%SETUP – A Utility Macro to Build Directory Structures and a SAS Initialization Program

Guowei Wu, Merck & Co., Inc., West Point, PA

ABSTRACT
This paper presents a utility macro %setup, which creates standard directories for a clinical study and an associated SAS program (startup.sas) to initialize the programming environment. The startup.sas defines options, library references, and global macro variables for the input and output directory paths. Startup.sas is the key component to building portable analysis programs and to maintaining consistency within a study, or even across projects. %setup was developed using base SAS® under Microsoft Window XP professional. This paper is intended for users of all skill levels.

INTRODUCTION
Pharmaceutical companies electronically submit analysis data sets and key SAS analysis programs to the FDA as part of an NDA submission. It is very important that all analysis programs are able to run at FDA sites without a single change.

Traditionally, all the programs need to be modified so that they are able to work under new locations even if they have been validated in the test environment. All modifications should be documented for tracking purposes. This is a time consuming process and may cause errors. In order to avoid this problem, portable programming strategies have been developed (Cheng, 2004; Shen & Lu, 2004). The common idea of those strategies is that libnames and directory paths are not defined in any analysis programs.

In order to simplify the transition process from testing to production locations, and from sponsor sites to the FDA sites, this paper recommends building standard directories across studies, portable analysis programs, and a startup.sas to define all libnames and directory paths. With this idea, the reviewer at the FDA would only be required to change one line of code to redefine the root directory in one program (startup.sas). Further more, a %setup macro has been developed to create standard directories across all studies and a SAS initialization program which is associated with those directories for each study. Some advantages of using a macro to create standard directories and the associated startup.sas are consistent, maintainable, and portable across studies.

DIRECTORY STRUCTURE
FDA provides an example for NDA submission in its website (The example of standard directory structure may be downloaded from: http://www.fda.gov/cder/guidance/NDA_Example.htm). The directories in the example are for an entire NDA submission and only to the protocol level. The appropriate subdirectories under protocol number can be doc, data, macro, and program. Subdirectory program may include subdirectories log, table, and output. Thus, the following structures under the Case Report Tabulations (CRT) directory of a NDA submission may be used:

```
~\N123456\CRT\datasets
  |__ doc
  |__ macro
  |__ 101
    |__ doc
    |__ macro
    |__ data
    |__ Program
  |__ 102
    ...
    ...
  |__ ISE
  |__ ISS
```

The following sub-directories are included under CRT\datasets:

1. doc
   This is the location for all project related documents. Specifically, files that shared between multiple protocols should be stored here.
2. **macro**
   This directory stores all utility macros and analysis macros used for multiple protocols.

3. **101**
   This is the root directory for all files specific to a particular protocol (101). Subdirectories containing only protocol-specific files should be kept under this directory. These subdirectories include:
   - Doc
   - Data
   - Macro
   - Program
   
   In the submission to FDA, the subdirectories output (store SAS output files), table (store rtf tables) and log (location for SAS log files) should not be included since they are empty at the time of submission (see details in **Guidance for Industry** at: http://www.fda.gov/cber/gdlns/esubapp.pdf). However, those folders should be automatically created when FDA reviewer wants to run the submitted programs.

4. **102**
   If a project has multiple studies, one directory for each protocol is created. Each protocol has the same subfolders as 101 above.

5. **ISS or ISE**
   An integrated summary of safety (ISS) or integrated summary of efficacy (ISE) may be part of the submission. In general, the subdirectories below the ISS or ISE should be the same as the protocol structures.

It is essential to follow a standard directory structure for each development stage and each protocol in order to ensure consistent programming strategies and make an easy transition.

**SAS INITIALIZATION PROGRAM**

The first step for making a portable program is to not define path information within an analysis program. Instead, the librefs, filerefs, or global macro variables are used to identify the locations of the directory structure of interest. Thus, every input and output path in the analysis program are relative. As a result, even if a study is moved to another drive with the same or different directory structures, all the analysis programs still work without any modifications.

However, using this strategy, the analysis program does not work by itself. It must set up the input and output locations when the SAS system is started. Thus, an initialization program is required to make this strategy work. The content of an initialization program could be option statements, library references for source data, analysis data and validation data, and global macro variable definitions (such as directory for log files, output list files, output RTF files, and programs). Also, the startup.sas should be able to detect the log, table, and output folders. If those folders do not exist, they are created automatically when startup.sas is run. In addition to making programs that can be transferred between studies or platforms easily, use of an initialization program can help to produce uniform output and requires minimal effort for maintenance.

There are two ways to set up the SAS working environment when the system is started. The first way is to include startup.sas in the AUTOEXEC.SAS. By default whenever the SAS System is started, it searches for a program called AUTOEXEC.SAS in the ISASROOT directory, and when it is present, this program is automatically executed. This method is good for the situation that only one study is worked for a period of time. However, in many cases, a statistical programming analyst may work on multiple studies at the same time. In this case, including startup.sas in autoexec.sas may not be a good idea since each study has its own startup.sas. Second, startup.sas is open and submitted manually for each study when the SAS system is started. The method is preferred if multiple studies are worked at the same time.

**%SETUP MACRO**

When a study is assigned, the first thing is to create work directories for a study. The macro %setup has been developed to create standard directories based on a given root directory for the project and protocol number, and automatically creates a startup.sas related to this directory structure.

Three keyword parameters are included in %setup macro.

- **rootdir**: Required, directory path to the project (no space in the path name).
- **protocol**: Required, protocol number.
- **author**: Optional, name of programmer.

Appendix 1 provides the key component of the macro %setup. The DOS prompt (“md &dir”) is used to create directory in this macro. The limitation of this DOS prompt is that no space is allowed in the directory path name.

The macro %setup can be used easily. For example, the following call will create a testing environment under project PharmaSUG and protocol 101. Appendix 2 and 3 are the directory and startup.sas program created through the following call:
%setup(rootdir= C:\PharmaSUG, protocol= 101, author=Guowei Wu);

Appendix 2 shows 5 subdirectories created for the protocol 101. They are docs, data, macro, program, and validation. Appendix 3 gives the startup.sas for those directories.

Before the study is moved from one location to another new location, this macro should be called to create a new directory structure and a startup.sas. The new startup.sas defines all the library and macro variables using the same names except they are now to be assigned a new location. Thus, all analysis programs will work when they are moved to this new location.

CONCLUSION
By building standard directory structures and a startup program automatically, we can make sure that all structures, libnames, librefs, and options are consistent within all projects within a company. The global consistency of directory structures, libnames, and librefs will promote the sharing of programs among project teams, speed up peer review and validation of programs. The portable analysis programs can greatly enhance the quality and efficiency from testing to production, and eventually, to FDA sites.

REFERENCES
Cheng, W. 2004, Build Portable Structures and Programs for Electronic Regulatory Submissions, PharmaSUG.

ACKNOWLEDGEMENT
The author would like to thank Susan Neidlinger and Carissa Mazurick for reviewing the manuscript.

CONTACT INFORMATION
Your comments and questions are valued and encouraged. The author may be contacted at:

Guowei Wu, Ph.D
Merck & Co., Inc.
HM-209
P.O Box 4
West Point, PA 19486-0004
Phone: 215-328-2111
Email: Guowei_wu@merck.com

SAS and all other SAS Inc. product or service names are registered trademarks or trademarks of SAS institute Inc. in the USA and other countries. ® indicates USA registration.
APPENDIX

1. Key component of %setup macro

%macro setup(rootdir=, protocol=, author=);
  %put;  
  %put -----------------------------------------------------------------------------------;
  %put --- Start of %upcase(&sysmacroname) macro ;
  %put -----------------------------------------------------------------------------------;
  %put;
  options ls=140 ps=55 center nodate nonumber noxwait noxsync;
  %local rootdir protocol author;

  ** Error check, if the value for the required key parameter is not given, aborts the process;
  %if %length(&rootdir) = 0  or %length(&protocol) = 0 %then %do;
    %put **************ERROR MESSAGE ******************************************************;
    %put ************  Key parameter was not assigned ;
    %put **************ERROR MESSAGE **********************************************************;
    %goto MacroEnd;
  %end;

  %if %length(&author) = 0 %then %let author = ;

  ** This macro mkdir is used to check if a directory exists, if not, this directory will be
  created;
  %macro mkdir(dir=);
    %if %sysfunc(fileexist(&dir)) ^= 1 %then %do;
      %x "md &dir";
    %end;
  %mend mkdir;

  **Create Standard directories for a study if it does not exist;
  %mkdir(dir=&rootdir\&protocol\data);
  %mkdir(dir=&rootdir\&protocol\doc);
  %mkdir(dir=&rootdir\&protocol\macro);
  %mkdir(dir=&rootdir\&protocol\program);
  .
  .
  .

  More %mkdir calls. Through macro calls, directories under a protocol can be created;

  ** Check to see if startup.sas exists. If it exists, shows the warning message and go to the
  end of macro;
  %if %sysfunc(fileexist(&rootdir\&protocol\program\startup.sas)) %then %do;
    %put **************WARNING MESSAGE ******************************************************;
    %put ************  &rootdir\&protocol\program\STARTUP.SAS exists ;
    %put ************  The existing startup.sas was not overwritten ;
    %put **************WARNING MESSAGE ******************************************************;
    %goto MacroEnd;
  %end;
** if a startup.sas does not exist, then creates startup.sas program;

    proc printto print="&rootdir\&protocol\program\startup.sas" new;
    title;
**This data _NULL_ step is used to build startup.sas program;

    data _NULL__;
    x=1;
    file print;

    put '/************************************************************************
    put '*************************************************************************/,'#;
    put 'options ls=140 ps=55 center nodate nonumber mlogic mprint symbolgen;';
    put 'options missing ='' ''; 
    put /'****Define macro variables that represent standard directories *****';
    put '%let ' "protpath = &rootdir\&protocol';
    put '/'"%%let analdir = &protpath\data\analysis';

    run;

    proc printto;
    run;

    %MacroEnd: %put;
    %put --------------------------------------;
    %put --- End of %upcase(&sysmacroname) macro ;
    %put --------------------------------------;
    %put;
    %mend setup;
2. The directories created by `%setup` in the example call.

- PharmaSUG
  - 101
    - data
      - analysis
      - external
      - source
    - doc
      - macro
    - program
      - log
      - output
      - table
    - validation
      - data
      - log
      - output
    - doc
    - macro
3. Initialization program (startup.sas) created by %setup in the example call

```sas
/** Initialization program (startup.sas) created by %setup in the example call */

/* Initialization program (startup.sas) created by %setup in the example call */

PROCEDURE NAME: 101
PROGRAM NAME: STARTUP.SAS
AUTHOR: Guowei Wu
DATE: 20SEP2004
SAS VERSION: V.8.2
OPERATING SYSTEM: Win XP
PURPOSE: This program creates macro variables that represent standard directories
          In addition, it sets options and defines SAS librefs
INPUT FILE:NONE
COMMENT: This is the first program to be run before you run any other analysis
          programs

options ls=140 ps=55 center nodate nonumber mlogic mprint symbolgen;
options missing =' ';**Define macro variables that represent standard directories;
%let protpath = C:\PharmaSUG\101;
%let analdir = &protpath\data\analysis;
%let rawdir = &protpath\data\source;
%let extdir = &protpath\data\external;
%let macdir = &protpath\macro;
%let pgmdir = &protpath\program;
%let listdir = &protpath\program\output;
%let logdir = &protpath\program\log;
%let tabledir = &protpath\program\table;
%let valdir = &protpath\validation;
%let vlistdir = &protpath\validation\output;
%let vlogdir = &protpath\validation\log;

**Create folders output, table and log if they are not existed, if not, it will be created;
%macro ccdir(dir=);
   options noxwait noxsync;
   %if %sysfunc(fileexist(&dir)) ^= 1 %then %do;
      x "md &dir"
   %end;
%mend ccdir;
%ccdir(dir=&logdir);
%ccdir(dir=&listdir);
%ccdir(dir=&tabledir);

**Define Library names for analysis, raw, external, and validation data;
libname analdir "&analdir";
libname rawdir "&rawdir";
libname extdir "&extdir";
libname valdir "&valdir";

**Define path to the macros used in this study;
options sasautos=(&macdir, sasautos) mautosource;

**Include format.sas;
%inc "&pgmdir\format.sas";

******************************************************************************end of startup.sas*******************************************************************************/;
```