APPENDIX 1

Cross-Architecture Access

Introduction

This appendix describes the cross-architecture capability in SAS/SHARE software. Host differences, host limitations, data translation, and host compatibility are provided. A summary of the communication options that are available between different hosts and how they are grouped according to data representations is also provided.

Audience

End-users who may be unaware of SAS/SHARE software but who use it to update shared data across hosts that have different architectures.

Programmers who develop SAS applications that access shared data when the applications execute on one host, and the shared data resides on another host that has a different architecture.

Server administrators who create and maintain SAS servers that are accessible from hosts that have different architectures.

Capabilities

Cross-architecture access is a feature of SAS/SHARE software that enables a server SAS session and its client sessions to execute on machines that have different architectures. That is, a server and its clients may execute on machines that have different internal representations of data such as the IBM System/390 and Intel 486, or Digital Equipment Alpha VMS and Hewlett-Packard Precision Architecture. For details about cross host connections, see *Moving and Accessing SAS Files across Operating Environments*.

Cross-architecture access enables you to move data or applications from one type of host to another. For example, an application on UNIX that uses a SAS library on OS/390 issues a LIBNAME statement just as an OS/390 application does, by specifying the OS/390 physical name for the SAS library and the name of the OS/390 server. For example, you can

□ move a SAS application from OS/390 to UNIX, leave the data on OS/390, and continue to access the data without changing the application.

□ move an application's data from UNIX to OS/390, leave the application on UNIX, and change only the LIBNAME statement that accesses the data.

□ duplicate the application on both hosts and simultaneously access the data on OS/390.

Cross-architecture access enables users to read and write SAS data across host architecture boundaries. It enables applications running on one type of host to read DBMS data that are accessed through server-managed SAS/ACCESS views when that DBMS is available only on another type of host. For example, a SAS session on a Sun workstation can use a SAS/ACCESS view to read the contents of a DB2 table on an OS/390 host. For more information about using views under SAS/SHARE software, see “SAS Data View Programming Considerations” on page 53.

SAS/SHARE software is particularly well-suited to applications that

□ require access to a single record at a time.

□ use a WHERE clause to subset large data sets.

□ execute procedures against small data sets.

An application that processes large quantities of data, especially through multiple passes, may benefit from moving a copy of the data to the computer on which it executes, or from using SAS/CONNECT software to remotely execute SAS on the computer on which the data is stored.

Neither Version 7 nor Version 8 SAS/SHARE software supports access to some types of SAS files, such as SAS catalogs, when the architecture of the server machine differs from the architecture of the client machine. See “Host Differences” on page 146 and “Restrictions and Limitations” on page 147 for more information.

Host Differences

Access is said to be cross-architecture when the client session and the server session are running on machines that have different internal representations of data. Differences in internal representation of data can arise because of hardware differences between two machines. For example, IBM mainframe machines represent floating-point numbers differently than computers that are based on Intel CPUs, which represent them differently than Digital VAX computers. The codes for internal representation of character data also vary; EBCDIC and ASCII are two major character-encoding systems.

Different operating systems and C-language compilers also cause differences in representation. Differences are due to different alignment requirements of aggregate data types, such as the inter-element padding in a particular C structure. Also, two compilers for the same type of CPU may implement simple data types with differing lengths.

It may not always be obvious when the cross-architecture features of SAS/SHARE software are required. OS/390 to CMS access is not cross-architecture because the underlying representation of data in the two operating systems is the same. On the other hand, sharing data between OpenVMS for VAX and OpenVMS for AXP does constitute cross-architecture access because data are represented differently on the Digital VAX and Alpha AXP architectures, even when the same operating system is used. For complete details about architectural compatibility, see "Identical Architectural Groups" on page 151.

Restrictions and Limitations

Catalog Access

When the user and server run cross-architecture, access to several types of data that is normally available through SAS/SHARE software is not supported, including catalog access. This means that you can use the FSEDIT procedure to update a SAS data set on a different host type, but you cannot specify a SCREEN entry for the procedure from that server. Also, you can neither retrieve a SAS/AF PROGRAM entry for execution nor update it with the BUILD procedure. The same is true for a user-written format, which is stored in a catalog entry of type FORMAT.

Other SAS File Access

With the exception of PROC SQL views, SAS view files (type VIEW) cannot be directly accessed across architectures. A DATA step or SAS/ACCESS view can be read cross-architecture as long as it is interpreted in the server session. In this case, you must not specify the RMTVIEW=NO option in the client LIBNAME statement because that option requests interpretation in the client SAS session and requires the SAS view file itself to be transmitted to the client session. See "SAS Data View Programming Considerations" on page 53.

Some common operations cannot be performed across architectures because they require direct access to SAS view files. Direct access to PROC SQL views is provided; however, direct access is not available across architectures for DATA step or SAS/ACCESS views. For example, you cannot use the COPY procedure to copy a SAS file with a file type of VIEW to a host with a different architecture. However, you can use the SQL procedure's DESCRIBE VIEW statement because direct access to SQL views is provided.

You should also note that access descriptor files (type ACCESS) cannot be accessed across architectures. Access descriptors are special files that are produced and used by SAS/ACCESS products to describe data in other vendors' databases, such as DB/2 or ORACLE. Cross-architecture interpretation of a SAS/ACCESS view is supported, but direct access to the descriptor file is not. Thus, you cannot use PROC ACCESS to create a SAS access descriptor file on a computer of a different architecture.

SAS files of type PROGRAM cannot be accessed across architectures. These files contain compiled DATA step code. You cannot execute such a DATA step in your local SAS session by using the PGM= option in the DATA step, nor can you write a DATA step PROGRAM entry to a cross-architecture server. A DATA step PROGRAM entry can be executed in a cross-architecture server session if it is referenced by a DATA step view that is interpreted there.

Short Numerics and Mixed-Type Variables

In order for SAS data sets to be accessed across architectures, they should not include two-byte numeric variables. This length is allowed on IBM mainframe machines, but other hosts on which SAS software runs have a minimum numeric variable length of three. As a result, a data set that contains a two-byte numeric cannot be accessed across architectures from other types of hosts.

With clever programming in the DATA step, it is possible to store numeric values in character variables and character values in numeric variables. However, you should not construct data sets in this fashion if you want to access them across architectures. SAS/SHARE software performs appropriate character translation of character variables and numeric translation of numeric variables when crossing architectures. Because of this, numeric values that are stored in character variables and character values that are stored in numeric variables that the programmer expects will not be preserved. SAS/SHARE software has no means to detect such usage.

Implications of Data Translation

Other parts of SAS translate data from one representation to another. The CPORT and CIMPORT procedures translate data into and from a single representation system called *transport* format. The DOWNLOAD and UPLOAD procedures of SAS/CONNECT software each perform this "to-transport/from-transport" translation as they move data between hosts that have dissimilar architectures.

In SAS/SHARE software, translation of numeric variables occurs when the server machine and the client machine represent floating-point numbers differently. For character variables, translation occurs when their character representations differ. Values are translated directly from the source representation to the target representation; they do not pass through transport format. Translation occurs both when data flows from the server to the client and when it flows from the client to the server. Therefore, data that flows across architectures from a server to a client and that are then sent back to that server undergo two translations.

For all hosts on which SAS/SHARE software currently runs, the remote engine performs all data conversion on behalf of itself and the server. Thus, it converts outgoing data to the server format, and it converts incoming data from the server to its own format. The administrative procedure, PROC OPERATE, works in the same way. However, the server does all data conversion for clients other than SAS, such as the SAS ODBC driver.

Numeric Translation

Translation from one numeric representation to another can alter the value of a variable. A common type of alteration is loss of precision. This occurs when the source representation uses more bits to represent the mantissa than the target representation. Such distortions always have a very small magnitude, but they might represent a significant percentage change to a very small original value.

CAUTION:

SAS/SHARE software does not produce any warnings about loss of precision during translation. △

A rare type of value distortion is loss of magnitude. This occurs when the source representation has a greater exponent range than the target representation and a value whose magnitude lies in the excess range of the source representation is translated. Of course, the magnitude of this type of distortion is potentially very great, as is the percentage change. SAS/SHARE software does produce a warning when this type of alteration occurs.

See *SAS Language Reference: Dictionary* for a detailed description of the numeric representation of SAS variable values.

Character Translation

Character data is not altered by the translation process when the translation between the two representation schemes is one-to-one. Current Institute-supplied translation tables provide one-to-one translation between the five character representations that SAS recognizes: EBCDIC, ASCII-OEM, ASCII-ANSI, ASCII-ISO, and ASCII-Mac.

Use caution if the client and server machines are configured for different natural languages, such as Swedish and English. Translation tables on each machine are typically customized to best support the native language. This provides the best mapping of the native language to the coding system, but it makes translation alteration more likely.

A site or individual user can freely modify their translation tables. In such cases, the modifications must be checked carefully to ensure that the new translation is still one-to-one, or to verify that alteration of character data through updating cannot occur (or is not important). See “Character-Translation Tables” on page 149 for information about translation tables.

Character-Translation Tables

The tables that are used for character translation in SAS/SHARE software are stored in SAS catalog entries of type TRANTAB. Each of these catalog entries contains two translation tables. The first is for import translation and the second is for export translation. For example, the EBCDIC/ASCII-OEM translation entry on OS/390 contains an import table for ASCII-OEM to EBCDIC translation and an export entry for EBCDIC to ASCII-OEM translation. The names of the catalog entries that contain these translation tables are given in the following table.

Translation Table Set	**Catalog Entry Name**
EBCDIC/ASCII-ISO	_0000030
EBCDIC/ASCII-ANSI	_0000060
EBCDIC/ASCII-OEM	_00000A0
EBCDIC/ASCII-MAC	_0000120
ASCII-ISO/ASCII-ANSI	_0000050
ASCII-ISO/ASCII-OEM	_0000090
ASCII-ISO/ASCII-MAC	_0000110
ASCII-ANSI/ASCII-OEM	_00000C0

Translation Table Set	Catalog Entry Name
ASCII-ANSI/ASCII-MAC	_0000140
ASCII-OEM/ASCII-MAC	_0000180

Character-translation catalog entries are stored in the SASUSER.PROFILE and SASHELP.HOST catalogs. The translation process locates a given translation entry by searching first the SASUSER.PROFILE catalog and then the SASHELP.HOST catalog.

The remote engine performs all data conversion on behalf of itself and the server. Therefore, only the translation tables in the client SAS session are used. However, the server translation tables are used in data conversion for clients other than SAS because the server performs all data conversion on behalf of itself and these clients.

SAS site administrators can use the TRANTAB procedure to replace or update the translation tables. See the *SAS Procedures Guide* and *SAS Language Reference: Dictionary* for details.

CAUTION:

Do not attempt to update a translation table in a client session while connected across architectures to a host that requires its usage. You cannot ensure that the new version of the table will be used for subsequent conversions. △

Note: The default character set for Microsoft Windows hosts is ASCII-ANSI, so the supplied _00000C0 TRANTAB entry contains an import translation table for ASCII-OEM to ASCII-ANSI and an export table for ASCII-ANSI to ASCII-OEM. If the WINCHARSET SAS system option is used to switch the character set to ASCII-OEM, the _00000C0 TRANTAB entry must be updated to switch the import and export tables if they differ. △

Data Translation Considerations

Data translation in SAS/SHARE software may have some implications that users need to consider. For example, suppose that a user has assigned two SAS data libraries, FOO and ZOO, through a server on a host that has a different architecture from the user's machine. The user copies the data sets contained in FOO to ZOO:

```
proc copy in=foo out=zoo mt=data;
run;
```

The contents of the copied data sets in ZOO are not guaranteed to be identical to the contents of the original data sets in FOO. The data sets in ZOO have undergone two translations from the originals in FOO. This may have resulted in a slight change in the precision of numeric variables in the data sets in ZOO.

For another example, suppose a user is using the FSEDIT procedure to edit a data set across architectures. A user who enters a DUP command and who then modifies variable X before saving the new record may find that, aside from the value of variable X, the new record is not identical to the old record. The original values of the duplicated record have undergone two translations, from server-machine format to user-machine format and back, while the new value that the user entered for the variable X has undergone only one translation from user-machine format to server-machine format.

Note: When editing or updating a data set across architectures by using the FSEDIT procedure, the FSVIEW procedure, DATA step MODIFY statement, and so forth, any variables that are not updated in an updated observation will be exempt from translation and will be unaltered. △

Identical Architectural Groups

The following lists show the Version 6, Version 7, and Version 8 hosts on which SAS/SHARE software runs. The lists are grouped on the basis of identical data representation. No data conversion is required in communications between any two hosts in the same group. When two hosts that are not in the same group communicate, conversion of at least one data type is required, and the restrictions and limitations described earlier in this chapter apply.

The IBM System/390 architecture hosts are:

□ CMS

□ OS/390

□ VSE

Note: VSE is a Version 6 host only. △

The UNIX RISC hosts are:

□ AIX for data sets and catalogs

Note: Version 6 AIX hosts do not support catalogs. △

□ HP-UX

□ MIPS ABI

□ Solaris

□ SunOS

□ SGI

The Windows 32-bit hosts are:

□ Windows NT

□ Windows 95

□ Windows 98

□ Windows 32s

Note: Windows 32s is a Version 6 host only. △

The Macintosh hosts are:

□ Macintosh PPC

□ Macintosh 68k

Note: Macintosh is a Version 6 host only. △

The hosts that are incompatible with all other hosts and each other are:

□ Compaq Tru64 UNIX (formerly Compaq's DIGITAL UNIX)

□ Intel ABI

□ IABI

□ OpenVMS Alpha

□ OpenVMS VAX

□ OS/2

Numeric Architectural Groups

The following lists show the hosts on which SAS/SHARE software runs. The lists are grouped on the basis of like numerical representation. No numeric (floating-point) conversion is required in communications between any two hosts in the same group. When two hosts in different groups communicate, numeric conversion is required, and the restrictions and limitations concerning numeric conversion apply.

The IBM System/390 format hosts are:

□ CMS

□ OS/390

□ VSE

Note: VSE is a Version 6 host only. △

The IEEE-format, standard missing-value hosts (big endian) are:

□ AIX

□ HP-UX

□ Macintosh PPC

□ Macintosh 68k

□ MIPS ABI

□ OpenVMS Alpha

□ Solaris

□ SunOS

□ SGI

Note: Macintosh is a Version 6 host only. △

Actually, some conversion of numeric data is necessary between some of these hosts due to different byte-ordering or alignment requirements. However, because they all use the same number of exponent and mantissa bits, loss of precision or range does not occur.

The IEEE format, standard missing-value hosts (little endian) hosts are:

□ Compaq Tru64 UNIX (formerly Compaq's DIGITAL UNIX)

□ Intel ABI

□ IABI

The IEEE format, PC missing-value hosts are:

□ OS/2

□ Windows NT

□ Windows 95

□ Windows 98

□ Windows 32s

Note: Windows 32s is a Version 6 host only. △

In a class of its own because of its unique numeric representation is:

□ OpenVMS VAX

Character Architectural Groups

The following lists show the hosts on which SAS/SHARE software runs. The lists are grouped on the basis of like character representation. No character conversion is required when communicating between any two hosts in the same group. When two hosts in different groups communicate, character conversion is required, and the restrictions and limitations concerning character conversion apply.

The EBCDIC format hosts are:

□ CMS

□ OS/390

□ VSE

Note: VSE is a Version 6 host only. △

The ASCII-ISO format hosts are:

□ AIX

□ Compaq Tru64 UNIX (formerly Compaq's DIGITAL UNIX)

□ HP-UX

□ Intel ABI

□ MIPS ABI

□ OpenVMS Alpha

□ OpenVMS VAX

□ Solaris

□ SunOS

□ SGI

□ IABI

The ASCII-ANSI format hosts are:

□ Windows NT

□ Windows 95

□ Windows 98

□ Windows 32s

Note: Windows 32s is a Version 6 host only. △

The ASCII-OEM format host is:

□ OS/2

The ASCII-MAC hosts are:

□ Macintosh PPC

□ Macintosh 68k

Note: Macintosh is a Version 6 host only. △

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/ SHARE User's Guide, Version 8*, Cary, NC: SAS Institute Inc., 1999. pp. 247.

SAS/SHARE User's Guide, Version 8

ISBN 1–58025–478–0

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, September 1999

The Institute is a private company devoted to the support and further development of its software and related services.