

SAS® Changes and Enhancements for Version 6.11 and Version 6.12

Arturo Barrios
University of North Carolina at Chapel Hill

Introduction

Versions 6.11 and 6.12 of SAS® introduced a series of new enhancements. In this paper I will talk about the ones that I found more useful. The first one is changes in the FILENAME statement, especially the use of the FTP access method and the URL access method, both very useful in an environment where the exchange of information with remote sites is an everyday task.

Another useful enhancement was the inclusion of an Import/Export facility wizard, a facility that helps users needing to exchange information among different applications. This GUI tool allows users to exchange data with popular file formats such as spreadsheets (MS Excel, Lotus), tab delimited files, comma delimited files, files from databases (*.dbf) and other delimited files.

There exist numerous other enhancements and I recommend you visit the SAS Institute web page for a more detailed list. The address is:
<http://www.sas.com/service/techsup/wn612/wn612.htm>

FILENAME Statement

The FILENAME statement in SAS supports these new access methods:

- the CATALOG access method, which allows you to read an entry in SAS catalog so that:
 - you can read it but not write to it
 - multiple users can read it at the same time
- the URL access method, which lets you read and write files on a HTTPD (hypertext transport protocol daemon) server.
- the FTP access method, which let you read and write files on a FTPD (file transfer protocol daemon) server.

FTP Access Method

Usually when you need to send or receive information from a different location, you do one of the following:

- create a SAS transport file, if your source data is in SAS format, and use one of the following methods: Proc CIMPORT/CPORT, or Proc COPY with the XPORT engine.
- Send/receive raw data, where raw data refers to ASCII data, or EBCDIC data, or plain text data. This data can be fixed column, tab delimited, comma delimited, etc.

Getting this data usually involves the following steps:

- creation of the file
- making the file available through an FTPD server
- download/upload the file
- process the file with SAS

What this access method (FTP) in the FILENAME statement allows you to do, is from inside your SAS session to read/write files from/to an FTP server. The advantage is that you will not need to generate extra copies of the file, SAS will keep and process these files from memory, saving valuable space in your hard disk and avoiding having copies of files with hard to identify versions floating around.

Syntax: FILENAME fileref FTP 'external-file' <ftp-options> ;

where:

fileref, is a valid fileref.

FTP, is the access method.

'external-file', is the name of the file to read from or write to on the remote host machine.

ftp-options, can be any of the common ftp options such as:

LRECL=lrecl where lrecl is the logical record length. The default value is 256.

BLOCKSIZE=blocksize where blocksize is the size of the socket data buffer in bytes. The default value is 8192.

RECFM=recfm where recfm is the record format. This option takes one of three values. V is the default. F is fixed record format. Thus, all records are of size LRECL with no line delimiters. Data are transferred in image (binary) mode. S is stream record format. The amount of data read is controlled by the current LRECL value or the value of the NBYTE= variable in the FILE or INFILE statement. The NBYTE= option specifies a variable equal to the amount of data to be read. This amount must be less than or equal to LRECL. Data are transferred in image (binary) mode. See the NBYTE= option under the FILE and INFILE statements for more information.

S370V indicates that the file being read is in IBM 370 variable format. The data are transferred in image or binary format, and are in local data format. Thus, you must use appropriate SAS informats to read the data correctly on non-EBCDIC hosts. If you specify this option and the RECFM= option, SAS ignores the RECFM= option.

S370VS indicates that the file being read is in IBM 370 variable-spanned format. The data are transferred in image or binary format, and are in local data format. Thus, you must use appropriate SAS informats to read the data correctly on non-EBCDIC hosts. If you

specify this option and the RECFM= option, SAS ignores the RECFM= option.

USER='username' where username is used to login to the FTP server.

HOST='host' where host is the network name of the remote host with the FTP server running. You can specify either the name of the host (for example, server.pc.sas.com) or the IP address of the machine (for example, 190.96.6.96).

PASS='password' where password is the password to use with the user name specified in the USER option. You can specify the PROMPT option instead of the PASS option, which tells the system to prompt you for the password. Note that if the user name is anonymous, the remote host may require that you specify your email address as the password.

PROMPT, specifies to prompt for the user login password, if necessary.

Example 1: Reading a File from a Remote Host

This example reads a raw file called "mydata" in the directory "/uabc" from the remote UNIX host "ftp.unc.edu"

```
filename myfile ftp 'mydata' cd='/uabc'
        user='guest' host='ftp.unc.edu'
        recfm=v prompt;
```

```
data mydata / view=mydata; /* Create a view */
    infile myfile;
    input x $10. y 4.;
run;
```

```
proc print data=mydata; /* Print the data */
run;
```

Example 2: Creating a File on a Remote Host

This example creates a file called "test.dat" in a directory called "c:\sasdata" for the user "arturo" on the host "salsant.unc.edu"

```
filename create ftp 'c:\sasdata\test.dat'
    host='salsant.unc.edu'
    user='arturo'
    prompt recfm=v;

data _null_;
    file create;
    do i=1 to 10;
        put i;
    end;
run;
```

Example 3: Retrieving and Executing SAS Statements from Another System

This example shows how to retrieve and then execute SAS statements from another system, and then submit them in your current SAS session.

```
filename inc ftp 'examp.sas' user='guest'
    host='ftp.unc.edu' prompt;
```

```
%include inc;
```

Example 4: Importing a Transport Dataset

This example uses the CIMPORT procedure to import a transport data set from an MVS host named "ftp1.unc.edu" for user "arturo". The new data set will reside locally in the SASUSER library. Note that user and password can be SAS macro variables. If you specify a full-qualified data set name, use double quotes and single quotes. Otherwise, the system will append the profile prefix to the name you specify.

```
%let user=arturo;
%let pw=itsesi2;
filename inp ftp "arturo.famcen96.xpt"
    user="&user" pass="&pw"
    rcmd='binary' host='ftp1.unc.edu';

proc cimport library=sasuser infile=inp;
run;
```

Example 5: Transporting a SAS Data Library

This example uses the CPORT procedure to transport a SAS data library to a MVS host named "ftp1.unc.edu" for user "arturo". It will create a new sequential file on the host called

"procen.xpt" with the recfm of fb, lrecl of 80, and blocksize of 8000.

```
filename inp ftp 'procen.xpt' user='arturo'
    pass="itsesi2" host='ftp1.unc.edu'
    lrecl=80 recfm=f blocksize=8000
    rcmd='site blocksize=8000 recfm=fb
    lrecl=80';
```

```
proc cport library=mylib file=inp;
run;
```

Example 6: Creating a Transport Library with Transport Engine

This example creates a new SAS data library on an MVS host UNCMVS. The FILENAME statement assigns a fileref to the new data set. Note the use of the RCMD= option to specify important file attributes. The LIBNAME statement uses a libref that is the same as the fileref and assigns it to the XPORT engine. The PROC COPY step copies all data sets from the SAS data library referenced by MYLIB to the XPORT engine. Output from the PROC CONTENTS step confirms that the copy was successful.

```
filename inp ftp 'project.xpt' user='arturo'
    pass="itsesi2" host='uncmvs'
    lrecl=80 recfm=f blocksize=8000
    rcmd='site blocksize=8000 recfm=fb
    lrecl=80';
```

```
libname mylib 'mysas.data.library';
libname inp xport;
```

```
proc copy in=mylib out=inp mt=data;
run;
proc contents data=inp._all_;
run;
```

Import/Export Facility Wizard

The Import/Export Wizard guides you through the processes of importing data from external sources to SAS, and exporting from SAS to external data sources. External data sources might include, database files, spreadsheets, delimited files, or files in formats you define using the External File Interface (EFI).

The Wizard is very easy to use. You just point and click, and the Wizard does the rest.

You invoke the Wizard any time you select Import or Export from the File pull-down menu in the SAS product you are working in. You can invoke the Import/Export Wizard from SAS/ASSIST, the SAS desktop, SQL Query Window, and the Program Editor.

For example, you can invoke the Import/Export Wizard by selecting Import or Export from the SAS desktop's File pull-down menu. In Query and Reporting, select Import or Export from the File pull-down menu to invoke either the Import or Export Wizard

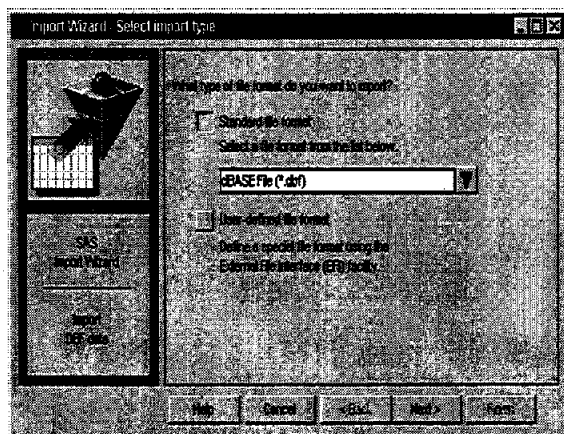
- generate HTML output from within your SAS session
- access the SAS System from within your web browser
- use Java to access SAS data
- use UNIX web servers to query and update SAS data

You can find complete descriptions of these and other tools under development, including working prototypes, in the Web Tools section of SAS Institute's web site at <http://www.sas.com/rnd/web>.

SAS is a registered trademark of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

You may contact the Author at:

Arturo Barrios, CB#3460, 300 Wilson Library,
University of North Carolina at Chapel Hill,
NC 27599-3460. (919) 966 1533.
Arturo_Barrios@unc.edu



What's New in Web Enablement Tools

SAS Institute is currently developing tools that integrate the latest web technology with the SAS System. Some of these tools are available now, as experimental components, for use with Release 6.12. These tools let you