a blank line is placed either before or after the line, as appropriate. Conversely, if these key words are not encountered in a line of text (not beginning with an `"**"`), then the line is indented by three spaces. It should be noted that each company has its own standards and tendencies and as a result, the list of key words should be changed appropriately to reflect these.

All of these formatting manipulations are made within a DATA `_NULL_` step. Once all text manipulations are complete, the resulting text are placed into a user-friendly output file which is either user defined (via Macro parameter) or defaulted to the name of the original SAS program, enclosed by "underscores". This program is fully executable and will produce an identical output as the original SAS program.

CALL PARAMETERS/SPECIFICATIONS

```
%generate_macro_code(dir =
                    infile =
                    outfile =);
```

`dir =` Not required, default = `/users/jemiolod`
(NOTE: Default can be pre-defined by user)
Location of the original SAS program and where the new SAS program will be located. The DIR structure can be that of either UNIX or WINDOWS.

`infile=` Required, no default.
Name of the original SAS program containing SAS Macro calls. INFILE is located in the passed parameter DIR.

`outfile=` Not required, default = `_&infile_`
Name of the SAS program, created and outputted by `%generate_macro_code`, containing the SAS Macro code based upon the original SAS program (i.e., INFILE). OUTFILE is located in the passed parameter DIR.
SAS program.

CONCLUSION

Although MPRINT and MFILE are very useful in capturing SAS Macro code, they still have two major limitations:

- ❖ Inability to directly capture non-macro SAS code
- ❖ Lack of a user-friendly presentation of captured SAS code

GENERATE_MACRO_CODE resolves these limitations making it a successful SAS macro tool used to create a fully executable, user-friendly SAS program based upon an original SAS program containing one or many SAS Macro calls. The tool is Macro based, which permits flexibility. Additionally, GENERATE_MACRO_CODE is simple to use, as it has only three call parameters. Script files can easily be created in order to apply this tool to entire SAS program directories.

ACKNOWLEDGMENTS

Paul Gilbert
Steven Light
Greg Weber
DataCeutics, Inc.

CONTACT INFORMATION

Your comments and questions are valued and encouraged.
Contact the author at:

David J. Jemiolo
DataCeutics, Inc.
1610 Medical Drive, Suite 300
Pottstown, PA 19464
Work Phone: 610.970.2333
Fax: 610.970.4884
Email: jemiolod@dataceutics.com
Web: www.dataceutics.com

