



CHAPTER

1

Moving and Accessing SAS Files between Operating Environments

<i>Deciding to Move a SAS File between Operating Environments</i>	3
<i>Deciding to Access a SAS File across Operating Environments</i>	3
<i>Strategies for Moving and Accessing SAS Files</i>	4
<i>Summary of Strategy Features</i>	5
<i>Using National Language Support To Move SAS Files between Computers</i>	6
<i>The Data Set Used for Examples</i>	7
<i>Naming Conventions Used for Examples</i>	7
<i>Accessibility Features in SAS Products</i>	8

Deciding to Move a SAS File between Operating Environments

Moving SAS files between operating environments is a common task. Reasons for moving a SAS file between operating environments include the following:

- to move SAS files to a new operating environment on a different computer (for example, moving HP-UX files to a RedHat Linux operating environment).
- to move a file and its processing to a high-performance operating environment and then return the file to the requesting operating environment.
- to make a static copy of a SAS file available to a physically separate operating environment for continued data processing. Files are duplicated for use in the receiving operating environment because the SAS files are not available to the receiving operating environment by means of NFS-mounted file systems.

In all of these scenarios, the move operations recognize differences between operating environment architectures and SAS releases, allowing the original files to be used in the receiving operating environment.

Deciding to Access a SAS File across Operating Environments

In some instances, accessing instead of owning and maintaining your own copy of a file might be preferable. Alternatively, you might need to read data from a locally mounted tape that was created elsewhere, or you might need to read, write, or update data that is remotely mounted on your network.

Note: Do not confuse the term *access* with the product SAS/ACCESS. In the context of moving or accessing SAS files across operating environments, *access* means to reach and process SAS files. SAS/ACCESS enables users to use third-party DBMS files. △

You can use these methods to access remote SAS files:

- CEDA (Cross-Environment Data Access) enables you to process SAS 8 and later SAS files.
- Using SAS/SHARE on your client enables you to access a remote SAS file that resides on an operating environment that a SAS/SHARE server runs under. SAS/SHARE facilitates a transparent concurrent access to remote data among multiple users. Restrictions apply to cross-release access of SAS data.
 - In addition, SAS/SHARE enables you to access certain third-party DBMS files by means of engines that are supported by SAS/ACCESS.
- Without the aid of SAS/SHARE or CEDA, you can rely upon network services for access to remote files (both SAS files and third-party DBMS files). Usually, the client and the server must share a compatible architecture, and they must run the same release of SAS software. The operating environment, the network software, and the security software might control users' permissions to access specific remote files. For more information, see the SAS companion documentation that is appropriate to your operating environment, and see the third-party documentation for the network software and security software that you use.

Strategies for Moving and Accessing SAS Files

You can use these strategies to move or access SAS files:

Cross-Environment Data Access (CEDA)

This feature of SAS enables a SAS file that was created in any directory-based operating environment (for example, Solaris, Windows, HP-UX, OpenVMS) to be processed by a SAS session that is running in another directory-based environment.

CPORT and CIMPORT procedures

In the source environment, you can use PROC CPORT to write data sets or catalogs to transport format. In the target environment, PROC CIMPORT can be used to translate the transport file into the target environment's native format.

XPORT engine with DATA step or PROC COPY

In the source environment, you can use the LIBNAME statement with the XPORT engine and either the DATA step or PROC COPY to create a transport file from a SAS data set. In the target environment, the same method can be used to translate the transport file into the target environment's native format.

Note: The XPORT engine does not support SAS 8 and later features, such as long file and variable names. \triangle

XML engine with DATA step or PROC COPY

In the source environment, you can use the LIBNAME statement with the XML engine and either the DATA step or PROC COPY to create an XML document from a SAS data set. In the target environment, the same method can be used to translate the XML document into the target environment's native format.

Data Transfer Services (DTS) in SAS/CONNECT

This feature enables you to transfer data sets and catalogs from the source environment to the target environment. DTS dynamically translates the data between operating environment representations and SAS versions, as necessary. The transfer is accomplished using the SIGNON statement to connect two SAS sessions and then PROC UPLOAD or PROC DOWNLOAD to move the data.

REMOTE engine and Remote Library Services in SAS/SHARE and SAS/CONNECT

These features give you transparent access to remote data using the REMOTE engine and the LIBNAME statement.

Summary of Strategy Features

Here is a summary of the features of each strategy that you can use to move or access SAS files.

Table 1.1 Summary of Strategy Features for Moving or Accessing SAS Files

Features	Strategies That Can Be Used					
	CEDA	PROC CPORT/ PROC CIMPORT	XPORT Engine	XML Engine	SAS/CONNECT DTS	SAS/CONNECT RLS and SAS/SHARE RLS
SAS Member Types Supported	Data File, PROC SQL views*, SAS/ACCESS views (Oracle and Sybase), MDDB*	Library, Data Set, Catalog, Catalog entry	Library, Data Set	Data File	Library, Data Set, Catalog, Catalog entry, PROC SQL view, MDDB, External third-party databases***	Library, Data Set, Catalog**, Catalog entry**, PROC SQL view, MDDB, DATA Step view, SAS/ACCESS view, External third-party databases***
Dynamic Translation or Create a File Format	Dynamic	Transport****	Transport****	XML	Dynamic	Dynamic
SAS Versions Supported	SAS 8 and later	SAS 6 and later	SAS 6 and later****	SAS 8.2 and later	SAS 6 and later	SAS 6 and later
Regression from a Later to an Earlier SAS Release	No	No	Yes	No	Yes	Yes

Strategies That Can Be Used						
Features	CEDA	PROC CPORT/ PROC CIMPORT	XPORT Engine	XML Engine	SAS/CONNECT DTS	SAS/CONNECT RLS and SAS/SHARE RLS
Limited to Operating Environments that Use Directory- Based File Structures	Yes	No	No	No	No	No
SAS Product License Required	Base SAS	Base SAS	Base SAS	Base SAS	SAS/CONNECT	SAS/CONNECT or SAS/SHARE

* Data set (files) can have read, write, and update access. PROC SQL views and MDDBs are read-only.

** SAS 9 does not support cross-operating environment access to catalog entries or catalogs in operating environments that are incompatible. For information about architecture groups, see the *SAS/CONNECT User's Guide* or *SAS/SHARE User's Guide*.

***SAS/CONNECT supports external text files and binary files. SAS/CONNECT and SAS/SHARE support third-party external databases by means of the Remote SQL Pass-Through Facility, but you must have a SAS/ACCESS license to access these databases.

****The XPORT engine does not support features that were introduced in SAS 8 (such as long file and variable names). If the XPORT engine is used to regress a SAS 8 or later SAS file to an earlier release, the features that are exclusive to SAS 8 and later are removed from the SAS file. Also, the transport formats that are produced by the XPORT engine and PROC CPORT are not interchangeable.

For complete details about relational databases, see *SAS/ACCESS for Relational Databases: Reference*. For details about nonrelational databases, see *SAS/ACCESS Interface to CA-Datcom/DB: Reference*, *SAS/ACCESS Interface to IMS-DL/I Software*, *SAS/ACCESS DATA Step Interface to CA-IDMS: Reference*, or *SAS/ACCESS Interface to SYSTEM 2000: Reference*, as appropriate.

Using National Language Support To Move SAS Files between Computers

In order to successfully move a transport file between two computers and operating environments, the encodings of the source and target SAS sessions must be compatible. For example, a source SAS session that uses the Wlatin1 encoding that is associated with the Spanish Mexico locale is compatible with the target SAS session that uses Wlatin1 encoding that is associated with the Italian Italy locale. Both sessions use the Wlatin1 encoding.

However, a transport file cannot be moved between incompatible source and target SAS sessions without national language support (NLS). For example, a source SAS session that uses the Wlatin2 encoding that is associated with the Czech Czechoslovakia locale is incompatible with the target SAS session that uses the open_ed-1141 z/OS encoding that is associated with the German Germany locale. The Wlatin2 encoding and the open_ed-1141 encodings are not compatible.

Before the data can be moved using the appropriate strategy, (for example, the XPORT engine or PROC CPORT and PROC CIMPORT), you would have to re-set the locale of the target SAS session to the locale of the source SAS session that created the

transport file. Strategies for specifying locale or encoding vary according to the version of SAS that is running on the source and target computers.

If you are moving SAS files across locales or encodings, you will use the `LOCALE=` and `ENCODING=` options. For this information, see the *SAS National Language Support (NLS): Reference Guide*. For details about using `PROC CIMPORT` to move transport files between source and target computers that use different locales and encodings, see the *Base SAS Procedures Guide*.

The Data Set Used for Examples

If you choose to experiment, you can create several simple data sets in a library. Here is a sample SAS program that creates the data set `GRADES`:

```
data grades;
  input student $ test1 test2 final;
  datalines;
Fred 66 80 70
Wilma 97 91 98
;
proc print data=grades;
run;
```

Here is the output:

```

The SAS System          10:59 Friday, April 25, 2008

Obs   student   test1   test2   final
1     Fred      66      80      70
2     Wilma     97      91      98
```

Naming Conventions Used for Examples

These naming conventions are used in the examples in this documentation:

WORK

is the default libref that points to the library that contains the data set `GRADES`.

XPORTOUT

is the libref that points to the location where the transport file is created with the `XPORT` engine.

XPORTIN

is the libref that points to the location on the target operating environment that you transferred the transport file to.

XMLOUT

is the libref that points to the location where the XML file is created with the `XML` engine.

XMLIN

is the libref that points to the location on the target operating environment that you transferred the XML file to.

CPORTOUT

is the fileref that points to the location where the transport file is created with `PROC CPORT`.

IMPORTIN

is the fileref that points to the location on the target operating environment that you transferred the transport file to.

SOURCE

is the libref that points to the location of the source file that is translated into transport or XML format.

LIST

is a catalog entry type.

GRADES

is the name of a data set.

TARGET

is the libref that points to the location at which the restored SAS file is created.

TESTCAT

is the name of a catalog.

TESTNPGM

is the name of a catalog entry.

Accessibility Features in SAS Products

For information about accessibility for any of the products mentioned in this book, see the documentation for that product. If you have questions or concerns about the accessibility of SAS products, send e-mail to accessibility@sas.com.