

Paper 139-2011

## Let's Get Connected: How We Link to our Data

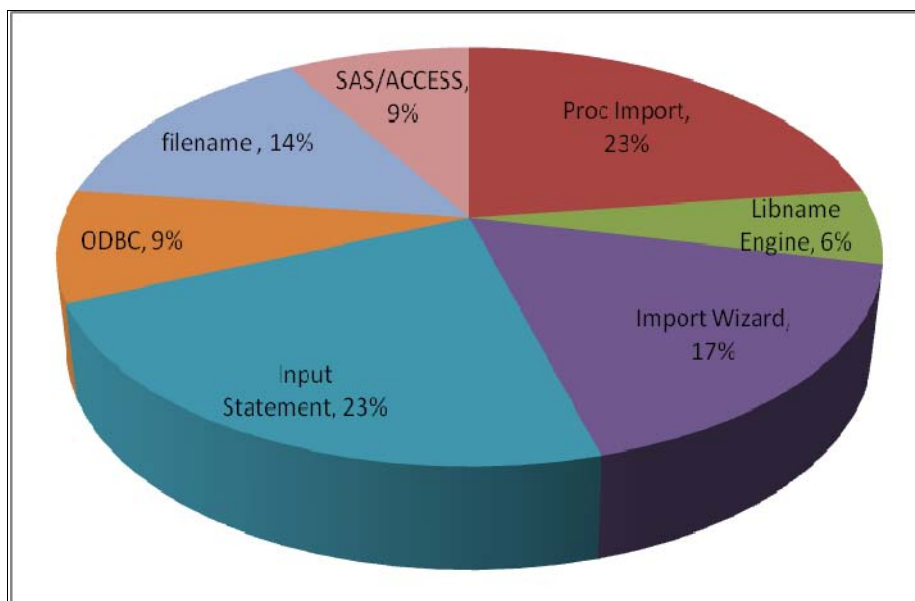
Jenine Milum, Wells Fargo Bank, Charlotte, NC

### ABSTRACT

You've got data, so how do you get it into SAS? One of the great strengths of SAS is that it is 'data agnostic' and can accurately read and write any of a dozen data storage formats. An additional benefit within SAS is that for any given format there is more than one method or technique to access it. How do you refer to them? Which operating systems are used? Which methods are most appropriate in which situations? This presentation provides, at a high level, a quick reference to the many options, tools, and methods that connect us with our data.

### INTRODUCTION

We have a very large SAS users group in Charlotte, NC with over 400 members representing 60 companies. I reached out to these SAS users to share with me the methods they use to connect to their data. When it comes right down to it, we keep it simple. No fancy methods or tools. We predominately import our data using Proc Import or the SAS Import Wizard. Delving a bit deeper, MS Excel files or .csv files are the most common forms of imported data. By far, Excel is the most commonly used Data Base converted into SAS data.

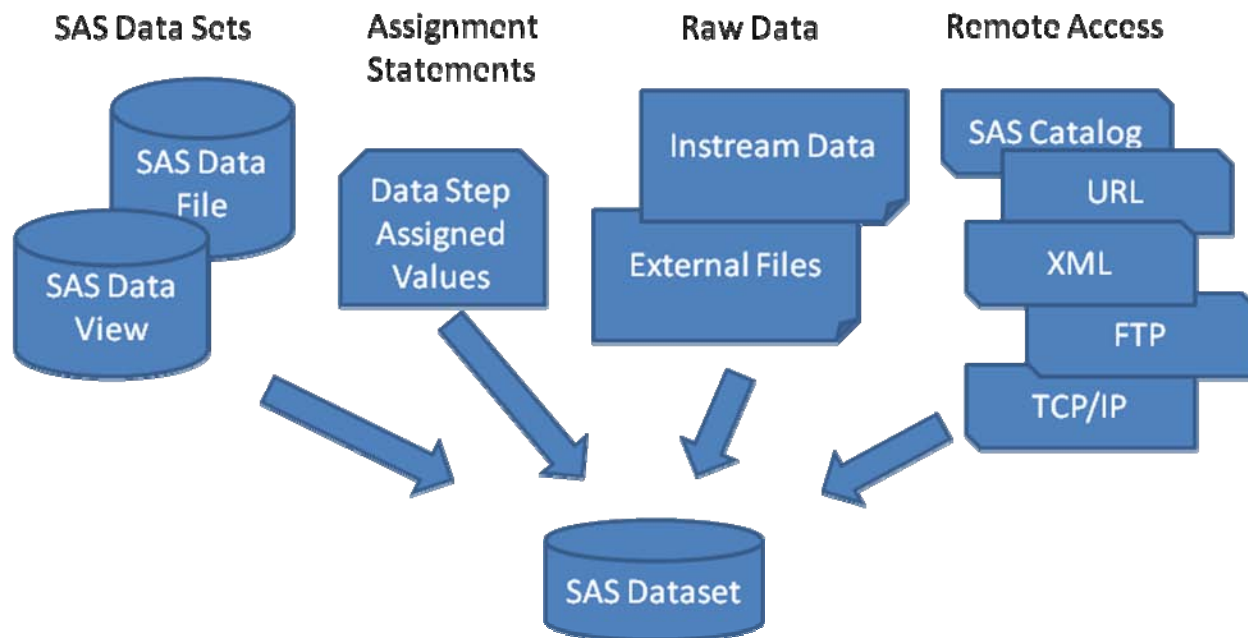


**Figure 1. Charlotte SAS Users**  
Methods used to Connect to non-SAS data

Even after years of being a SAS software developer, I find the vast array of possibilities to connect to data a bit daunting. The terminology keeps changing, the approaches can be vastly different and the look and feel of the actual coding can seem convoluted. In writing this paper, I wanted to summarize these possibilities, provide the terminology and give basic examples of each method.

## TYPES OF DATA SOURCES

To begin connecting to our data we need to identify and define the different types of Data Sources we may encounter. Some are far more prevalent than others as we see above. These inputs can be categorized as SAS Data Sets, Assignment Statements, Raw Data and Remote Access Inputs.



**Figure 2. Data Sources**

- A *SAS Data Set* is data in a SAS format and already available and accessible.
- *Assignment Statements* are data accessed from within a SAS data step.
- *Raw Data* is data that has yet to be processed to create SAS data.
- *Remote Access* allows you to connect to data from non traditional sources.

## SAS DATA SETS

It goes without saying; SAS uses SAS data sets best. There is no data conversion; there are no worries about formatting or concerns with data interpretation. SAS stores the information about SAS data using SAS metadata. SAS stores its data in sequential files in a column / row format. It flows naturally and as efficient as absolutely possible. That means data that isn't already in a SAS format needs to be converted. Conversions may expand from the very simple to the complex.

A SAS Data File is SAS system stored data which includes both descriptor information and data values in a file formatted by the SAS System.

A SAS Data View is a definition of a virtual data set. It contains no data values, but rather the information required to access the data values. A SAS data view can be analyze the contained data values. Views can be created with PROC SQL, the DATA step's View option, or SAS/ACCESS software.

Type	Method	Example
<b>SAS dataset</b>	direct	<pre>Data out_dsname; set in_dsname; run;</pre>
<b>SAS data view</b>	view	<pre>Data out_dsname / view=out_dsname; Set in_dsname; run;  * read SAS data view;  proc print data=out_dsname; run;</pre>

## ASSIGNMENT STATEMENTS

Assignment Statements are used to create SAS data sets within a data step. While Assignment Statements are certainly an access method to data, it is not comparable to the others because of its obvious limitations. Assignment statements are largely utilized to attribute values to variables that will be used throughout a program and not as data in its more true sense to be analyzed and reported.

Type	Method	Example
<b>Assignment Statement</b>	Data step	<pre>data students; name = 'Smith'; age = 11; count = 4345; run;</pre>

## RAW DATA

Raw Data is information which has not yet been processed to create a SAS data set. There are 2 types of raw data. External Files and Instream data. External files are by far the most customary. External Files contain rows of data that are either formatted data arranged in columns or free formatted. In contrast, Instream data is information included in a SAS data step. You can read raw data using SAS statements, SAS Functions, External File Interface (EFI) and the Import Wizard.

Let's look at Instream data first. When you read raw instream data with a DATA step, you can use a combination of the INPUT, DATALINES, and INFILE statements. SAS automatically reads your data when you use these statements.

Type	Method	Example
<b>List</b>	Scanning	<pre>data students;   length name \$ 12;   input name \$ age count;   datalines; Smith 11 4345 ;</pre>
<b>Column</b>	Aligned Data	<pre>data students;   infile datalines trunc;   input name \$ 1-12 age 17-18 count 27-30;   datalines; Smith          11          4345 ;</pre>
<b>Formatted</b>	Column input with formatting capabilities	<pre>data students;   input name \$12.+4 age 2. +2 count comma5.;   datalines; Smith          11  4,345 ;</pre>
<b>Named</b>	Named values	<pre>data students;   input name=\$ age= count=;   datalines; name=Smith age=11 count=4345 ;</pre>

Some SAS functions provide an opportunity to read records of raw data from external files. These functions provide a little flexibility in handling raw data. A few such functions are FREAD, FPUT and SYMSG.

External files are managed well using the Import Wizard as a tool to transfer data between external data sources and SAS data sets. It is a series of windows presenting logical selections and options to read data from an external data source and write it to a SAS data set.

File types supported by The Import Wizard:

- Excel
- Access
- CSV
- Text
- Delimited
- Db
- JMP
- Lotus

The EFI (External File Interface) is an extension of the Import Wizard with which you can read data that is not in the SAS System's internal format. Utilizing the EFI imports data by mapping the external file's data records into SAS observations (rows of data), creating a data set you can use within SAS. This is effective when you want to have more control over the formatting of the actual data columns than is provided in standard Import Wizard conversions. Below is how to activate the EFI portion of the Import Wizard.

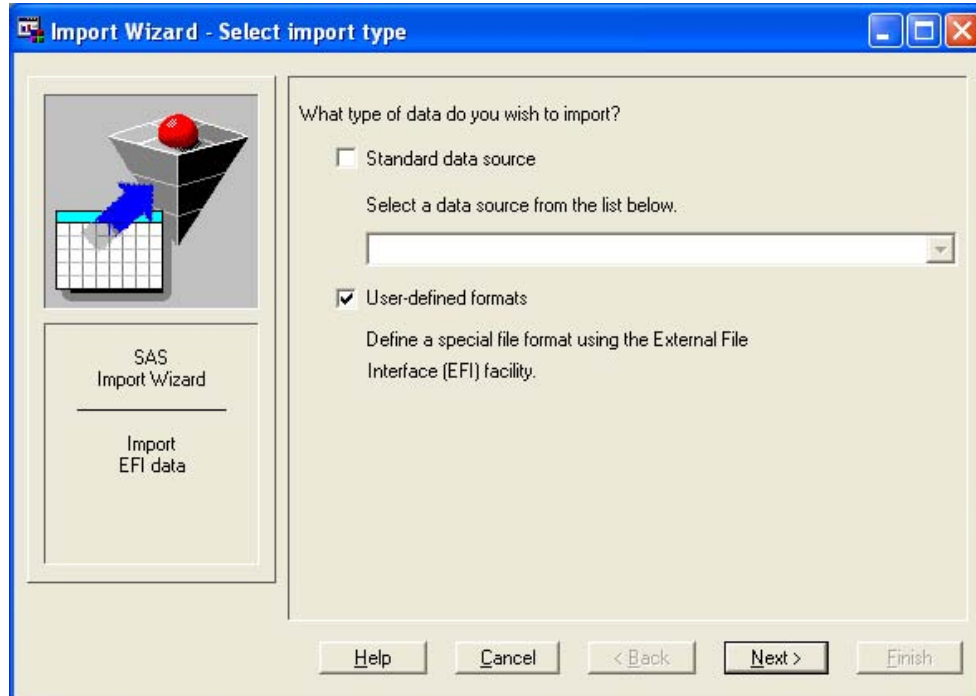


Figure 4. Import Wizard: How to initiate EFI.

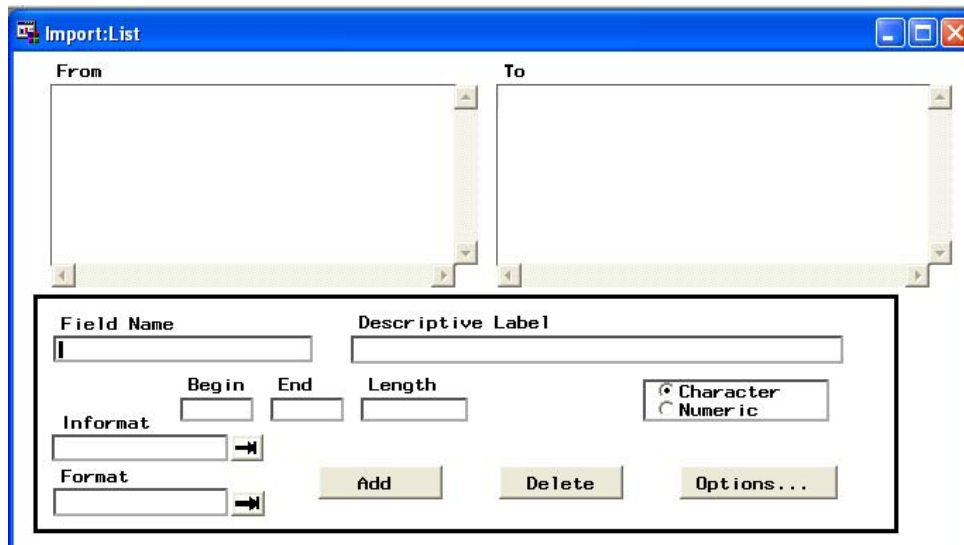


Figure 5. Import Wizard. EFI window

## REMOTE ACCESS

We saved Remote Access for the last because there is a lot more to be said here. Remote Access allows you to read input data from nontraditional sources such as a TCP/IP socket or a URL. SAS treats this data as if it were coming from an external file. SAS allows you to access your input data remotely in the following ways:

SAS catalog	This access method enables you to reference a SAS catalog as an external file.
URL	An access method which enables you to use a URL (Universal Resource Locator) to read from a file from any host machine that is connected to a network with a URL server running.
XML	The XML access method converts the hierarchical XML data structures and maps them into a relational data structure used by SAS
FTP	This method initiates the use of File Transfer Protocol (FTP) to read a file from any host machine that is connected to a network with an FTP server running.
TCP/IP socket	This method enables the user to read from a Transmission Control Protocol/Internet Protocol (TCP/IP) socket. The most familiar methods utilize SAS ODBC and SAS/ACCESS.

## SAS Catalog

SAS Catalogs are special SAS files that contain information called catalog entries. These entries are usually stored in a text format. In Base SAS software, SAS catalogs are usually accessed and read automatically by SAS when the information stored in them is required for processing. Your sasuser.profile is an example of such a catalog entry. Catalog entries contain system or application information. You can list the contents of a catalog using SAS Explorer and PROC CATALOG. These methods also allow the user to use SAS Catalogs as input and may be converted into SAS Data.

Type	Method	Example
<b>SAS Catalog</b>	File Reference	<pre>filename mycat catalog 'catalog';  * Example reading a SAS catalog as input;  Proc catalog cat=sasuser.profile;   Contents; Run;</pre>

## URL

Utilizing the SAS URL access method makes reading web pages as easy as reading text or other sequential files. SSL (Secure Socket Layer) protocol is utilized when the URL begins with *https* rather than *http* providing network security. This example reads a typical web page and the information embedded and is converted into SAS data.

Type	Method	Example
<b>URL Access Method</b>	Filename	<pre>FILENAME fileref URL 'external-file' &lt;url-options&gt;;</pre>

## XML

XML (Extensible Markup Language) documents are most easily read using the XML libname engine with an associated XMLMap. The map guides SAS into converting the XML hierarchal data into a SAS data set with rows and columns as is the SAS way. The example below shows how the xml file interacts with an xml map to create a reference to the newly created sas data set.

Type	Method	Example
<b>XML Access Method</b>	Libname engine with XML map	<pre>filename xmlfile 'C:\My Documents\XML\xmlfile.xml'; filename xmlmap 'C:\My Documents\XML\xmlmap.map'; libname xmlfile xml xmlmap=xmlmap;  proc print data=xmlfile.filename; run;</pre>

## FTP

The FTP access method invokes the File Transfer Protocol (FTP) to access remote files. FTP commands are issued as options on the FILENAME statement but are run only when the fileref is opened using an INFILE or FILE statement. When you need to access a file stored on a remote server when there is no direct facility provided, the FTP access method should be considered since it makes the whole process seamless. Additional uses of the FTP Access Method includes the basic reading of a file from a remote host, z/OS data, importing a transport data file and reading external directories.

The example retrieves the information stored on a SAS FTP site and makes a copy of it as a SAS data set.

Type	Method	Example
<b>FTP Access Method</b>	Filename using FTP protocol	<pre>FILENAME fileref FTP 'external-file' &lt;ftp-options&gt;;  filename students ftp '' ls user='teacher' host='host.ip.address.com' prompt;  data student_list;   infile students;   input;   put _INFILE_; run;</pre>

## TCP/IP

To utilize TCP/IP, base SAS, SAS/CONNECT or SAS/SHARE need to be installed on both the client and the server. SAS ODBC Driver (Open Database Connectivity) allows you to read other data sources such as SAS data sets, flat files, VSAM and DBMS (database management systems) as long as it has a SAS/ACCESS license.

So what are SAS/SHARE, SAS/CONNECT and SAS/ACCESS software?

SAS/SHARE	Allows multiple users to access the same information at the same time
SAS/CONNECT	Allows the user to connect to remote sessions of SAS
SAS/ACCESS	Allows the user to access data from other vendor's data

The ODBC Driver, associated with SAS/ACCESS, connects to data through either a local or remote SAS server session using the TCP/IP Protocol. The specific interface engines supported include:

- DB2
- SQL/DS
- IMS-DI/I
- IDMS
- DB2/2
- SQL/400
- ORACLE
- Rdb/VMS
- CA-DATACOM/DB
- INGRES
- INFORMIX
- SYBASE's SQL Server
- Microsoft's SQL Server
- ODBC
- ADABAS
- System 2000 Data Management Software
- PC file formats

SAS/ACCESS also has an interface to OLEDB. It requires a provider name and provider-specific connection information such as the user ID, password, schema, or server name. If you know all of this information, you can connect directly to a provider. If not, you can submit a libname statement with the OLEDB option and the system will prompt you for the connection information.



Type	Method	Example
<b>SAS/SHARE</b>	Managed by administrator	<code>LIBNAME sasuser1 '/sasuser' SERVER=server1;</code>
<b>SAS/CONNECT</b>	Remote submit	<pre>Options remote=servername comamid=TCP ; Filename rlink "!sasroot\connect\saslink\tcpunix.scr"; signon servername;  Rsubmit;  - Sas code  Endrsubmit;</pre>
<b>SAS/ACCESS ODBC</b>	Libname engine	<pre>* Oracle example;  libname mydblib oracle user=username password=user_password path='myoracleserver';</pre>
<b>SAS/ACCESS OLEDB</b>	Libname engine with OLE DB Interface	<pre>* Allow prompt for system information;  libname oledbname oledb;</pre>

## Pipe Access Methods

Another way which may be familiar to link to non-SAS data from external sources is utilizing the PIPE option within a filename statement. Pipe allows you to issue a system command through a filename statement. Such a command accesses external data and pulls it into your SAS environment creating a file listing which can then be references with an infile statement.

Type	Method	Example
<b>Pipe</b>	Issues system command	<pre>FILENAME fileref PIPE 'program-name' option- list;  * Windows example;  Filename dosfile pipe 'time;  Data time; Infile dosfile trunccover; Input line \$char200.; Run;</pre>

## **CEDA**

A note about CEDA (Cross-Environment Data Access); CEDA is a Base SAS component which allows a SAS data set housed in a directory-based operating system to be accessed by a SAS session that is running in another directory-based environment. The user does not have to be concerned with environment formats as CEDA allows a seamless and transparent conversion.

## **CONCLUSION**

In writing this paper, at least for myself, I was able to demystify the possibilities of all the possibilities available within SAS to access data. SAS software provides such a vast array of opportunities, choices, and options allowing the user to tailor fit nearly every possible situation. Formats, platforms, size, security and much more are all possible combinations of issues that would need to be considered in determining a method to suite your needs. There is almost always more than one approach to get our information into SAS so you can perform tasks on your data set to display it, sort it, create charts, compute statistics, and much more.

## **CONTACT INFORMATION**

Your comments and questions are valued and encouraged. Contact the author at:

Name: Jenine Milum  
Enterprise: Wells Fargo  
City, State: Charlotte, NC  
Work Phone: (704) 998-9877  
E-mail: Jenine.Milum@WellsFargo.com

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.