SAS® Data Libraries: V6 to V7 Compatibility (or, will Version 7 be able to access my Version 6 files?)

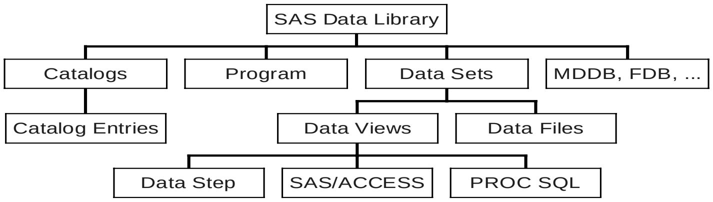
James Holman, SAS Institute Inc.

Abstract

This paper presents an overview of backwards and forwards compatibility between Versions 6 and 7 of the SAS System and is intended to be used as a reference in determining what manner of access is allowed. Furthermore, this paper takes into account differences in compatibility between the Base SAS System and when products such as SAS/CONNECT® and SAS/SHARE® are used.

SAS I/O Services

The I/O Services provided by the SAS System are essentially the same in Version 7 as they were in Version 6. The fundamental component of I/O Services is the ability to access a SAS Data Library and its members. A SAS Data Library can be visualized as follows:



SAS Data Libraries

The two major goals of Version 6 to Version 7 compatibility are to allow the most transparent access possible to Version 6 data and to prevent Version 7 data from being written to Version 6 data libraries, as that would render them corrupt and unusable by Version 6.

Accessing Version 6 Libraries

Version 6 libraries can be accessed in Version 7 as follows:

```
libname v6lib ’path-to-v6-library’;
```

where ’path-to-v6-library’ contains only Version 6 members. By using the same syntax used in Version 6, Version 7 of the SAS System is able to run existing SAS programs with ***no change***. The engine used will be determined by the nature of the library; if the library is Version 6 (i.e., no Version 7 files are present) the Version 6 engine will be used. Due to the fact that the SAS System has to determine the engine, directly specifying the desired engine can save time in the libname assignment, as seen below:

```
libname v6lib v6 ’path-to-v6-library’;
```

Versions 6 and 7 Data Files in the Same Library

On directory based hosts, Version 6 and Version 7 files have a different extension for data files, catalogs, views, etc. As a result, the SAS System is able to differentiate between the two. For example, consider the SAS System running under a UNIX operating system:

Version 6	Version 7
MYDATA.SSD01	MYDATA.SAS7BDAT
MYCAT.SCT01	MYCAT.SAS7BCAT
MYVIEW.SSV01	MYVIEW.SAS7BVEW

On hosts that do not support the concept of directories such as MVS, SAS data libraries are considered to be **bounded libraries**. Thus, on these hosts, the SAS System is able to differentiate between Version 6 and Version 7 libraries due to the different attributes each type of library possesses.

In Version 7, it is very possible for Version 6 and 7 files to be intermingled in the same library or directory on directory based hosts. This is commonly referred to as a **mixed-mode** and is dealt with as follows:

- SAS defaults to the Version 7 Engine.
- The Version 6 Engine can be used when assigning a library reference (as seen above) to access Version 6 files.

A new Version 7 feature, *library concatenation*, can be used to create a single library reference for Version 6 and Version 7 files. Some new features available in Version 7 are:

- Longer names are allowed (e.g. data file variable names and catalog entry names can be up to 32 characters)
- Library and Catalog Concatenation
- Integrity Constraints for Data Files

Many other new features have been added to Version 7; for an overview of them, see *Sometimes You Get What You Want: SAS® I/O Enhancements for Version 7* by Steve Beatrous and Billy Clifford, SAS Institute Inc.; it can be found in the *SUGI 24 Conference Proceedings*.

Compatibility of SAS Data Library Members

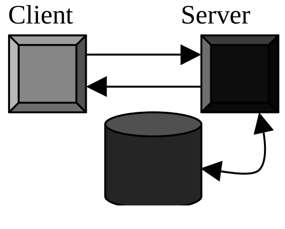
Compatibility of the SAS Data Library members depends mainly on the SAS data access product being used to access them. For the purpose of the following discussion, the following divisions will be used:

Traditional Single User SAS

The base and compatibility engines are the only engines being used, no SAS/SHARE® or SAS/CONNECT® server is running, and there is only one session of the SAS System in use.

SAS/SHARE®

SAS/SHARE® consists of two scenarios: a Version 6 client accessing a Version 7 SAS/SHARE® server and a Version 7 client accessing a



Version 6 SAS/SHARE® server.

Data

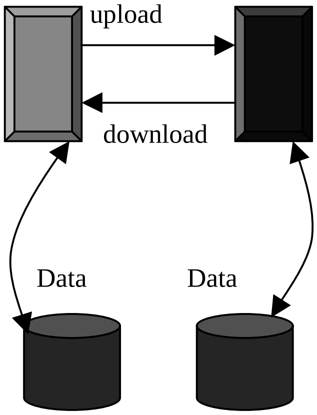
The server allows the client to access data it contains as seen in the diagram.

SAS/CONNECT®

SAS/CONNECT® uses the conceptual client/server model.

The following are interchangeable:

Source / Target
Client / Server
Local / Remote



For clarity, local/remote will be used throughout, unless used in conjunction with SAS/SHARE®, in which case, client / server will be used.

SAS/CONNECT® consists of:

- Remote Library Services (RLS)
- Remote Data Transfer Services
- Remote Compute Services

Data is shared back and forth between local and remote hosts, as seen above.

Compatibility of SAS Data Libraries and their members will be discussed in the following sections using the above three SAS configurations.

SAS Data Sets - Data Files

A *SAS data file* is a collection of data values (observations) and their associated descriptive information (variables) arranged and presented in a form that can be recognized and processed by the SAS System. Furthermore, the member type of a SAS data file is **DATA** within a SAS Data Library. Logically, SAS data files contain data values organized into a rectangular structure of rows and columns and descriptive information that identifies characteristics of the data file itself as well as those of the data values.

Traditional Single User SAS

Access to Version 6 data files from Version 7 and vice versa is described in the following table:

	Version 6 Data File	Version 7 Data File
Version 6 SAS Session	Read/Write/Update	■
Version 7 SAS Session	Read/Write/Update[†]	Read/Write/Update

[†] No Version 7 specific features are allowed

A Version 7 SAS session is able to manipulate Version 6 data files, hence a Version 6 data file is said to be *forwards compatible* with Version 7. The only restriction a user faces is that they are not allowed to use Version 7 specific features when modifying a Version 6 data file.

For example, in Version 7:

```
libname v6lib 'path-to-v6-library';
data v6lib.test; TooLongForV6Name = 3; run;

ERROR: The variable name TooLongNameForV6 is
illegal for file V6LIB.TEST.DATA.

NOTE: The SAS System stopped processing this step
because of errors.

WARNING: The data set V6LIB.TEST was only
partially opened and will not be saved.
```

This data step resulted in an error because the variable *TooLongForV6Name* has a length greater than 8 bytes, something not supported by the V6 engine attempting to create the data file. Hence, data created in version 7 is not *backwards compatible* to Version 6.

However, again in Version 7, if a valid Version 6 variable name is used:

```
libname v6lib 'path-to-v6-library';
data v6lib.test; x = 3; run;

NOTE: The data set V6LIB.TEST has 1 observations
and 1 variables.
```

A Version 6 session of SAS could now be invoked that would be capable of reading v6lib.test.data as if Version 6 and not Version 7 had originally created it.

A user can continue like this indefinitely, using Version 7 to create Version 7 data files with new features and manipulating Version 6 data files when needed. However, this type of user will not be able to fully maximize the potential of Version 7 until converting their Version 6 data files into Version 7 format. When such a conversion is completed, usually with PROC COPY, the user would be able to use all of Version 7's rich feature set.

SAS/SHARE® and SAS/CONNECT® RLS

Compatibility of data access of SAS Data Files using SAS/SHARE® or Remote Library Services in SAS/CONNECT® is broken down into the following two cases:

- Version 6 Client → Version 7 Server
- Version 7 Client → Version 6 Server

Both these cases are described in the following table:

	Version 6 Server	Version 7 Server

	Version 6 Data File	Version 7 Data File	Version 6 Data File	Version 7 Data File
V6 Client	R/W/U		R/W/U	R/W/U†
Version 7 Client	R/W/U†		R/W/U†	R/W/U

† No Version 7 specific features are allowed

As seen above, the potential for mixed-mode libraries exist in Version 7 of the SAS System. When a Version 6 client attempts to access Version 6 data files on a Version 7 server, and the library is of the mixed-mode type, i.e., Version 7 data files are present as well as Version 6 data files, the library must be assigned using the RENGINE= option.

Suppose mixedlib has been assigned on the Version 7 server and that it is a mixed-mode library. Furthermore, assume that the following is executed on a Version 6 client:

```
libname mixedlib rengine=v6 ’path-to-mix-library’
  server=hostname.shr1;
data mixedlib.test; x = 3; run;

NOTE: The data set MIXEDLIB.TEST has 1
observations and 1 variables.
```

This allows the Version 6 client to specify that the V6 engine will be used on the server instead of the default engine for Version 7, the Version 7 engine. With this default, and the fact that Version 6 and Version 7 have different extensions, the Version 7 engine would not be able to see the Version 6 data files. Thus, the RENGINE= option should be specified at the time the library reference is defined when a Version 6 client wants to access Version 6 data files on a Version 7 server, when the aforementioned data files are contained in a mixed-mode library. Note however that when using the RENGINE= option, the path for the library must be supplied.

As stated previously, one of the goals of Version 6 to Version 7 compatibility is to prevent Version 7 data from being written in a Version 6 library. As a result, Version 7 of SAS/SHARE® and Version 7 Remote Library Services under SAS/CONNECT® prevent Version 7 data files being created on a Version 6 server or remote host. However, a Version 7 client or local host is allowed to manipulate Version 6 data files provided that it does not use any new Version 7 specific feature, such as long variable names.

SAS/CONNECT®

Remote Data Transfer Services

Once again, compatibility between Version 6 and Version 7 is essentially seamless.

When a Version 6 local host accesses a Version 7 remote host, all necessary data format conversions are performed without any intervention by the user. PROC UPLOAD produces a Version 7 data file from the files transferred by the Version 6 local host. PROC DOWNLOAD provides full support for the transfer of Version 6 data files residing on the Version 7 remote host. However, transfer of Version 7 data files to the Version 6 local host is permitted only for those Version 7 data files that do not contain Version 7 specific features.†

When a Version 7 local host accesses a Version 6 remote host, PROC UPLOAD is able to transfer Version 7 data files to the remote system, provided once again that the Version 7 data files are not using Version 7 specific features. † PROC DOWNLOAD provides full support and will transfer Version 6 data files to the local system.

For example,

```
signon v7host;

rsubmit;

  data employee;
    LastName = ’Smith’;
    FirstName = ’John’;
    EmployeeID = 50;
  run;

  proc download data=employee out=empdata; run;

endrsubmit;
```

The resulting Version 6 data file, **empdata**, will contain the variables **LASTNAME**, **FIRSTNAM**, and **EMPLOYEE** and will be usable by any Version 6 or Version 7 application or SAS program.

Remote Compute Services

The remote submitting of SAS programs to and from a Version 6 and a Version 7 system that accesses data files differs very little from remote submitting between two Version 6 systems, with the following exceptions:

- A Version 6 local host can remote submit a SAS program that references either Version 6 or Version 7 data files to a Version 7 remote host with no complications.
- A Version 7 local host can remote submit a SAS program to a Version 6 remote host but the programs must not contain any references to Version 7 specific features. The reason is that the Version 6 SAS System executing the remotely submitted program has no facility to process any Version 7 specific features. For example, the new PROC CONTENTS option, **varnum,** could not be used in a job remote submitted by a Version 7 local host to a Version 6 remote host as that option would not be recognized by the Version 6 PROC CONTENTS.

SAS Data Sets - Data Views

Three types of views will be addressed here:

- SAS Data Step Views

  SAS Data Step Views are data step programs that can read from one or more SAS data files.

- SAS/ACCESS® Views

  SAS/ACCESS® Views are SAS data views designed to create an interface to data formatted by other software products, such as ORACLE®. When a SAS/ACCESS® View is processed, the data

† It is important to note that PROC UPLOAD and PROC DOWNLOAD are able to create a truncated name for variables that use the new Version 7 feature of long names. This truncation is based on the stem of the variable name and is guaranteed to be unique, though not as descriptive as the long name itself. This same truncation can be achieved with PROC COPY, provided the VALIDVARNAME=V6 option has been set.

contained within remains in its original format.

- PROC SQL Views

  PROC SQL Views are views whose data values have been obtained from one or more SAS data files or SAS data views.

Traditional Single User SAS
Access to Data Step Views, SAS/ACCESS® Views, and PROC SQL Views is restricted as follows:

- Version 7 accessing Version 6 Data Views

  - Data step views are neither forwards or backwards compatible, i.e., Version 7 cannot read/write/update Version 6 data step views. As a result, data step views must be recreated in Version 7.

  - SAS/ACCESS® views are forwards and backwards compatible, i.e., Version 7 can read/write/update a Version 6 SAS/ACCESS® view.

  - PROC SQL views are forwards compatible in that Version 7 can read a Version 6 PROC SQL view but it cannot write or update one.

- Version 6 accessing Version 7 Data Views

  - Version 7 views are not backward compatible due to differences in the SAS Data View file format.

  - Version 6 cannot access any Version 7 Data Views, i.e., Version 6 cannot read/write/update Version 7 Data Step, SAS/ACCESS®, and PROC SQL views.

Since Version 6 data step views cannot be accessed by Version 7 of the SAS System, users will need to recreate their views. Once they do so, Version 7 will be able to process them.

SAS/SHARE® and SAS/CONNECT® RLS
The following table summarizes the restrictions on access to data step and PROC SQL views by SAS/SHARE® and the Remote Library Services of SAS/CONNECT®:

	Version 6 Server		**Version 7 Server**	
	V6 View	V7 View	V6 View	V7 View
V6 Client	R/W/U	■	R/W/U[†]	R[‡]
V7 Client	R[‡]	■	R[±]	R/W/U

[†] RMTVIEW=NO has been set
[‡] RMTVIEW=YES has been set
[±] PROC SQL views only

Through the use of the RMTVIEW option, SAS/SHARE® and the Remote Library Services of SAS/CONNECT® are able to specify where the view should be processed. The default setting of YES implies that the view should be processed on the server whereas a setting of NO implies that the view is processed on the client.

For example, suppose we have a Version 7 client trying to read a Version 6 view on a Version 6 server. The following is executed on the client and the library v6lib exists on the server and contains myview:

```
libname v6lib server=shr1; /* RMTVIEW=YES */

proc print data=v6lib.myview;run;

The SAS System     1   13:12 Thursday, July 2, 1998

                                   Obs    X    Y

                                     1    1    2
```

As RMTVIEW=YES is the default setting, the view v6lib.myview is processed on the Version 6. As a result, the server transmitted the observations referenced by v6lib.myview back to the Version 7 client, which in turn was able to display them using PROC PRINT.

Note however, access to SAS/ACCESS® views differs from that allowed for data step and PROC SQL views. The following table illustrates those differences.

	Version 6 Server		**Version 7 Server**	
	V6 View	V7 View	V6 View	V7 View
V6 Client	R/W/U	■	R/W/U	R/W/U
V7 Client	R/W/U	■	R/W/U	R/W/U

The only restriction in this case is that a Version 6 server would not be allowed to contain Version 7 SAS/ACCESS® Views as Version 7 data is not allowed to be contained in a Version 6 SAS Data Library.

SAS/CONNECT®
As with SAS/SHARE®, SAS Views affected by SAS/CONNECT® include data step views, SAS/ACCESS® views and PROC SQL views. It is important to note that SAS/CONNECT® uses the data that the view references, not the view itself.

Remote Data Transfer Services
Access to Data Step Views, SAS/ACCESS® Views, and PROC SQL Views is restricted as follows:

- Version 6 local → Version 7 Remote

  PROC UPLOAD and PROC DOWNLOAD are fully supported and are capable of transferring views using the DATA= option to specify the data associated with the view that is being transferred. Alternatively, if the user desires to transfer the contents of an entire library, that library can be specified with the use of the INLIB= option.

- Version 7 local → Version 6 Remote

  PROC UPLOAD allows the Version 7 local host to upload the data associated with a view to a Version 6 remote host, provided that data does not contain Version 7 specific features.[‡]

Consider the following example…

```
/* Create local side (Version 7) data file. */

data ten;
  do i=1 to 10;
    x=i*2.5;
    num=put(i,2.);
    output;
  end;
run;

/* Reference the newly created data file by */
/* creating a Version 7 PROC SQL view. */
proc sql;
  create view five as select * from ten where i<=5;
quit;

/* Connect to the Version 6 remote host. */
signon v6host;

rsubmit;

  /* SQL view five will be transferred as a data */
  /* set because "V7 SQL view file format is not */
  /* compatible with V6 library." */
  proc upload inlib=work outlib=work status=no
            mt=view;
  run;

endrsubmit;
                              (continued)

rsubmit;

  proc sql;
    create view six as select * from ten
            where i < 7;
  quit;

  /* SQL view six will be downloaded to the V7 */
  /* local. */
  proc download inlib=work outlib=work status=no
             mt=view;
  run;

endrsubmit;
```

The above SAS program illustrates the differences in compatibility of Version 6 and Version 7 PROC SQL Views when accessed through SAS/CONNECT® using Remote Data Transfer Services. As the Version 7 PROC SQL view is not compatible with Version 6, the data it references will be transferred. However, as the Version 6 format of the PROC SQL view is compatible with Version 7, it will be transferred as a view.

Remote Compute Services

Access to Data Step Views, SAS/ACCESS® Views, and PROC SQL Views is restricted as follows:

- Version 6 local → Version 7 Remote

  Jobs can be submitted that references any type of SAS Data View.

- Version 7 local → Version 6 Remote

  Jobs can be submitted that references any type of SAS view, provided the data referenced within the view contains no Version 7 specific features.

SAS Catalogs

A *SAS Catalog* is a collection of heterogeneous entries, distinguished by the type of the entry and its name. Various SAS procedures or software products are responsible for creating and managing entry types they need.

It is important to note that the engine creating a SAS catalog determines its format (the actual file format). Furthermore, this format is different in Versions 6 and 7 of the SAS System.

On the other hand, the format of a SAS catalog entry is determined by the SAS program or application that created it and may or may not be forwards or backwards compatible. See the attached table (figure 1) for a list of catalog entries that are backwards compatible in Version 7 of the SAS System.

Traditional Single User SAS

The catalog file format changed from Version 6 to Version 7. As a result, Version 6 is not capable of reading Version 7 catalogs, in any form. If there is no need to ever update Version 6 SAS catalogs, only read from them, then no form of conversion is necessary; Version 7 fully supports reading all Version 6 catalogs.[§] However, if the ability to create new entries is needed or if the desire to update an exiting entry in a Version 6 catalog is present, PROC COPY or PROC CPORT / PROC CIMPORT can be used to convert a Version 6 catalog to one of Version 7 format. It is important to note that the nature of the resulting catalogs is different.

PROC COPY can take a Version 6 SAS Catalog and produce a Version 7 SAS Catalog. However, the resulting format of the catalog entries that are created are in **Version 6** format due to the fact that the application or SAS program that originally created them was a SAS Version 6 application or program. As entries are updated, they will be changed to Version 7 format but those never updated are never changed. All new entries added to a PROC COPY converted catalog in Version 7 will of course be in the Version 7 format.

PROC CPORT / PROC CIMPORT take as input an existing Version 6 SAS catalog and produce as output a Version 7 catalog. In this case, the resulting entries contained in the new catalog are in Version 7 format, not the Version 6 entry format produced by PROC COPY.

Consider the following example running under Version 7 of the SAS System:

```
/* Make the library assignments for the Version */
/* 6 library, v6lib and the Version 7 library,  */
/* v7lib. */
libname v6lib ’path-to-v6-library’;
libname v7lib ’path-to-v7-library’;

/* Create a transport file for the V6 catalog, */
```

[‡] It is important to note that PROC UPLOAD and PROC DOWNLOAD are able to create a truncated name for variables that use the new Version 7 feature of long names. This truncation is based on the stem of the variable name and is guaranteed to be unique, though not as descriptive as the long name itself.

[§] This is not the case when running the SAS System on RS/6000 based hosts. Due to host specific issues, the Version 6 catalog file format is not forwards compatible with Version 7 of the SAS System and as a result, all Version 6 catalogs on this platform must be converted with PROC CPORT / PROC CIMPORT before use in Version 7.

```
/* v6lib.mycat. */
proc cport cat=v6lib.mycat file=’myxpt.xpt’;
run;

NOTE: Proc CPORT begins to transport catalog
V6LIB.MYCAT

NOTE: The catalog has 3 entries and its maximum
logical record length is 32767.

NOTE: Entry SASAFCBT.AFCBT has been transported.
NOTE: Entry AF.AFGO has been transported.
NOTE: Entry TESTCBT.CBT has been transported.

/* Import for the newly created transport file */
/* to produce v7lib.mycat. */
proc cimport cat=v7lib.mycat file=’myxpt.xpt’;
run;

NOTE: Proc CIMPORT begins to create/update catalog
V7LIB.MYCAT

NOTE: Entry SASAFCBT.AFCBT has been imported.
NOTE: Entry AF.AFGO has been imported.
NOTE: Entry TESTCBT.CBT has been imported.

NOTE: Total number of entries processed in catalog
V7LIB.MYCAT: 3
```

The catalog V6LIB.MYCAT has been fully converted to a usable Version 7 catalog, V7LIB.MYCAT. At this point, V7LIB.MYCAT can be read or updated by any Version 7 specific application and can make use of Version 7 specific features, such as renaming the entry AF.AFGO to LONGNAMEFORTHISENTRYAF.AFGO.

The following table shows what modes of catalog access are allowed in the traditional single user SAS case:

	Version 6 Catalog	**Version 7 Catalog**
Version 6 SAS Session	Read/Write/Update	■
Version 7 SAS Session	Read/Write[†]	Read/Write/Update

[†] Limited writing is allowed so that PROC COPY can copy one V6 catalog to another V6 library

As seen above, Version 7 can only write a Version 6 catalog when PROC COPY is being used to copy one Version 6 catalog or library to another Version 6 catalog or library. Any other allowable form of writing could produce a Version 6 catalog containing Version 7 data. This catalog would be unreadable by any Version 6 SAS application or program and hence, the library it was contained in would be corrupt.

SAS/SHARE®

Access to catalogs through SAS/SHARE® differs somewhat from the access that is allowed in the Traditional Single User model. Version 6 is allowed some ability of creating Version 7 catalogs that will contain catalog entry data in Version 6, not Version 7, format. However, SAS/SHARE® does not allow Version 7 data to be placed into a Version 6 SAS Data Library.

The following table describes the catalog access allowed between Versions 6 and 7 clients and servers:

	Version 6 Server		**Version 7 Server**	
	V6 Catalog	V7 Catalog	V6 Catalog	V7 Catalog
V6 Client	R/W/U	■	R	R/W/U[†]
V7 Client	R/W[‡]	■	R/W[‡]	R/W/U

[†] The catalog produced will be a Version 7 catalog with Version 6 data
[‡] Limited writing is allowed so that PROC COPY can copy a V6 catalog to another V6 library

When a Version 6 client attempts to manipulate catalogs on a Version 7 server, the results depend on the version of the catalog. If the catalog is Version 6, the only operation the client is allowed is the ability to read its entries. If the Version 6 client was allowed to write or update Version 6 catalogs on the Version 7 server, the resulting catalog would be of Version 6 format with Version 7 entry data. After this occurrence, no Version 6 SAS program or application would be able to use the new catalog; it would be corrupted.

When a Version 7 client accesses a Version 6 catalog on either a Version 6 or Version 7 server, only a limited form of writing is allowed. Again, PROC COPY can be used to copy the Version 6 catalog from one Version 6 library to another. Any other form of writing could produce the aforementioned corruption.

SAS/CONNECT®

Remote Data Transfer Services

The following table summarizes the restrictions on catalog access through SAS/CONNECT® with Remote Data Transfer Services:

	Version 6 Remote		**Version 7 Remote**	
	V6 Catalog	V7 Catalog	V6 Catalog	V7 Catalog
V6 Local	R/W/U	■	R[†]	R/W/U[‡]
V7 Local	R/W[‡]	■	R/W[†]	R/W/U

[†] PROC UPLOAD cannot create a Version 6 catalog entry
[‡] Relies on the entry’s backwards compatibility code

PROC UPLOAD running under Version 7 is not allowed to create a Version 6 catalog entry. If it were allowed to do so, the resulting Version 6 catalog would contain Version 7 data, and as seen with SAS/SHARE® and the standalone version of SAS as well, this would lead to a corrupt catalog that unusable by other Version 6 applications.

As previously mentioned, there are several catalog entries that are backwards compatible to Version 6 (see the attached table (figure 1) for a complete list of these entries) but the majority of Version 7 catalog entries are not backwards compatible with Version 6. For example, the **AMETHOD** entry type can go back to Version 6 but **SCL** and **FRAME** cannot. As shown above, a Version 7 catalog entry can contain Version 6 data and hence, it would be readable by Version 7 applications.

Remote Compute Services
Access to SAS catalogs is restricted as follows:

- Version 6 local → Version 7 Remote

  Jobs are allowed to be submitted that reference SAS catalogs.

- Version 7 local → Version 6 Remote

  Jobs are allowed to be submitted that reference SAS catalogs, provided that the jobs contain no Version 7 specific features the Version 6 remote host would be unable to process.

Conclusions

The main goals of Version 6 to Version 7 compatibility are to allow the most transparent access possible to Version 6 data and to prevent Version 7 data from being written to Version 6 data libraries. There are restrictions placed upon this access and the type of restrictions depend heavily on the environment being used: Traditional Single User SAS, SAS/SHARE®, and SAS/CONNECT®. As you migrate to Version 7, solutions vary, depending on your environment. This paper was intended to serve as a reference to that end and it is my hope that you found it worthwhile.

Acknowledgments

Many people were involved in the development of the features described in this paper. They include:

Anne Albright	Testing and verification.
Steve Beatrous	SAS I/O Design and verification.
Lisa Brown	Library Concatenation.
Giselle Cannon	Testing and verification.
Greg Dunbar	Testing and verification.
Cheryl Garner	SAS/CONNECT®.
James Holman	SAS/SHARE® Catalog Services.
Beth Langston	Testing and verification.
Diane Olson	Integrity Constraints and PROC COPY.
Heesun Park	Testing and verification.
Rebecca Perry	SAS Catalogs.
Joe Slater	SAS/SHARE® Data Files and Views.
Stephanie Tysinger	Testing and verification.

Author

James Holman
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513
(919) 677-8000
jaholm@wnt.sas.com (James Holman)

Figure 1 - SAS Catalog Entry Types Backwards Compatible to Version 6

AFCBT
AFGO
AFPGM
AGENTDEF
AGENTRUN
AGENTSCH
AMETHOD
CATAMS
CMPSRC
CMPSUB
COMAMOPT
CPARMS
DEVMAP
DISTNARY
DM_CLASS
DM_INFO
DM_LEVEL
DM_R2
DM_SPLIT
DOMAINA
DOMAINC
DOMAINE
DOMAINX
ENGINE
FMSFORM
FMSPROJ
FONT
FONTIMAG
FONTLIST
FUNCL
GEDIT
GLOBAL
GOPTIONS
GRIMAGE
HSERVICE
IMAGE
IMOD
INVEST
KEYMAP
KEYS
LOG
MACLIB
MATRIX
MODEL
MODOUT
MODPARMS
MODPROJ
MODSPEC
MODSRC
OUTPUT
RISK1
RISK2
RISK3
RISK4
RISK5
RISK6
RISK7
RISK8
RISK9
RISK10
RISK11
RISK12
RISK13
RISK14
RISK16
RISK17
RISK18
RISK19
RISK20
RISK21
RISK22
RISK23
RISK24
RISK25
RISK26
RISK27
RISK28
RISK29
RISK31
RISK32
RISKOBJ
SOURCE
STATGRAF
TEMPLATE
TITLE
TOOLBOX
TOOLSET
TRANTAB
UNXLIBS
UNXPREFS
VMSLIBS
VMSPREFS
WSAVE
XPRINTER