Providing software solutions since 1976

Usage Note *46944:* New SAS® transport format and tools available

Details | Full Code | Output | Downloads | About | Rate It

The SAS Version 5 (V5) transport file format is an open standard developed by SAS to support data transfers between systems, especially those running different operating systems. SAS has recently developed a new transport file format specification, along with three macros, that extends the older V5 transport file format and provides new capabilities.

- The new specification, similar to the SAS V5 transport specification, is also nonproprietary.
- The new macros, available on the **Downloads** tab, can read or write the new extended transport format as well as the older specification. These macros can run equally well in SAS® 8 and SAS® 9. From **SAS Technical Paper *TS-140: Record Layout of a SAS Transport Data Set***, you can click on either the SAS V5 or SAS V8 transport specifications. This link takes you to the public specification for the **Record Layout for a SAS® Version 8 or 9 Data Set in SAS® Transport Format**.

The SAS V5 transport file is widely used in the biopharmaceutical industries. The availability of these new macros does not change the expectation that the FDA should receive clinical trial data in SAS Version 5 (V5) transport files.

The new SAS transport format and tools support the following:

- Variable names can be up to 32 characters, and they are stored in their original case (upper or lower). The previously released transport file format only allows up to 8 characters.
- Character variables can have lengths up to 32,767 bytes. The previously released transport file format only allows lengths up to 200 bytes.
- Variable names can contain any characters other than null ('00'x). The previously released transport file format only allows alphanumeric and underscore as characters. Note that a variable name cannot be completely blank. Any variable name that contains characters other than alphanumeric or underscore would be represented in the SAS language as an n-literal (for example, 'a b'n). Starting in SAS® 9.3, using the VALIDMEMNAME=EXTEND option, variable names can contain embedded blanks and these additional characters:

  ```
  .~`!@#$%^&()_+={}[];',
  ```

  These characters are allowed in the new transport file format, although you must be running SAS 9.3.
- Variable labels can be up to 256 characters. The previously released transport file format only allows 40-character labels.

The macro **LOC2XPT** (from local session to xport) can identify whether the data set you are reading in has any extended data set features named above. For example, if the data set has long variable names, a new transport file is created. Otherwise, the older V5 transport file is made. You can specify the type of transport file that you want to create using the FORMAT= parameter. A sample invocation of the **LOC2XPT** macro is:

```
%loc2xpt(libref=your_libref, memlist=dataset_name, filespec=your_filespec)
```

The macro **XPT2LOC** (from xport to local session) can convert older V5 transport files and the new transport file format. There are no restrictions regarding which version of SAS you are running in order to import the new transport file, as long as you are not running SAS Version 6. An example of calling the macro **XPT2LOC** macro follows:

```
%xpt2loc(libref=your_libref,filespec=your_filespec)
```

The **XPTCOMMN.SAS** file contains common code used by both macros.

Using a filename extension, you can differentiate between the transport file types. Any extension can be used, but we are using **.v9xpt** as the extension in the sample program because long format names exist, which is a new feature in SAS 9. If you receive a file ending in .v9xpt and want to convert it to a SAS data set using %XPT2LOC (assuming that you have submitted the programs from the **Downloads** tab in your SAS session), you would simply use the following statement:

```
%xpt2loc(libref=work, filespec='c:\trans.v9xpt')
                or
%xpt2loc(libref=work, filespec=fileref) /* where fileref is unquoted */
```

The XPORT engine is restricted to reading the older V5 transport files but the macro **XPT2LOC** can read both the older V5 transport files and the new transport file format. If you attempt to use the XPORT engine to read a new transport file, an error is generated. Instead, use the macro **XPT2LOC**.

Operating System and Release Information

Product Family	Product	System	SAS Release	
			Reported	Fixed*
SAS System	Base SAS	Microsoft Windows NT Workstation	8.1 TS1M0	
		Microsoft Windows 2000 Server	8.1 TS1M0	
		Microsoft Windows 2000 Professional	8.1 TS1M0	
		Microsoft Windows 2000 Datacenter Server	8.1 TS1M0	
		Microsoft Windows 2000 Advanced Server	8.1 TS1M0	
		Microsoft Windows 95/98	8.1 TS1M0	
		OS/2	8.1 TS1M0	
		OpenVMS VAX	8.1 TS1M0	

		z/OS	8.1 TS1M0	
		Microsoft Windows XP Professional	8.1 TS1M0	
		Windows Millennium Edition (Me)	8.1 TS1M0	
		ABI+ for Intel Architecture	8.1 TS1M0	
		AIX	8.1 TS1M0	
		HP-UX	8.1 TS1M0	
		IRIX	8.1 TS1M0	
		OpenVMS Alpha	8.1 TS1M0	
		Solaris	8.1 TS1M0	
		Tru64 UNIX	8.1 TS1M0	

* For software releases that are not yet generally available, the Fixed Release is the software release in which the problem is planned to be fixed.

The SAS Version 5 (V5) transport file is an open standard developed by SAS to support data transfers between systems, especially those running different operating systems. SAS has recently developed a new transport file format, along with a series of macros, that extends the older V5 transport file format and provides new capabilities. These capabilities are described on the **Details** tab.

The only step necessary to run the code below is to save the programs from the **Downloads** tab to the C:\TEMP directory. The sample program uses %INCLUDE statements to locate them in this directory and then automatically compiles the macros to the Work library when you submit the code. Alternatively, you can copy and paste the programs into your SAS Enhanced Editor and run them individually. In this case, you are manually compiling the macros to the Work library, so to avoid errors, you need to comment out the %INCLUDE statements. The advantage of using the %INCLUDE statements is that you retain a copy of the macros rather than losing them when you exit your SAS session.

Additional instructions are given below the sample program for making the macros readily accessible using the AUTOCALL facility. With the AUTOCALL facility, you no longer need to compile the macros by using %INCLUDE statements or submit them as SAS programs in any future SAS session. A one-line statement is all that is required to read and write the new transport files.

```
/* This sample creates and compares the data set before and after the transport operation. */
/* This program assumes the files from the DOWNLOAD tab exist in the C:\TEMP directory. */

/* Toggle options off */
options nosource2 nosource;

/* Test case */
/* Build a format whose name is greater than 8 characters. */

proc format;
   value testfmtnamethatislongerthaneight 100='numeric has been formatted';
run;

/* Build a data set with > 8-character data set name, variable name, and > 40-character label */

libname test 'c:\temp';
data test.Thisisalongdatasetname;
   varnamegreaterthan8=100;
   label varnamegreaterthan8='jjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjj';
/* Assign permanent format */
   format varnamegreaterthan8 testfmtnamethatislongerthaneight.;
run;

/* Compile the macros in the Work library */
%include "c:\temp\loc2xpt.sas";
%include "c:\temp\xpt2loc.sas";
%include "c:\temp\xptcommn.sas";

/* Use the %LOC2XPT macro to create a V9 transport file */

/* libref= points to a directory containing datasets specified in memlist=     */
/* filespec= specifies path and name of transport file we are creating         */
/*    name and extension can be anything but we are using trans.v9xpt -        */
/*    to remind us it contains a data set with features specific to v9 and     */
/*    we should use a SAS 9 session to read it.                                */

%loc2xpt(libref=test,memlist=Thisisalongdatasetname,filespec='c:\trans.v9xpt' )

/* Use the %XPT2LOC to convert V9 transport file to a SAS data set. */
/* Note that data set features are retained in the PROC COMPARE output. */

/* libref= points to target folder where data sets will be stored              */
/* filespec= points to existing transport file                                 */

%xpt2loc(libref=work, memlist=Thisisalongdatasetname, filespec='c:\temp\trans.v9xpt' )

/* Compare data sets before and after the transport operations */
ods listing;
title 'Compare before and after data set attributes';
proc compare base=test.Thisisalongdatasetname compare=work.Thisisalongdatasetname;
run;

/* Restore option settings */
options source2 source;
```

```
***********************************************************************;

 How to add the macros to your Autocall library


Simply add the path where you stored the extended transport macro programs to the list of
paths specified in the SASV9.CFG file under -SET SASAUTOS.  This makes using the macros as
simple as specifying the macro name.
```

Alternatively, if you do not have the ability to modify the SASV9.CFG file,
add the statements below to the AUTOEXEC.SAS file or submit them when you first open your SAS session.

These statements will help you find your current SAS Autocall library and add your folder to it.

```
filename testauto 'path-to-folder-containing-the-downloaded-macros';
/* Check the current path to your autocall library*/
%put %sysfunc(getoption(sasautos));
/* Add testauto to the current sasautos setting */
options sasautos=('C:\Program Files\SAS\SASFoundation\9.2\core\sasmacro' testauto) mautosource mrecall;
```

Additional documentation for use of extended transport file macros

```
%loc2xpt(libref=your_libref,memlist=your_memlist,filespec=your_filespec,
        format=your_format);
```

Where:

your_libref indicates the libref where the members reside. If the
LIBREF= option is omitted, the default is WORK.

your_memlist indicates the list of members in the library that are to
be converted. The default is that all members will be converted.

your_filespec indicates either a fileref (not quoted) or a path name
(single quoted) which is the location where the Version 8 transport file will
be written. This argument is required and has no default value.

your_format indicates the format of the transport file. It can be
V6 (Version 6 transport), V8 (Version 8 transport file), or AUTO
(determined by data). If your data set contains no long variable names,
long labels, or character variables over 200, and you used AUTO, then
a Version 6 transport file is written. You will receive an error if
you use V6 and any of these attributes are present.
**********;

```
%xpt2loc(libref=your_libref,memlist=your_memlist,filespec=your_filespec);
```

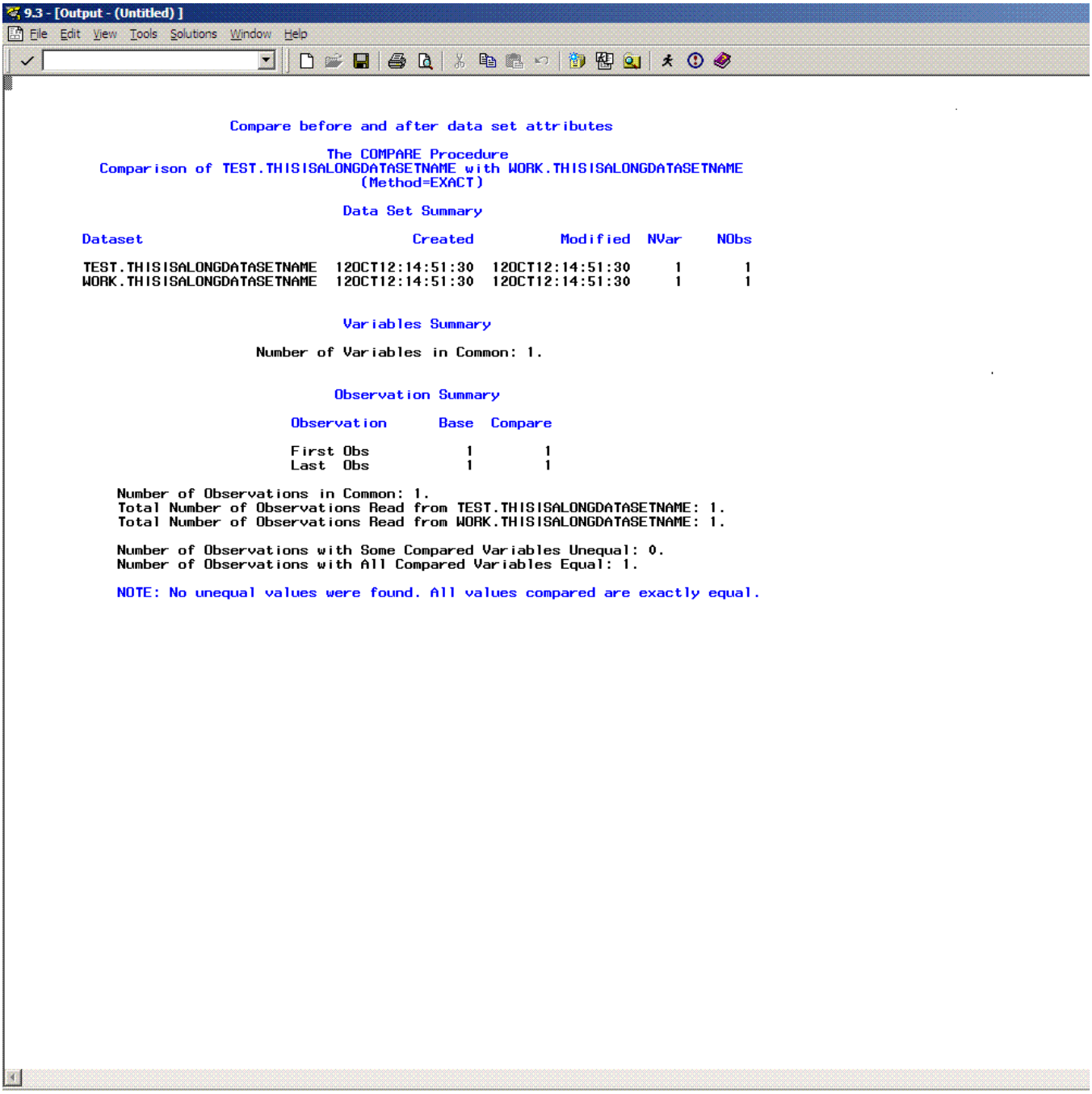
Where:

your_libref indicates the libref where the members will be written. If the
LIBREF= option is omitted, the default is WORK.

your_memlist indicates the list of members in the transport file that are to
be converted. The default is that all members will be converted.

your_filespec indicates either a fileref (not quoted) or a path name
(single quoted) which is the location where the Version 8 transport file resides.
This argument is required and has no default value.
**********;

Comparison showing that SAS 8 data set features are retained



LOC2XPT.SAS is used to export SAS 8 and SAS 9 data sets into extended transport format.

XPT2LOC.SAS is used for importing both traditional Version 5 transport files or extended transport files.

XPTCOMMN.SAS is used by both %LOC2XPT and %XPT2LOC.

Variable names longer than eight characters and other unsupported data set features in the XPORT engine are supported. This availability does not change the expectation that the FDA should receive clinical trial data in SAS Version 5 (V5) transport files.

Type: Usage Note

Priority:

Topic: Common Programming Tasks ==> Reading and Writing External Data
Common Programming Tasks ==> Reading and Writing External Data ==> File Options
Common Programming Tasks ==> Reading and Writing External Data ==> with PROC EXPORT
Common Programming Tasks ==> Reading and Writing External Data ==> with PROC IMPORT
Data Management ==> Data Sources ==> CEDA (Cross-environment Data Access)
Data Management ==> Data Sources ==> SAS Data Sets/Tables
Common Programming Tasks ==> Reading and Writing SAS Data

Date Modified: 2012-10-19 11:46:00

Date Created: 2012-07-07 17:38:33