SAS Libraries and Permanent SAS Data Tables

When programming in SAS, data tables or data sets are created in a DATA step using either an INPUT or INFILE statement. These data tables are temporary. When the SAS session is closed, these data tables are gone too. We may have the data in an external text file and a copy of the program, but a SAS data table does not exist once SAS is closed. A SAS data table is a file that SAS recognizes and can read the contents without the use of a DATA statement.

So far in this course, the programming done in SAS required data tables to be created during the current SAS session. These data tables are temporarily stored in a library called WORK.

Objective 1: Run the following short program in SAS. Observe the contents of the LOG window.

```
DATA   ONE;
INPUT  X  Y   Z;
DATALINES;
25  27  34
28  31  29
41  58  29
37  28  83
RUN;
```

All that has been requested in the program is that a data table called ONE be created. After executing this program, examine the SAS Log window. You should observe the following:

```
NOTE: The data set WORK.ONE has 4 observations and 3 variables.
```

In the above program the data set was named ONE. SAS stores this data table in a library called WORK. The complete specification of the data set is WORK.ONE. SAS refers to this as the two-level data table name. In general, data sets are specified by libname.tablename where libname is the library name and tablename is the name of the data table (or set).

When data tables are large and reading in the data takes a long time, or when data tables are frequently used, a permanent SAS data set can be created. A SAS dataset is a file that SAS recognizes. SAS recognizes the variables, the values they take, and the labels assigned to the variables.

You can create a new library name for the purposes of saving your data table as a permanent SAS data table on disk or other media. The library names you create are not permanently recognized by SAS by default. Similar to data table names, once you create a library during a SAS session, it is active until you close SAS. When SAS closes, these library names that you have created will be erased. This keeps your SAS program from becoming too cluttered with library names. You will see an option later in these instructions for a method of keeping the library permanently recognized. (This option has been made available in SAS releases 8.0 and newer.)

Creating a Permanent SAS Data Table

To create a permanent SAS data table you need to use a LIBNAME statement in conjunction with a DATA step operation. The syntax of the LIBNAME statement is:

LIBNAME *libname* '*location of data table*' ;

where *libname* is the name we wish to call the library, and *location of data table* is the directory or folder in which the permanent SAS data table will be saved and read from in the future.

Objective 2: Create a SAS library named CLASS on the a:\ drive (or specify a drive:\folder for your use) and store the small 4 observation 3 variable data table in that location.

```
LIBNAME  class  'a:\' ;

DATA  class.demo;
INPUT  X  Y  Z;
DATALINES;
25  27  34
28  31  29
41  58  29
37  28  83
RUN;
QUIT;
```

Note the messages in the Log Window.

```
NOTE: Libref CLASS was successfully assigned as follows:
      Engine:        V9
      Physical Name: a:\

NOTE: The data set CLASS.DEMO has 4 observations and 3 variables.
```

Engine refers to the version of SAS that creates the data set. Each new version of SAS can read the data files created by older versions of SAS. More notes on this are in **How to Tell Which Libraries are Assigned (Active)** section. (V9 indicates that this objective was performed using SAS 9. Your version of SAS may differ.)

After running this program, check the contents of the a:\ drive. You should have a new file called demo.sas7bdat. The name we chose for our data table has now become the file name. The sas7bdat extension is assigned by SAS. (Earlier versions of SAS use an ssd or sd2 extension.) In general, the filename will have the form *tablename.sas7bdat.* Files with the sas7bdat extension cannot be opened in the program editor or Editor windows of SAS. The file demo.sas7bdat will remain in a:\ until you write over it or delete it.

Using a Permanent SAS Data Set

Within a SAS program, SAS data tables can be called into a procedure by their two-level names. For each procedure where the DATA = *tablename* option appears in the procedure (PROC) statement, the value of *tablename* is given by libname.setname. This is called the two-level SAS name.

Objective 3: Run the MEANS procedure on the demo data table created in an earlier SAS session. Upon opening SAS, one could run the following program without a DATA step.

```
LIBNAME  class  'a:\';

PROC MEANS DATA=class.demo;
VAR X Y Z;
RUN;
QUIT;
```

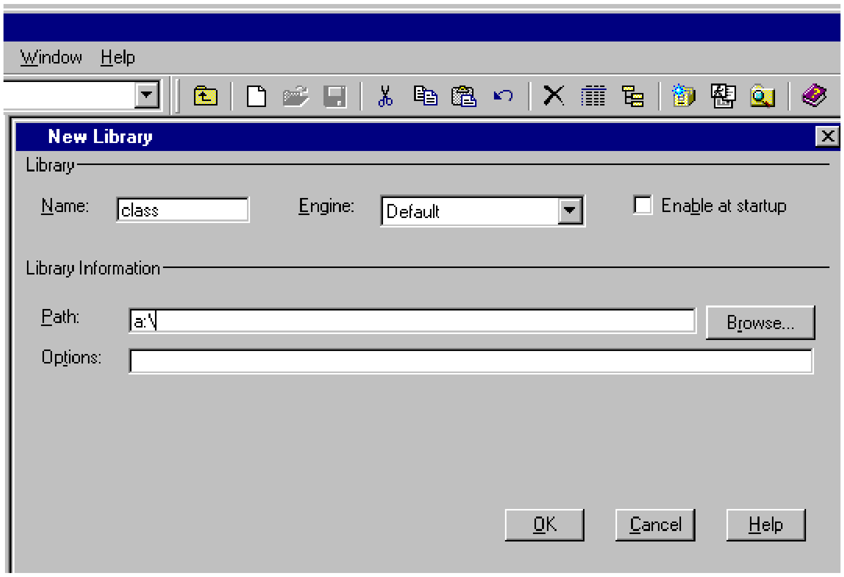
Notice this program runs without a DATA step. SAS recognizes the file demo.sas7bdat and its contents.

The assigned library name is only valid during the current SAS session even though the permanent SAS data tables exist on disk. If you are going to call permanent SAS data tables into a program, you will need to first define a library that contains that data table. One need not choose the same library name each time.

Here are a few ways to define or assign a library name.

1. LIBNAME statement – the syntax is shown above. You can create a library in a one-line program. In the Editor simply type in a complete LIBNAME command and submit the program. This new library will be active after the LIBNAME statement successfully executes. You will see the note in the SAS log confirmed the library assignment.

2. **Add New Library** button at the top of the screen (file drawer icon with a starburst on it). When using the **Add New Library** button, you be prompted to enter a library name and the path (the directory in which you wish the data table to be stored). The default engine is the version of SAS that you are running. Most users will leave engine set at Default. See Figure below.



New Library

Library

Name: class

Engine: Default

□ Enable at startup

Library Information

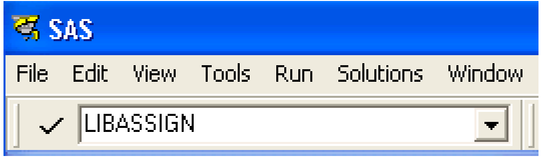
Path: a:\

Options:

You can click on the box for **Enable at startup** if you want SAS to automatically assign this library when SAS begins. You can reverse this action later by clicking on the box again to deselect the option. **Browse** allows you to peruse the available directories on your computer to assign as a library. This is helpful if you do not know the complete path specification where the data sets will be stored.

3. There is an Explorer window that appears at the left of the SAS screen. To view the Explorer window, click on the Explorer tab at the bottom left of the screen. You may have been closing this window in the past as you were writing programs. If so, then select from the pull-down menus **View – Explorer**. Move to the Explorer window in SAS, select **Libraries** and view **Active Libraries**. (Do not select any libraries here.) Select **File – New** from the pull-down menus and the New Library dialog box shown above will open.

4. Typing LIBASSIGN on the command line



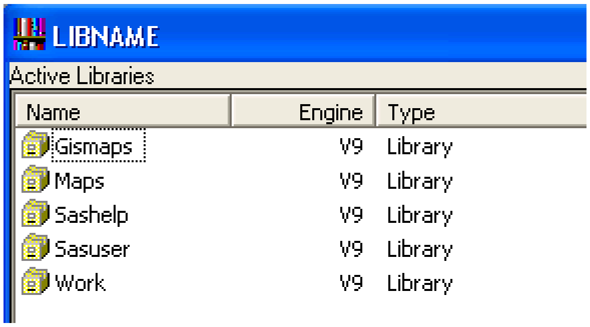
LIBASSIGN

in the upper left hand corner of the screen will also open the New Library dialog box.

How to Tell Which Libraries Are Assigned (Active)?

Since you could name several libraries and since SAS has a few libraries that are named and active in every SAS session, you can find out the names of the libraries and their locations in the following ways.

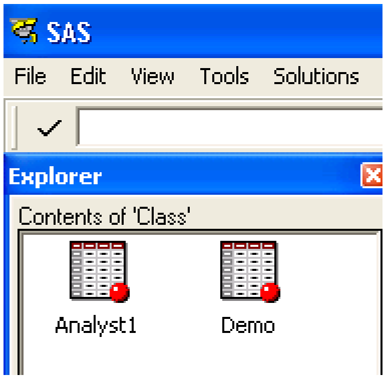
1. For all versions of SAS, type LIB on the command line. You will be shown a window that contains the active libraries, the version of SAS in which the files were created, and their



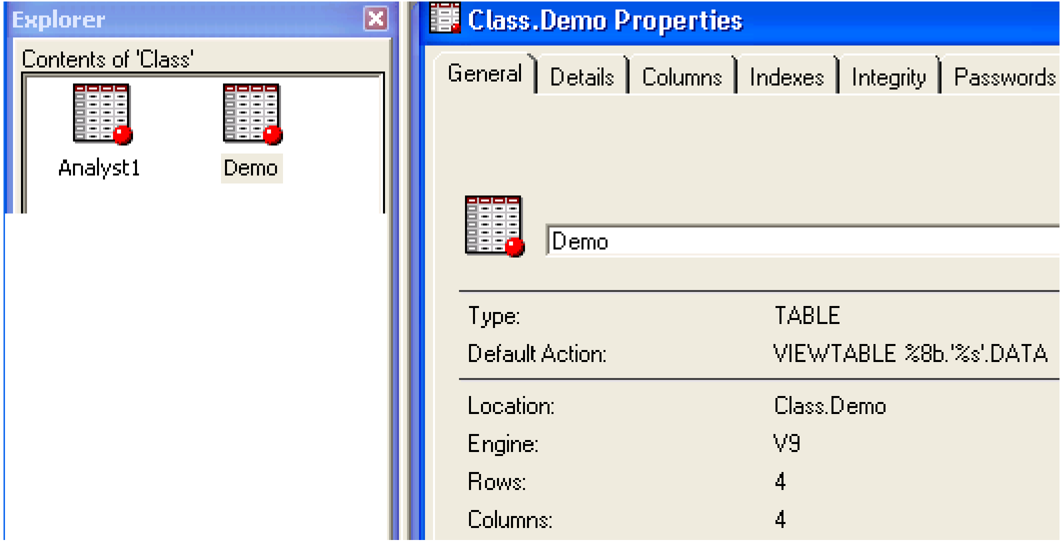
Name	Engine	Type
Gismaps	V9	Library
Maps	V9	Library
Sashelp	V9	Library
Sasuser	V9	Library
Work	V9	Library

locations (Host Path Name, not shown in picture). Clicking on the library icon will open the library and list the SAS data tables in that library.

2. Move to the Explorer window in SAS. If you may have been closed this window, then select from the pull-down menus **View – Explorer**. Once you can view the contents of the Explorer window, it will display Active Libraries. The Library CLASS that you have created during the earlier demonstration should appear in this list. To examine the contents of any of the libraries, double-click on the icon for that library.



The Explorer window does not indicate the engine type for these libraries. If you right click on the name of the data table, and select **Properties** from the list, the engine (or version of SAS) used to create the data set will be shown in the resulting table of information.



Demo

Type: TABLE
Default Action: VIEWTABLE %8b.'%s'.DATA

Location: Class.Demo
Engine: V9
Rows: 4
Columns: 4

To view the contents of any of the temporary or permanent data tables, double-click on the icon beside the data table name. This will open the **Viewtable** product within SAS and display your data table in a spreadsheet type of format. Before running a SAS program requiring this data set, you will need to close **Viewtable** by clicking on the X in the upper right hand corner of the Viewtable window. Be careful not to click on the X of the SAS window and thus exit SAS.

Objective 4: Learn how to use the LIBNAME statement by following the steps given below.
In the Editor type this line and submit it as a program. You must have a media storage device in the A drive for this objective to work, or specify a drive:\folder other than a:\.

```
LIBNAME  two   'a:\' ;
```

Examine the log window. You should see a message that the library name has been successfully assigned. The engine referred to in the log window is the version of SAS. For example:

Engine v608 refers to version 6.08 of SAS
Engine v611 refers to version 6.11 of SAS
Engine v8 refers to version 8 of SAS
Engine v9 refers to version 9 of SAS

Type LIB on the command line and note the addition of TWO to the list of libraries.

SAS is sensitive to the engine that created data sets you have saved or accumulated. When you create a library name for a folder or directory that contains files with an sd2 extension, for example, SAS will reply in the LOG window that the engine type is V6 rather than V7. You can force the engine type by including it in the LIBNAME statement. Extending this objective,

LIBNAME class1 V6 'a:\' ;	will recognize the data tables in a:\ that were created using version 6 of SAS. (Recognizes files with sd2 extension.)
LIBNAME class2 V8 'a:\'	will recognize the data tables in a:\ that were created using version 8 of SAS. (Recognizes files with sas7bdat extension.)

Note both of these libraries refer to the same drive:\folder, but each library contains different items. Also, SAS versions 7, 8, and 9 are nearly the same engine. All data files created in these three versions of SAS have sas7bdat extensions. SAS 9 allows for longer labels on variables.