

Transporting SAS Libraries

June, 2004

A SAS Library is any subdirectory (folder) containing SAS datasets and catalogs. In general, SAS datasets and catalogs created on one operating system cannot necessarily be read directly on another operating system. Within an operating system, files created using an earlier release of SAS can usually be read using special "engines". An exception arises when the attempting to process 32-bit files with a newer 64-bit SAS release. See [32 bit and 64-bit files](#).

The procedures described here are for converting SAS datasets and catalogs *between* operating systems or for converting files from 32-bit to 64-bit format on the same operating system.

The new PROC MIGRATE utility introduced in SAS Release 9 is intended solely for the purpose of converting SAS files created under earlier releases of the SAS system to newer releases on the *same* operating system, and will not be described in this document. It can be used for 32 to 64-bit conversion, but should NOT be used for moving SAS files between operating systems. For more information on PROC MIGRATE and 32 to 64-bit conversion, see <http://support.sas.com/rnd/migration/resources/procmigrate/index.html>

SAS Library Compatibilities:

The way SAS stores datasets and catalogs depends on both the version of SAS used to create the dataset/catalog and the operating system on which it is running. For brevity, this documentation will use the term "SAS files" to mean either SAS datasets or SAS catalogs.

SAS Versions

Higher versions of SAS can usually read SAS files created by lower versions on the same operating system, provided both releases use 32-bit storage. For example, SAS 9 on Windows can read SAS files created by SAS 6.12 on Windows. To do this, you may need to use a "compatibility engine".

If all the SAS files in a SAS library (subdirectory) were created by the same version of SAS, SAS will automatically assign the appropriate engine when you assign the libname. If a SAS library contains files created by various versions of SAS (a "mixed library"), you must assign the proper compatibility engine for the file(s) you want to access. You do this on the LIBNAME command:

```
libname lib6 v6 'c:\sasdata\';  
libname lib9 v9 'c:\sasdata\';
```

The above statements assign the V6 and V9 engines, with distinct names, to the same subdirectory C:\sasdata\. You can then use the library reference lib6 to read SAS files created by SAS versions 6.08 to 6.12 and stored in c:\sasdata. Use the library reference lib9 to read SAS files created by SAS 9, stored in the same directory.

It is desirable to keep SAS files created by different versions of SAS in separate directories. That way the correct compatibility engine will be assigned automatically. Further, if a SAS file contains features introduced in a later SAS release, you will not accidentally attempt to use it with a release that does not support that feature. Keeping version 8 and version 9 files in separate directories is particularly desirable because the file extensions used by these versions are the same. As you begin to use SAS 9 features not supported by SAS 8, it will be difficult to tell which files are compatible with SAS 8 unless they are segregated. See [Forward Compatibility](#).

SAS File Extensions

SAS assigns specific file extensions to all SAS files. The extensions used depend on the version of SAS and the operating system. If the extensions used by two operating systems are the same, you can move SAS files created on one system and read them on the other without special processing, provided there is no 32-bit vs. 64-bit storage incompatibility. For example, SAS 8 and SAS 9 use the extension `.sas7bdat` for datasets on both Windows and OITUNIX. You can move SAS 8 files between Windows and OITUNIX and read them on either system, but SAS 9 files on OITUNIX are only partially compatible with SAS 9 files on Windows and with SAS 8 file on OITUNIX. See [32-bit and 64-bit files](#).

If the SAS extensions on two operating systems are different, you must use transport files to move the data between them. Manually changing the extension to the "correct" one for the new operating system will not help – the extension is a reflection of internal differences.

Forward Compatibility

In general, SAS files that use the same extension are forward compatible to the extent that they do not contain features not supported by earlier releases. For example, SAS 8 user-defined formats are limited to names no more than 8 characters long, while SAS 9 supports user-defined format names up to 30-32 characters long. SAS 9 files can be used with SAS 8 provided they do not violate any SAS 8 restrictions. A SAS file with long format names cannot be used with SAS 8, regardless of the file extension, nor can it be transported to SAS 8 using any of the procedures described here. In order to move such a file to SAS 8, the long format names must first be removed using SAS 9.

The forward compatibility issues you are most likely to encounter are:

- Format and informat names over 8 characters long are not supported prior to SAS 9
- Variable names over 8 characters long are not supported prior to SAS 7
- 64-bit SAS catalogs

32-bit and 64-bit files:

SAS Release 9 introduces 64-bit data storage. All previous versions of SAS use 32-bit data storage. 32-bit and 64-bit SAS files are "foreign" to each other, and are only partially compatible, regardless of operating system or file extension. The UMass SAS Windows license uses 32-bit file storage, but OITUNIX SAS Solaris uses 64-bit storage. This makes datasets created with earlier SAS releases on OITUNIX and ALL Windows SAS releases (including 9) "foreign" to SAS 9 on OITUNIX. Furthermore, although the UMass SAS Windows license is 32-bit, a 64-bit Windows version exists. Therefore SAS Windows files received from other sites may also be "foreign" to SAS Windows at UMass.

32-bit SAS data files can be read by the 64-bit version transparently, and vice versa, though the processing requires resources that may adversely impact performance. However, 32-bit and 64-bit SAS catalogs are mutually incompatible without conversion, and not all processing of "foreign" data files is supported. Therefore, for improved performance and full access to all features, SAS files created on OITUNIX under SAS 8.2 (32-bit) or any SAS release on Windows should be converted for full compatibility with SAS 9 (64-bit) on OITUNIX. Similarly, 64-bit files created using SAS 9 on OITUNIX (or acquired from outside) should be converted to 32-bit for use with SAS on Windows. Small data files without associated catalogs, where performance is not a priority, can be used on either system/version without conversion.

PROC CPORT and CIMPORT can be used to convert both SAS catalogs and data files between 32-bit and 64-bit forms at the same SAS release number. PROC MIGRATE can be used to convert data files but not catalogs from 32-bit to 64-bit.

Additional information on the limitations of using SAS 9 with "foreign" datasets, and on the relative merits of using PROC MIGRATE versus PROC CPORT and CIPMORT for 32-bit to 64-bit conversion is available at

<http://support.sas.com/rnd/migration/planning/platform/64bit.html>

Determining whether a SAS file is 32-bit or 64-bit files:

PROC CONTENTS in SAS 9 has a new entry that shows whether the data representation is 32-bit or 64-bit. The following output line from PROC CONTENTS shows that the file uses 32-bit data representation under Windows:

```
Data Representation  WINDOWS_32
```

Files created using 64-bit SAS would have `_64` as the final part of the Data Representation field.

Using the **MSGLEVEL=i** option puts a message in the log whenever SAS converts a "foreign" file:

```
OPTIONS MSGLEVEL=i;      * Letter I, not number 1;
```

With this option, if you are using a 32-bit file on a 64-bit system (or vice versa) you will see a message in the SAS log:

```
INFO: Data file U.IRIS8.DATA is in a format native to another host or the file encoding does not match the session encoding. Cross Environment Data Access will be used, which may require additional CPU resources and reduce performance.
```

Compatibility Summary Table

This table displays compatibilities between operating systems and versions of SAS available at UMass.

- "Full" means that SAS datasets and catalogs can be moved freely between systems, without special processing.
- "Contingent" means SAS files do not require special processing to move between systems, but are subject to issues discussed under *Forward Compatibility*.
- "Limited" means SAS data files, but not catalogs, can be moved across systems, but will have some restrictions due to 32-bit versus 64-bit data representation. Use PROC CPORT and CIMPORT to move catalogs and remove restrictions on data files.
- "V6 Engine" means datasets and catalogs can be used directly with the V6 engine.
- "PROC COPY" means PROC COPY is likely to be the best choice for transporting data. The source file cannot contain feature not supported by the target system – e.g. long variable names if the target is SAS 6.12,
- "CPORT/CIMPORT" means PROC CPORT and PROC CIMPORT are likely to be the best choice for transporting data and catalogs.

If you are planning to take/get SAS data to/from another site, use transport files unless you know for sure that the other system is compatible with yours.

The entries in parentheses under each row/column heading show the file extension used for SAS datasets/catalogs – for example, under Windows, SAS 8 datasets have the extension .sas7bdat, and catalogs have the extension .sas7bcats.

	Target System	Target System	Target System	Target System	Target System	Target System
Source System	Windows SAS 9 (sas7bdat/sas7bcats)	Windows SAS 8 (sas7bdat/sas7bcats)	Windows SAS 6.12 (sd2/sc2)	OITUNIX SAS 9 (sas7bdat/sas7bcats)	OITUNIX SAS 8 (sas7bdat/sas7bcats)	Macintosh SAS 6.12 (ssd01/sct01)
Windows SAS 9 (sas7bdat/sas7bcats)		Contingent	PROC COPY	Limited	Contingent	PROC COPY
Windows SAS 8 (sas7bdat/sas7bcats)	Full		PROC COPY	Limited	Full	PROC COPY
Windows SAS 6.12 (sd2/sc2)	V6 Engine	V6 Engine		CPORT/CIMPORT	CPORT/CIMPORT	CPORT/CIMPORT
OITUNIX SAS 9 (sas7bdat/sas7bcats)	Limited	Limited and Contingent	PROC COPY		Limited and Contingent	PROC COPY
OITUNIX SAS 8 (sas7bdat/sas7bcats)	Full	Full	PROC COPY	Limited		PROC COPY
Macintosh SAS 6.12 (ssd01/sct01)	CPORT/CIMPORT	CPORT/CIMPORT	CPORT/CIMPORT	CPORT/CIMPORT	CPORT/CIMPORT	

Transporting SAS Files

To move SAS data files to another system:

- Create a transport file using any SAS release that is fully compatible with the one that created the SAS file.
- Move the transport file to the new system. If you use ftp, be sure to use binary transfer mode.
- Import the transport file on the new system.

Proc COPY vs. Proc CPORT/CIMPORT

Transport files can be created and read using either PROC COPY or PROC CPORT & PROC CIMPORT, but you cannot mix and match. Transport files created with PROC COPY must be read with PROC COPY; those created by PROC CPORT must be read with PROC CIMPORT.

PROC CPORT/CIMPORT can be used to transport both SAS datasets and SAS catalogs, but cannot be used to move SAS data to an earlier release of SAS. In contrast, PROC COPY can be used to go to an earlier release of SAS, but will not transport SAS catalogs. If you must move catalogs with PROC COPY they have to be converted to a dataset using PROC FORMAT with the CNTLOUT option. This process will not be discussed here. Whenever moving SAS data to an earlier release, keep in mind any restrictions the earlier release may have. See "[Forward Compatibility](#)".

If you need to take/get SAS data to/from another site, be sure to find out what release of SAS is used at the other site, whether the file is 32-bit or 64-bit, and whether catalogs are involved so you can use the procedure best suited to the task.

Creating a Transport File – PROC CPORT

Windows

To create a transport file of an entire SAS library, including datasets and catalogs on Windows, use PROC CPORT:

```
LIBNAME saslib 'C:\mysasdir\';
PROC CPORT LIBRARY=saslib FILE='C:\mysasdir\trans.exp';
RUN;
```

You can use version 6, 8 or 9 to run this code. PROC CPORT writes all the SAS datasets and catalogs in the current directory to file `trans.exp`. (There is no need to use the `FMTSEARCH` option to include all formats catalogs.) If the directory has SAS files created by SAS 6 as well as SAS 8 or 9, you must specify the engine for the datasets to be transported. For Version 6.12 datasets, use:

```
LIBNAME saslib V6 '';
```

NOTE: Version 8 and version 9 datasets and catalogs in the same directory are indistinguishable based on filenames. However, they may not be fully compatible - see [Forward Compatibility](#). Keep separate directories to avoid confusion.

If you have more than one SAS library to transport, run this code in each directory. Substitute a different name for each transport file created.

Options and subcommands can be added to PROC CPORT to select or exclude some files, or to select only datasets or only catalogs;

```
LIBNAME saslib 'C:\mysasdir\';
PROC CPORT LIBRARY=saslib FILE='trans.exp' memtype=data;
EXCLUDE junkfile;
RUN;
```

The above creates a transport file, `trans.exp`, of only datasets, not catalogs (`memtype=data`) in the current working directory. The dataset `junkfile` is excluded. You can use `select` instead of `exclude`, if that's more convenient.

OITUNIX

The code for PROC CPORT is the same on any operating system. You just need to specify appropriate paths and filenames:

```
LIBNAME saslib '~username/mysasdir/';  
PROC CPORT LIBRARY=saslib FILE='trans.exp';  
RUN;
```

Macintosh

On Macintosh, specify the diskname, path and filename separated by colons:

```
LIBNAME saslib 'HD:mysasdir';  
PROC CPORT LIBRARY=saslib FILE='HD:mysadir:trans.exp';  
RUN;
```

Creating a Transport File – PROC COPY

Windows

To create a transport file of all the datasets in a SAS library on the Windows using PROC COPY:

```
LIBNAME saslib 'C:\mysasdir\';  
LIBNAME tranfile XPORT 'C:\mysasdir\trans.exp';  
PROC COPY IN=saslib OUT=tranfile MEMTYPE=DATA;  
RUN;
```

You can use version 6, 8 or 9 to run this code. PROC COPY writes all the SAS datasets but not the catalogs in the current directory to file `trans.exp`. Use the MEMTYPE option to prevent warnings that catalogs cannot be transported. As with PROC CPORT, specify an engine for libraries that contain both version 6 and version 8 or 9 SAS files. Add the SELECT or EXCLUDE subcommands if needed.

```
LIBNAME saslib V6 'C:\mysasdir\';  
LIBNAME tranfile XPORT 'trans.exp';  
PROC COPY IN=saslib OUT=tranfile MEMTYPE=DATA;  
EXCLUDE JUNKFILE; RUN;
```

If you have more than one SAS library to transport, run this code in each directory. Substitute a different name for each transport file created.

OITUNIX

On OITUNIX, use the same code with unix-appropriate paths and filenames:

```
LIBNAME saslib '~username/mysasdir/';  
LIBNAME tranfile XPORT 'trans.exp';  
PROC COPY IN=saslib OUT=tranfile MEMTYPE=DATA;  
RUN;
```

Macintosh

On Macintosh, specify the diskname, path and filename separated by colons:

```
LIBNAME saslib 'HD:mysasdir';  
LIBNAME tranfile XPORT 'HD:mysasdir:trans.exp';  
PROC COPY IN=saslib OUT=tranfile MEMTYPE=DATA;  
RUN;
```

Move the Transport File

If you use ftp to move the transport file to the new system, be sure to use binary mode to avoid any conversion.

Import the Transport File – PROC CIMPORT

On the new system, make a directory to store the SAS files you are about to import. Put the transport file in this directory.

Windows

This code imports the contents of the file 'C:\mysadir\trans.exp' to the directory 'C:\mysasdir\':

```
LIBNAME saslib 'C:\mysasdir\';  
PROC CIMPORT LIBRARY=saslib INFILE='C:\mysadir\trans.exp';  
RUN;
```

OITUNIX

Use the same code with unix filenames to import the file 'trans.exp':

```
LIBNAME saslib '~username/mysasdir/';  
PROC CIMPORT LIBRARY=saslib INFILE='trans.exp';  
RUN;
```

Macintosh

Use the same code with Macintosh filenames:

```
LIBNAME saslib 'HD:mysasdir';  
PROC CIMPORT LIBRARY=saslib INFILE='HD:mysadir:trans.exp';  
RUN;
```

Import the Transport File – PROC COPY

On the new system, make a directory to store the SAS files you are about to import. Put the transport file in this directory.

Windows

This code imports the contents of the file 'trans.exp' to 'C:\mysasdir\trans.exp'

```
LIBNAME saslib 'C:\mysasdir\';  
LIBNAME tranfile XPORT 'C:\mysasdir\trans.exp';  
PROC COPY IN=tranfile OUT=saslib;  
RUN;
```

OITUNIX

Use the same code with unix filenames to import the contents of the file 'trans.exp':

```
LIBNAME saslib '~username/mysasdir/';  
LIBNAME tranfile XPORT 'trans.exp';  
PROC COPY IN=tranfile OUT=saslib;  
RUN;
```

Macintosh

Use the same code with Macintosh filenames:

```
LIBNAME saslib 'HD:mysasdir';  
LIBNAME tranfile XPORT 'HD:mysasdir:trans.exp';  
PROC COPY IN=tranfile OUT=saslib;  
RUN;
```