Good Programming Practice – Headers

Program Header

A standard header should be used for every program. The purpose of the header is to identify the program and provide documentation including revision history. It provides the necessary information to a code reviewer to identify and understand the program and its development life cycle. Standardizing the header will allow the information contained in the header to be leveraged programmatically for things such as auditing, project documentation, macro and dataset use tracking, consistency checking, and revision history reporting. There are a number of existing version control systems available (see appendix A).

The template below presents an example for a SAS program, but the basic elements needed in a good header are generalizable to most types of programs. The elements included in a header will vary from organization to organization, but there are certain pieces of information which are recommended to be included.

Highly Recommended

- Identification of the project of which the program is a part
- Program name
- Author identification which should be human readable and unique
- Date - convention used in different organizations will vary, so be clear as to what this date represents, for example date program initiated, date program passed final validation prior to database lock, etc.
- Revision History

Recommended:

- Brief description of the purpose of the program
- All outputs generated by the program, including creation of files and/or modifications to files

Optional:

- External files used such as datasets or databases that are used as data inputs to the program or macros used
- Additional information for users which may include such things as usage instructions, macro parameters, or sample macro calls
- Platform and \operating system which the program was developed to run in
- Software/programming language which the program was programmed in
- Copyright information

```
|**********************************************************************;
* Project           : Sample Drug, Sample Indication,Study1
*
* Program name      : sample.sas
*
* Author            : smithb
*
* Date created      : 20100816
*
* Purpose           : Summarize demographics data for the study.
*
* Revision History  :
*
* Date        Author      Ref    Revision (Date in YYYYMMDD format)
* 20100818    smithb      1      Removed subjects with who have not been dosed per spec.
*
|**********************************************************************;
```

In addition to your name or initials, use a unique identifier such as your login ID to identify yourself in the header. This is so there is no ambiguity on the identity of each programmer.

Update the revision history at each code modification made after the finalization of the first version of a program.

When a program is copied from another project, the person copying the program becomes the author and the revision history should be cleared. The origin of the program can be specified in the “Template” section of the header.

Revision History

The revision history section is critical to document the revisions made to the program. It should be updated each time the code is revised after finalization of the first version of a program. The Ref column in the revision history section is used to provide a reference to the location in the code where revisions were made. Here is an example:

```
* Revision History  :
*
* Date        Author      Ref    Revision (Date in YYYYMMDD format)
* 20100818    smithb      1      Removed subjects with who have not been dosed per spec.
```

The searchable reference number #1 would appear in the code where the corresponding revisions were made:

```
proc sort data=library.dm(where=(rfstdtc ne '')) out=dm; *** [1] ***;
  by usubjid;
run;
```

If code is removed, the reference should be placed where the code was removed from and if revisions are later removed, the original reference should remain in the code along with the new reference indicating that the previous revision was removed.

Special Sections for Scripts and Macros

The template above is for stand-alone programs, however additional sections may be needed for other types of programs. For example at SAS macro or a Unix script will require descriptions of parameters, along with their default values, used by the script or macro as well as example call statements.

Here is an example for a SAS macro:

```
* Parameters        :
*   INDATA - input dataset name
*   OUTDATA - output dataset name
*   WHERE - for subset the input dataset
*   MEANVAR - variable for statistical calculation
*
* Macro Dependency: (List those macros needed to be called prior to this macro)
*
* Example call      :
* %stat(indata=im, outdata=sec1, meanvar=lesnum)
```