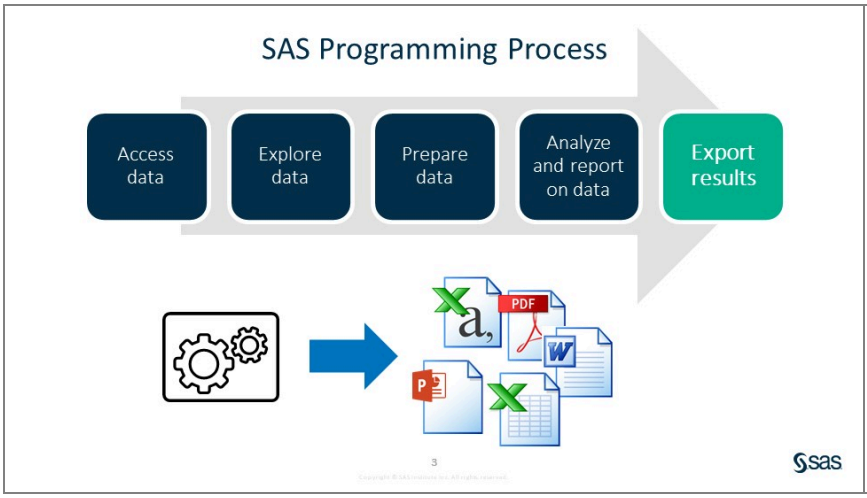
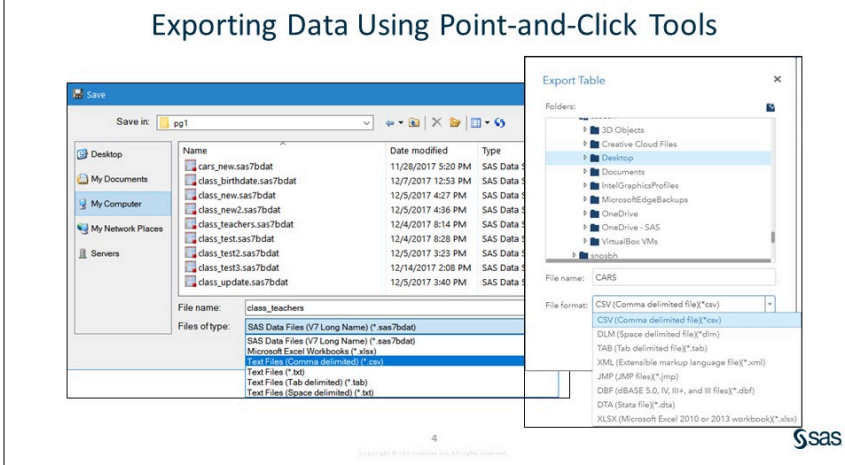
# B6.1 - Exporting Data



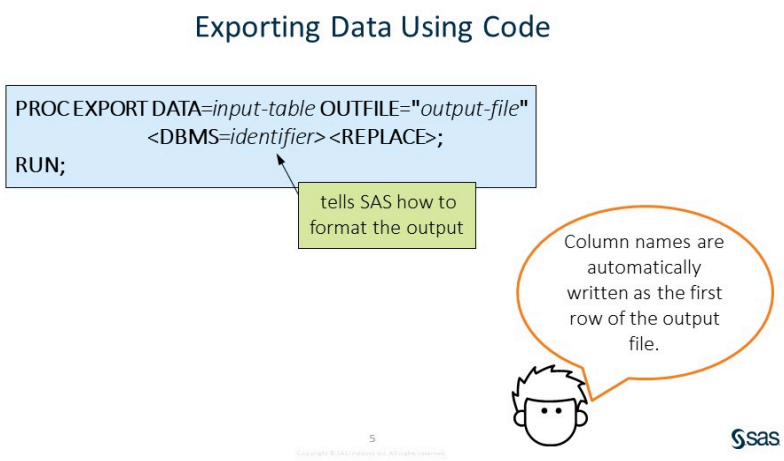

We've reached the end of the process. We have clean data and accurate, interesting reports, so now we need to share what we have created with others. Not everyone who needs to view your results uses SAS, so we need methods to export data and reports in formats that will be easy to view outside SAS.

## Exporting Data: Using Point and Click Tools



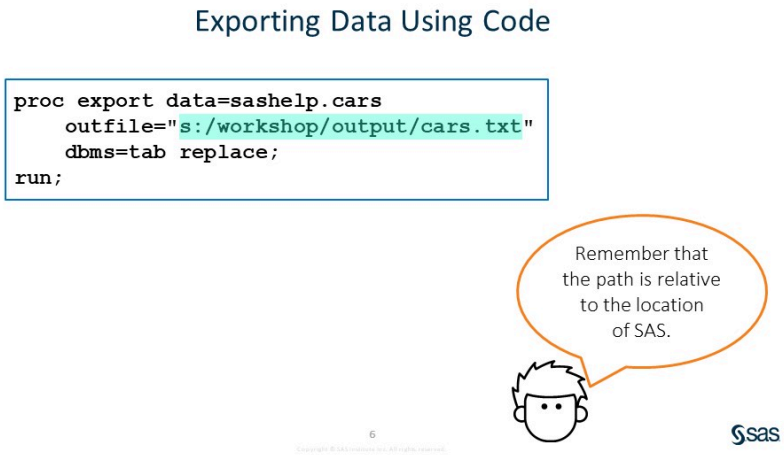

Let's start with exporting data. If you want to export data using a manual process, each of the SAS programming environments includes point-and-click tools for exporting data to various delimited text formats, such as comma-separated values (CSV), tab-delimited values (TAB) and space-delimited (DLM) files. In Enterprise Guide, you can start this process by selecting **Export** from the **File** menu. In SAS Studio, you can right-click a table in the Library panel and select **Export**.

## Exporting Data: Using Code




We are programmers. and there are several easy methods to programmatically export data, too. By writing a program to export data, you can easily integrate the export into your overall program to automate the final export step. This is SAS, so, of course, we have a PROC for that! PROC EXPORT can export a SAS table to a variety of external formats.

The DATA= option specifies the data source. The OUTFILE= option specifies the fully qualified path and file name of the exported data file. The DBMS= option tells SAS how to format the output. Several options are available here, including CSV, TAB, DLM, or XLSX. The REPLACE options tells SAS to overwrite the output if it already exists.




In this code example, PROC EXPORT creates a tab-delimited text file that has column names in the first row of the file. This is a good time for a reminder: the path that you put in the OUTFILE= option must be relative to the location of SAS. In other words, if SAS is running on a server, the path must be accessible from the server location.

## Activity 6.01:

1. Open the **libname.sas** program in the course files folder.

2. Create a macro variable named **outpath** that stores the location of the **output** folder in your course files location.
3. Run the code and save the program.

Click here for Solution.

## Activity 6.02:

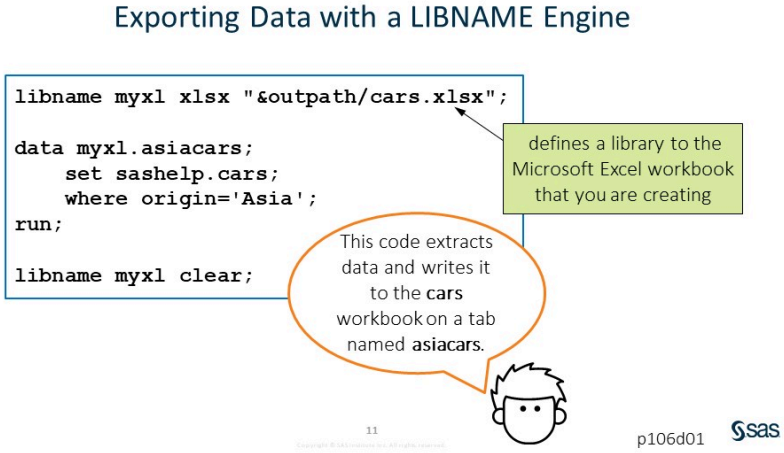
Open **p106a02.sas** from the **activities** folder and perform the following tasks:

1. Complete the PROC EXPORT step to read the **pg1.storm_fina**l SAS table and create a comma-delimited file named **storm_final.csv**. Use **&outpath** to substitute the path of the **output** folder.
2. Run the program and view the text file:

   SAS Studio – Navigate to the **output** folder in the Navigation pane, right-click **storm_final.csv**, and select **View File as Text**.

Click here for Solution.

# Export Data: Using LIBNAME Engine




PROC EXPORT is simple, but another easy way to export data is to use a SAS/ACCESS Interface LIBNAME engine. We simply create the data in the desired format right from a SAS process. For example, a DATA step or procedure OUTPUT statement can write results directly to the target data source. I don't have to create a SAS table first

| | |
|---|---|
| | and then export the SAS table in a separate step. Of course, you need Write permission to the target destination.<br><br>As an example, this program uses the SAS/ACCESS Interface to PC File Formats XLSX engine to define a library to an Excel workbook named **cars**. The DATA step references the library and output worksheet named **asiacars**. The code extracts data about cars manufactured in Asia from **sashelp.cars** and writes the result directly into the worksheet **asiacars**. |

## Demo: Exporting Data to Excel Workbook

**6_1 - Demo - Exporting Data to an Excel Workbook.pdf**
**(https://clemson.instructure.com/courses/237270/files/23074697/download?wrap=1)**

**(https://clemson.instructure.com/courses/237270/files/23074697/download?download_frd=1)**

## Activity 6.03:

Open **p106a03.sas** from the **activities** folder and perform the following tasks:

1. Complete the LIBNAME statement using the XLSX engine to create an Excel workbook named **storm.xlsx** in the **output** folder.
2. Modify the DATA step to write the **storm_final** table to the **storm.xlsx** file.
3. After the DATA step, write a statement to clear the library.
4. Run the program and view the log to confirm that **storm.xlsx** was exported with 3092 rows.
5. If possible, open the **storm.xlsx** file. How do dates appear in the **storm_final** workbook?

Click here for Solution.