

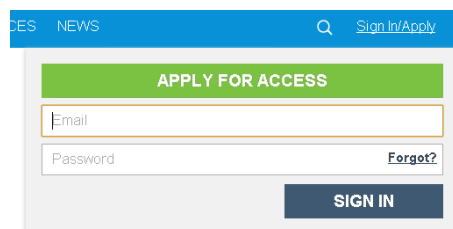
# Creating and Running SAS® Programs Using the Project Data Sphere® Online Service

## Introduction

This guide provides SAS programmers with the additional instructions necessary to write and run SAS programs using data on the *Project Data Sphere*® Online Service. This guide assumes no previous knowledge of the *Project Data Sphere*® Online Service or SAS® Drug Development. This guide does assume that the user has applied for and been granted access to the *Project Data Sphere*® Online Service.

To begin, access the [Project Data Sphere® Online Service](#). At the top right, there is a link to sign in. Click **Sign in** and enter User ID and Password.

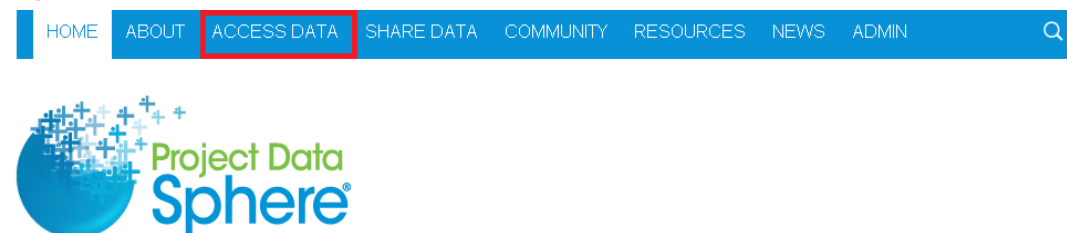
Figure 1: Sign In



The screenshot shows a web interface with a blue header containing 'DES NEWS' and a search icon. Below the header is a green button labeled 'APPLY FOR ACCESS'. Underneath are two input fields: 'Email' and 'Password'. A 'Forgot?' link is positioned to the right of the password field. At the bottom right of the form is a dark blue button labeled 'SIGN IN'.

Once signed in, from the home page, choose **ACCESS DATA**.

Figure 2: Select ACCESS DATA



From the [ACCESS DATA](#) link, under Analytics Tools, click the SAS® Drug Development icon.

Figure 3: Select SAS® Drug Development

### Analytics Tools

You can download data and use your own tools for analysis or you can access the data directly through the tools links provided below.

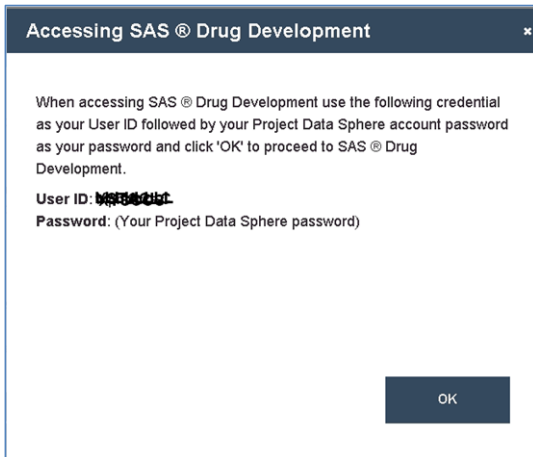


The information contained in this document is considered confidential and the *Project Data Sphere*® Online Service User Agreement prohibits your use or disclosure of any of this information in any manner that would jeopardize its confidentiality.

Copyright © 2015 SAS Institute Inc. All rights reserved. SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies.

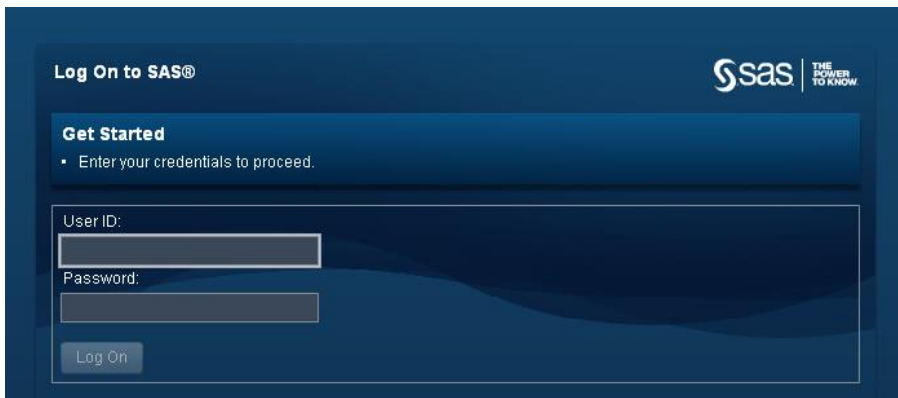
A pop-up message will provide directions regarding the User ID to be used. This information is blurred in Figure 4.

**Figure 4: User ID Information**



Click **OK**. SAS® Drug Development will automatically deploy in a new browser tab. If the new browser tab function is turned off your web browser, you may need to select an additional option to allow SAS® Drug Development to open. When the new browser tab opens, SAS® Drug Development starts, but it will be necessary to log on per the directions above.

**Figure 5: SAS® Drug Development Log On**



Enter User ID and Password and click the **Log On** button.

Once logged on, a dashboard will appear that includes a number of different windows that may not be familiar to new users of SAS® Drug Development. SAS® Drug Development provides a suite of tools and technologies necessary to manage clinical trials data in a professional, audited, qualified environment. It is designed for the management of data and execution of SAS programs implemented by the statistical teams that analyze real-world clinical trials data. The full functionality of this software is available on the *Project Data Sphere*® Online Service, but this guide will focus only on what you need to know to copy data and create and run a SAS program for individual research.

The three main tabs include the Dashboard (the default opening view), the Repository, and the Workspace.

**Figure 6: Dashboard, Repository, and Workspace**



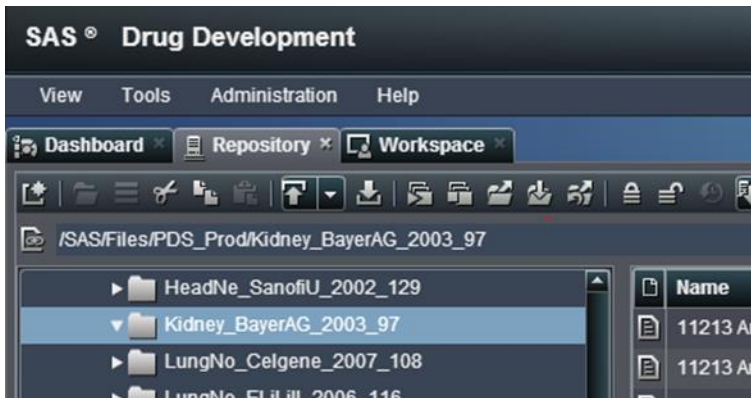
The Repository is the place where shared data are stored. The Workspace is a personal workspace where you can store your programs and any datasets you copied from the Repository to use. The Workspace provides a secure work area to conduct analyses without modifying the original dataset as it appears in the Repository. The following section of this guide describes how to use the Repository and Workspace.

## Accessing Data from the Repository and Setting up Workspace

To demonstrate this process, we will access a real data set from the Repository and make a copy in the Workspace. This example will use the kidney data uploaded by Bayer AG.

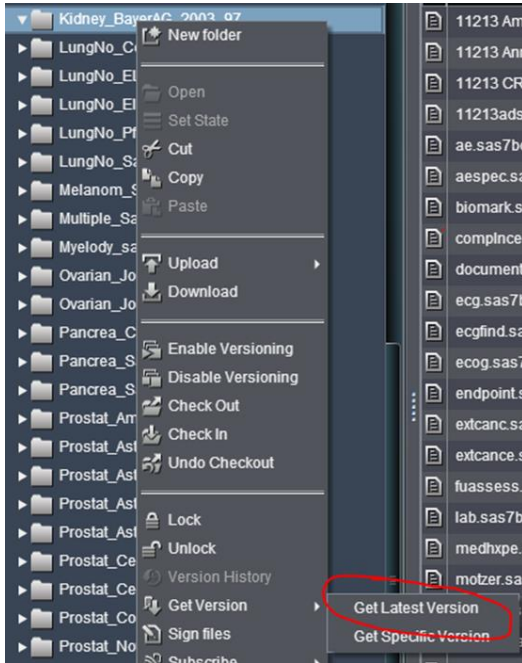
First, select the Repository tab, which will default to the SAS directory. Expand the SAS directory then expand Files and PDS\_Prod. Scroll to the **Kidney\_BayerAG\_2003\_97** subdirectory and highlight that directory.

**Figure 7: Selecting a study**



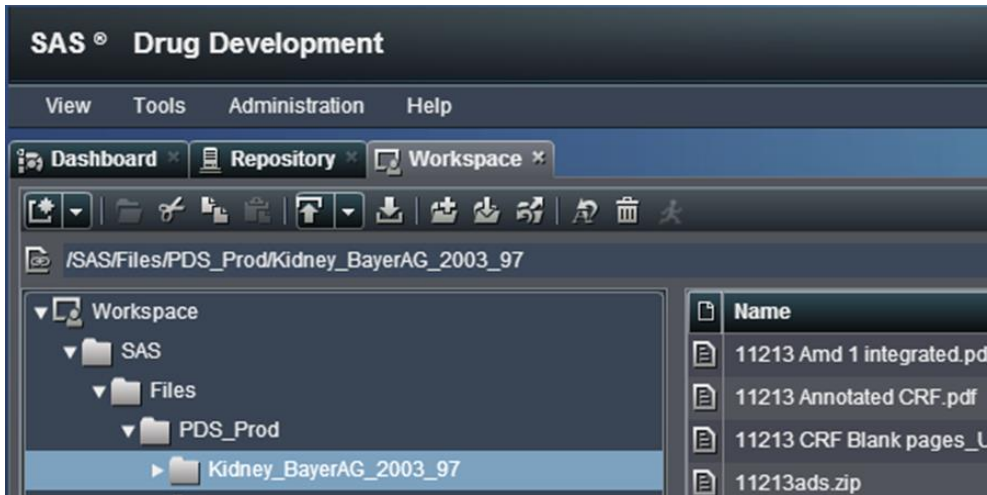
Once selected, click the right mouse button (RMB) and select **Get Version → Get Latest Version**.

**Figure 8: Get the latest version of the study**



When **Get Latest Version** is selected, the system copies the folder to the Workspace: *Workspace/SAS/Files/PDS\_Prod/Kidney\_BayerAG\_2003\_97*. Notice this has the same directory structure as the Repository. You may need to expand SAS/Files/PDS\_Prod to see the copied data files.

**Figure 9: Latest version of the study copied to the workspace**



Now that a copy of the selected dataset is in the Workspace, access that copy with a SAS program. SAS programmers know that a data directory should be accessed using a libname statement. But due to the need to maintain much tighter data security for clinical trial data, accessing a dataset within SAS® Drug Development requires an additional bit of code.

Behind the scenes, for each SAS session, SAS has reserved macro variables that contain the base values for User IDs and Workspaces. These variables are initialized for each session. In the SAS log of programs, there will be code similar to the following that defines a User and a Workspace:

---

The information contained in this document is considered confidential and the *Project Data Sphere®* Online Service User Agreement prohibits your use or disclosure of any of this information in any manner that would jeopardize its confidentiality.

**User ID:** `_SDDUSR_`

```
%let _SDDUSR_=%bquote(<User ID>);
```

**SAS workspace:** `_SASWS_`

```
%let _SASWS_ = %nrstr("/sddshared/SASWorkspaces/<User ID>");
```

There is not a need to understand all of the details of these processes, but the `_SASWS_` variable must be used in a libname statement to access a copy of any dataset in the Workspace in order to use SAS® Drug Development. This variable can be called like any macro variable with the addition of an ampersand to the variable name: `&_SASWS_`. This call to `&_SASWS_` must be used at the start of any libname reference within SAS® Drug Development.

The following code is all that is needed to run a simple frequency analysis of the `aegrade` variable in the `AE` dataset. Notice the highlighted portion of this line of code that shows where the `&_SASWS_` macro variable needs to be. This libname points to the Workspace area and the `Kidney` project dataset loaded in the previous example. Once the `&_SASWS_` macro is used in the libname statement, the libname can be used like any other libname.

```
libname proj "&_SASWS_/SAS/Files/PDS_Prod/Kidney_BayerAG_2003_97";

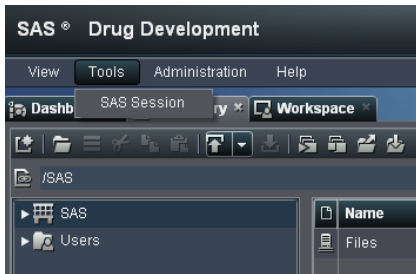
proc freq data=proj.ae;
  tables aegrade;
run;
```

To see how this works in practice, start a SAS session.

## Running a Simple SAS Program

To start a SAS session with the traditional CODE, LOG, and OUTPUT windows, select SAS SESSION from the TOOLS menu.

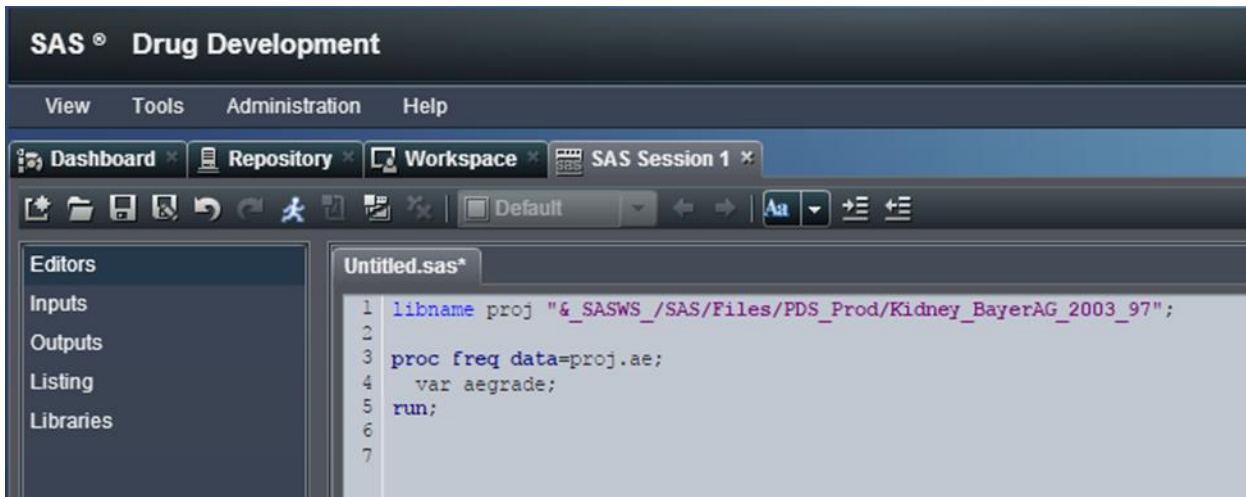
Figure 10: Starting a SAS session



This is the traditional SAS interface programmers know.

In the editor, start the session by typing in the libname statement using the macro variable `&_SASWS_` and point to the Workspace directory where the Kidney dataset was loaded.

Figure 11: Writing SAS code in the Editor




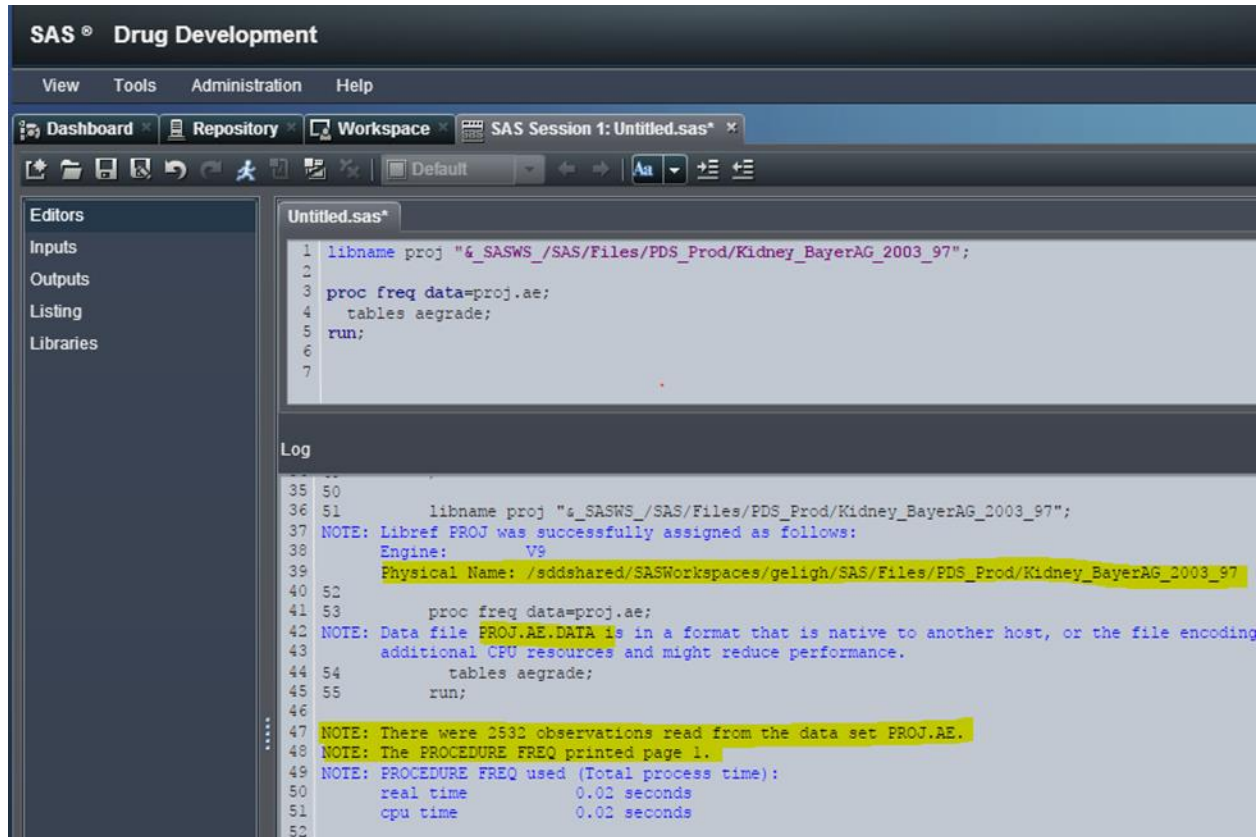
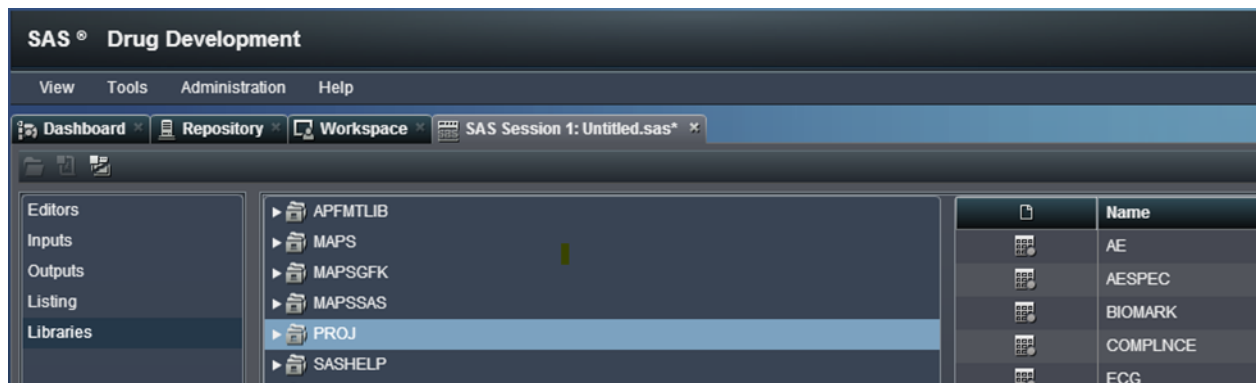
Submit the SAS session by clicking on the **run** icon . The code will run and verify the directory was correctly assigned, the proj.ae dataset was correctly accessed, and the frequency analysis ran successfully.

Figure 12: Reviewing the SAS log



By selecting **Libraries** in the left most pane, and then the **PROJ library**, the contents of the PROJ library will be available to the SAS session for browsing.

Figure 13: Accessing the PROJ1 library in the SAS session





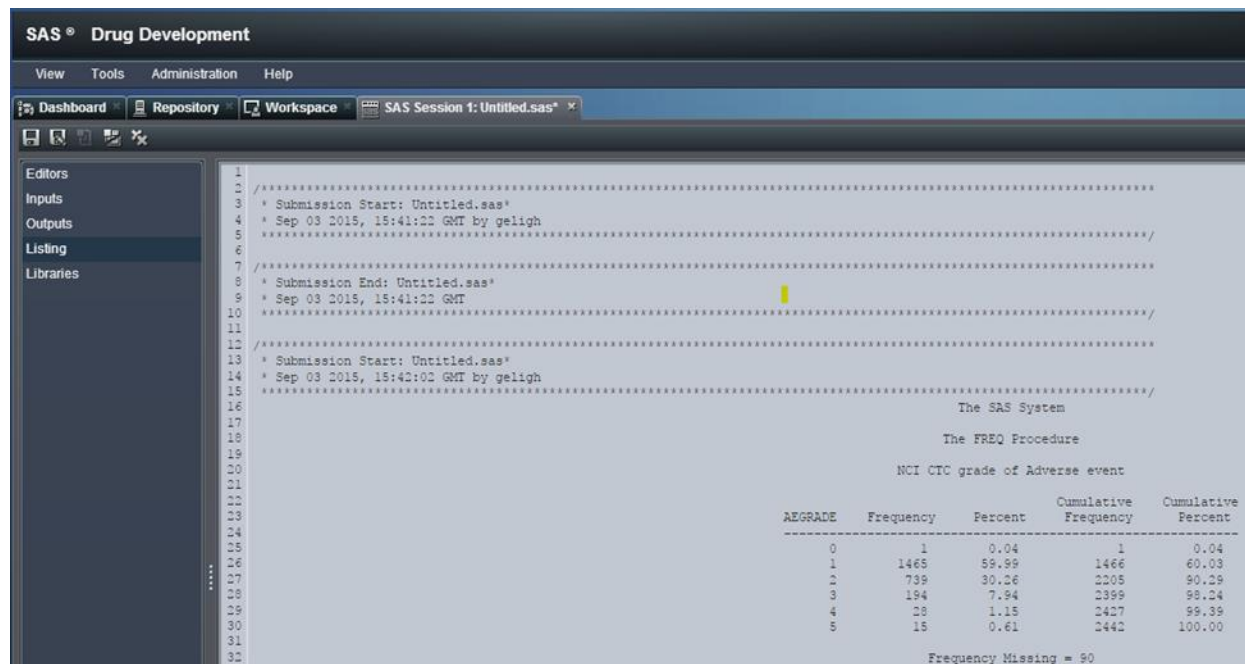
Selecting **Listing** in the left most pane will access the listing generated by the submitted code. The following figure is an example, but results may vary slightly. The listing and program can be saved using the Save icons   in the upper left panel.

Figure 14: Accessing the SAS listing



The information contained in this document is considered confidential and the *Project Data Sphere*® Online Service User Agreement prohibits your use or disclosure of any of this information in any manner that would jeopardize its confidentiality.



## A More Complex SAS Program

The example above used the Kidney\_BayerAG\_2003\_97 AE dataset to run a simple frequency analysis. Far more complex SAS programs can be developed and run. Consider the following SAS program written to access some of the Prostate Cancer datasets in the Repository. Annotated notes appear below the figure, referenced by number.

Figure 15: A more complex SAS program

```

1 *****
* ProstateProj_LB.sas *
* * *
* Generates and maps the LAB data set for the specified study *
*****;

2 libname proj "&_SASWS_/SAS/Files/PDS_Prod/Prostat_AmgenI_2006_121";
libname outfile "&_SASWS_/";

3 options nofmterr;

4 proc sort data=proj.albsaf out=lb1 (keep=STUDYID SUBJID LBSEQ LBTESTCD LBTEST LBORRES LBORRESN
LBORRESU LBDY rename=(subjid=usubjid));
by studyid usubjid lbseq;
run;

proc sort data=proj.aslbase out=dm (keep=STUDYID SUBJID rename=(subjid=usubjid));
by studyid subjid;
run;

*****
* Build the LB data set *
*****;

5 data LB1;
%include "&_SASWS_/prostate_attr.sas";
merge work.lb1 (in=ina) work.dm (in=inb);
%include "&_SASWS_/clear_lab_formats.sas";
by studyid usubjid;
if ina and ^inb then put 'Missing data from DM' usubjid=;
if ina;
study="Prostat AmgenI_2006_121";
dataset="ALBSAF ASLBASE";
sas_note='';
lbseq=.;
run;

6 proc sort data=LB1 out=outfile.amgeni2006_121_LB;
by studyid usubjid lbseq;
run;

```

### 1. A common header block

```

*****
* ProstateProj_LB.sas *
* * *
* Generates and maps the LAB data set for the specified study *
*****;

```

### 2. Libname statements that make use of the &\_SASWS\_ macro calls.

The &\_SASWS\_ macro variable is used. The value of this variable is /sddshared/SASWorkspaces/<Your User ID>. This value is generated automatically when the user starts a SAS session.

For **libname proj** the full directory is /sddshared/SASWorkspaces/<Your User ID>/SAS/Files/PDS\_Prod/Prostat\_AmgenI\_2006\_121.

For **libname outfile** it is /sddshared/SASWorkspaces/<Your User ID>.

```
libname proj "&_SASWS_/SAS/Files/PDS_Prod/Prostat_AmgenI_2006_121";
libname outfile "&_SASWS_/";
```

3. Nofmterr option used so any formatting in the dataset will be ignored.

```
options nofmterr;
```

4. From this point on, the copy of the dataset in the Workspace can be accessed using regular SAS code. This code sorts the datasets.

```
proc sort data=proj.albsaf out=lb1(keep=STUDYID SUBJID LBSEQ LBTESTCD
LBTEST LBORRES LBORRESN LBORRESU LBDY rename=(subjid=usubjid));
  by studyid usubjid lbseq;
run;
proc sort data=proj.aslbase out=dm(keep=STUDYID SUBJID
rename=(subjid=usubjid));
  by studyid subjid;
run;
```

5. For the %include, this example uses the \$\_SASWS\_ global macro variable the same way, it is a physical value to the Workspace. The prostate\_attr.sas program contains common attributes used across all Prostate datasets to standardize their values. The clear\_lab\_formats.sas is used to clear any associated format statements applied to the original dataset.

```
*****
* Build the LB data set *
*****;
data LB1;
  %include "&_SASWS_/prostate_attr.sas";
  merge work.lb1(in=ina) work.dm(in=inb);
  %include "&_SASWS_/clear_lab_formats.sas";
  by studyid usubjid;
  if ina and ^inb then put 'Missing data from DM' usubjid=;
  if ina;
  study="Prostat_AmgenI_2006_121";
  dataset="ALBSAF ASLBASE";
  sas_note='';
  lbseq=.;
run;
```

6. Sort the work.LB1 dataset and write out the final dataset AMGENI2006\_121\_LB.sas7bdat to the outfile libname as specified in section step 2.

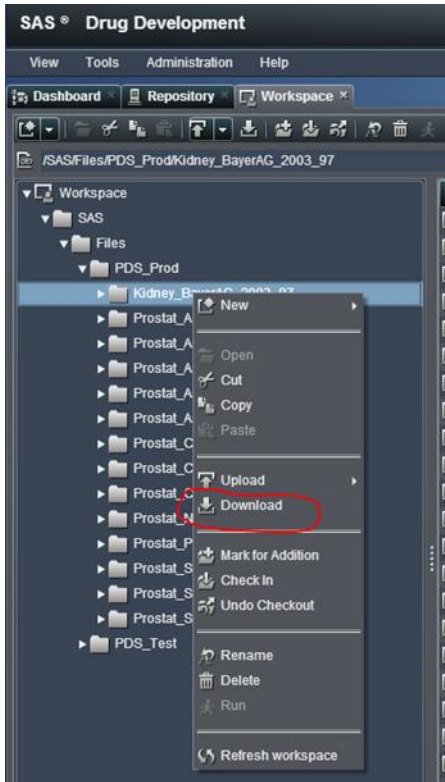
```
proc sort data=LB1 out=outfile.amgeni2006_121_LB;
  by studyid usubjid lbseq;
run;
```

## Downloading Datasets

Downloading a dataset is easy. It can be done for an entire Folder, a selection of files within a Folder, or a single file. Simply go to the output folder for files generated by the SAS session or select any folder with the dataset to be downloaded.

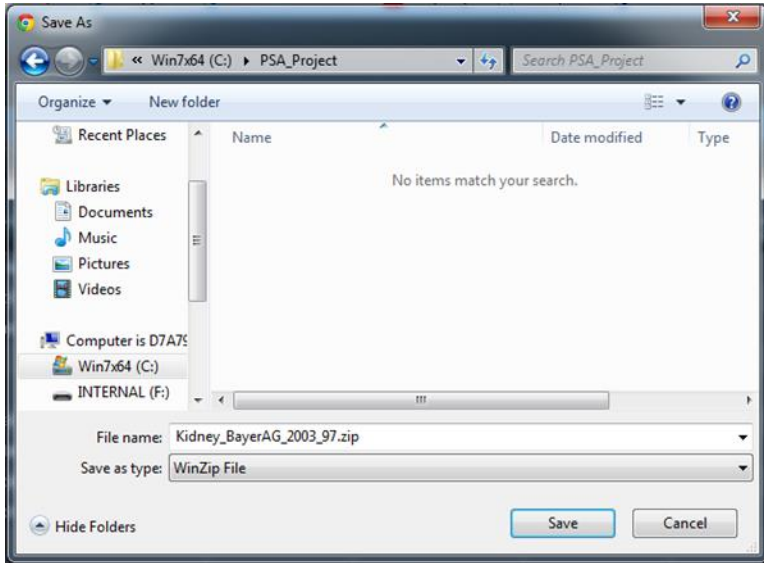
At the Folder level, select a Folder and using the RMB select the **Download** option from the menu.

Figure 16: Downloading files



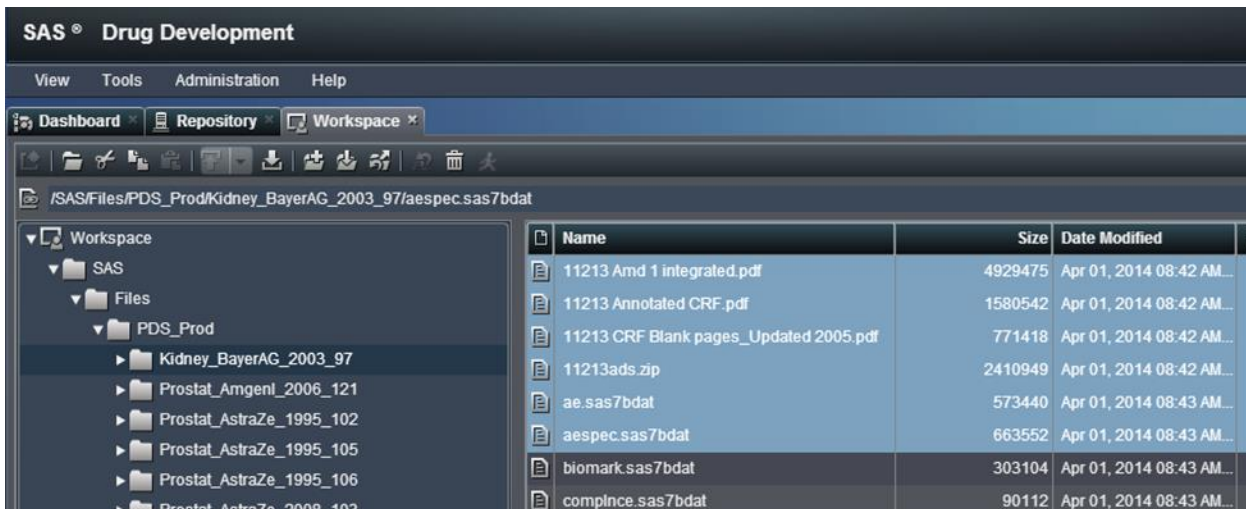
After **Download** is selected, the **Save As** window appears for directory selection on the local machine. When a folder is selected the Download will automatically place the contents in a pre-named ZIP file. In this case Kidney\_BayerAG\_2003\_97.zip. Click **Save**.

**Figure 17: Accessing the download area on the local drive**



Within the Folder level, select a file. Hold the keyboard **shift key** and **click** another file and this group will be highlighted for download. Using the same sequence above, click the **RMB** and select **Download**.

**Figure 18: Downloading a selection of files**



The **Ctrl key** on your keyboard can be used for selecting individual files for a group download. In both examples the last file in the selected list will be the name of the ZIP file, for the above example it will be `aespec.sas7bdat.zip`.

For an individual file, select the file, click the **RMP** and select **Download**. The file will be downloaded in its original format, not as a ZIP file.