



# CHAPTER 34

## Compatibility of Version 8 with Earlier Releases

---

<i>Definitions</i>	493
<i>Overview of Version Compatibility</i>	494
<i>SAS Library Engines</i>	495
<i>Accessing SAS Data Libraries</i>	497
<i>Concatenating Version 8 Libraries with Libraries from Earlier Releases</i>	497
<i>Combining Version 8 Files with Files from Earlier Releases</i>	497
<i>Accessing SAS Data Files</i>	498
<i>Using Version 8 to Access Data Files from Earlier Releases without Converting</i>	498
<i>Using Version 6 to Access Version 8 Data Files</i>	498
<i>Converting Version 6 Data Files to Version 8 Format</i>	498
<i>Creating Version 6 Data Files in Version 8</i>	499
<i>Accessing SAS Views</i>	500
<i>Using Version 8 to Access Views from Earlier Releases without Converting</i>	500
<i>Using Version 6 to Access Version 8 Views</i>	500
<i>Converting Version 6 Views to Version 8 Format</i>	500
<i>Creating Views from Earlier Releases in Version 8</i>	500
<i>Accessing SAS Catalogs</i>	501
<i>Using Version 8 to Access Version 6 Catalogs without Converting</i>	501
<i>Using Version 6 to Access Version 8 Catalogs</i>	501
<i>Converting Version 6 Catalogs to Version 8 Format</i>	501
<i>Creating Version 6 SAS Catalogs in Version 8</i>	502
<i>Accessing Stored Compiled DATA Step Programs</i>	502
<i>Accessing MDDB Files</i>	502

---

### Definitions

*convert a SAS file*

changes the format of a SAS file from one version to another, for example, from Version 6 to Version 8 format.

*engine*

is a part of the SAS System that reads from or writes to a SAS file in a data library. Each engine allows SAS to access files with a particular format. Having multiple engines enables SAS to access different formats and versions of SAS files.

*libref*

is a shortcut name associated with a SAS data library.

*mixed-mode directory*

is a directory that contains SAS files from more than one release, for example, Version 6 and Version 8.

*SAS catalog*

is a SAS file that stores different kinds of information in separate units called catalog entries, which are distinguished by the entry type and name. A SAS catalog has the member type CATALOG. Various SAS procedures and products create and manage entry types.

*SAS data file*

is a SAS data set that contains both the data values and the descriptor information. A data file has the member type DATA.

*SAS data library*

is a collection of one or more SAS files that are recognized by SAS. Each file is a member of the library.

*SAS data set*

is a SAS file that consists of descriptor information and data values organized as a table of observations (rows) and variables (columns) that can be processed by SAS. A SAS data set can be either a SAS data file or a SAS view.

*SAS file*

is a specially structured file that is created, organized, and maintained by SAS. SAS files reside in SAS data libraries as members with specific types. Examples of SAS files are a SAS data set (which can be a SAS data file or a SAS view), a SAS catalog, a stored compiled DATA step program, an access descriptor file, and database files such as MDDB, FDB, and DMDB files.

*SAS view*

is a SAS data set that contains only the information required to retrieve values. The data is obtained from another file. A SAS view has the member type VIEW. There are three types of SAS views:

- DATA step view
- SAS/ACCESS view
- PROC SQL view.

*stored compiled DATA step program*

is a SAS file that contains a DATA step program that has been compiled and stored in a SAS data library. A stored compiled DATA step program has the member type PROGRAM.

---

## Overview of Version Compatibility

When you migrate to Version 8, you'll want to seamlessly access your existing data and programs. You may also need to operate with both Version 8 and an earlier release simultaneously. Therefore, a major goal of Version 8 is to provide the most transparent access possible to SAS files from earlier releases. Accordingly, the Version 8 SAS System provides access to all Version 7 files and most Version 6 files without converting them.

Accessing a SAS data library and its members is essentially the same in Version 8 as it is in earlier releases. Depending on the type of SAS file and the SAS version being used, compatibility issues are generally handled

- automatically by the SAS System
- by specifying an engine in a LIBNAME statement or with the ENGINE= system option
- by converting a file.

*Note:* This information explains version compatibility for SAS files in base SAS software for a single operating environment. For related documentation, consult the following SAS documents:

- For platform-dependent compatibility issues, see the SAS documentation for your operating environment.
- The *SAS/SHARE User's Guide* and the *SAS/CONNECT User's Guide* contain specific information for those products regarding file compatibility.
- For information on moving SAS files between operating environments, see *Moving and Accessing SAS Files across Operating Environments*.

△

---

## SAS Library Engines

To access a SAS data library, SAS needs a libref and a library engine. You assign a libref to the SAS data library, for example, using the LIBNAME statement, but usually you do not have to specify an engine because SAS automatically selects the appropriate one. For base SAS software, Version 8 provides the following library engines.

*Note:* Engine availability is platform dependent. See the SAS documentation for your operating environment. Also, specific SAS products provide additional engines. △

**Table 34.1** Version 8 Base SAS Software Library Engines

Type Of Engine	Engine Name	Description
Default Version 8 engine	V8 (or V7, V701, BASE)	Creates and accesses Version 8 SAS files and accesses Version 6 and Version 7 SAS files.
Version 8 sequential engine	V8TAPE (or TAPE, V7TAPE)	Creates and accesses Version 8 SAS files and accesses Version 6 and Version 7 SAS files on storage media that do not allow random access methods, for example, tape or sequential format on disk.
Version 6 compatibility engine	V6 (or V608, V609, V610, V611, V612)	Creates and accesses SAS files created by Release 6.08 through Release 6.12 without converting to Version 8 format.  <i>Operating Environment Information:</i> For Version 6 files prior to Release 6.08, see the SAS documentation for your operating environment. △
Version 6 sequential engine	V6TAPE	Creates and accesses Version 6 SAS files on storage media that does not allow random access methods, for example, tape or sequential format on disk.

Type Of Engine	Engine Name	Description
Transport	XPORT	Accesses transport files. This engine creates machine-independent SAS transport files that can be used for all hosts.
Version 5 compatibility engine	V5	Accesses Version 5 SAS files without converting to Version 8 format. On OS/390 and CMS, the V5 engine is read only. On VMS, the V5 engine is both read and write. This engine cannot access Version 6 or later files.

If you do not specify an engine name, SAS automatically assigns an engine based on the contents of the data library. That is, SAS is able to differentiate between Version 6 libraries and those in later releases, because the engine that creates a SAS file determines its format and the format is different between Version 6 and later versions.

For example, in a Version 8 SAS session, if you issue the following LIBNAME statement to assign a libref to a data library containing only Version 6 SAS files, SAS automatically assigns the V6 compatibility engine:

```
libname mylib "SAS-data-library";
```

SAS automatically assigns an engine based on the contents of the data library as shown in the following table:

**Table 34.2** Default Library Engine Assignment

Engine Name Assignment	Data Library Contents
V8	No SAS files; the library is empty
V8	Only Version 8 SAS files*
V6	Only Version 6 SAS files*
V8	Both Version 8 SAS files and SAS files from earlier releases
V8TAPE	Both Version 8 TAPE files and TAPE files from earlier releases

\* If a library contains only files that were created from a single engine, that engine is the default. Note that Version 8 and Version 7 files are created from the same engine.

*Note:* Even though SAS will automatically assign an engine based on the library contents, it is more efficient to specify the engine name in a LIBNAME statement. For example, specifying the engine name in the following LIBNAME statement saves SAS from determining which engine to use:

```
libname mylib v6 "SAS-data-library";
```

$\Delta$

---

## Accessing SAS Data Libraries

---

### Concatenating Version 8 Libraries with Libraries from Earlier Releases

A technique that can help you migrate to Version 8 is to reference multiple SAS libraries with a single libref, referred to as *library concatenation*. For example, by concatenating Version 6 and Version 8 data libraries, you can migrate your files one at a time. That is, you can convert some files to Version 8 format (for example, using the COPY procedure), while other files remain in Version 6 format.

For example, suppose you have files in both a Version 6 library and a Version 8 library for which an application needs to process. The following LIBNAME statements allow you to access both Version 6 and Version 8 libraries using one libref. Note that the engine names are specified in the first two LIBNAME statements for clarity; specifying the engine name is optional.

- 1 You assign a libref to the Version 6 library to use the V6 compatibility engine.

```
libname old v6 "v6-SAS-data-library";
```

- 2 You assign a libref to the Version 8 library to use the Version 8 default engine.

```
libname new v8 "v8-SAS-data-library";
```

- 3 You concatenate the two into one libref.

```
libname mylib (new old);
```

Now you can invoke the application using the MYLIB libref, which accesses both data libraries.

For more information on library concatenation, see “Library Concatenation” on page 390.

---

### Combining Version 8 Files with Files from Earlier Releases

In some operating environments, you can combine Version 6 and Version 8 files in one directory, which is referred to as a *mixed-mode directory*. To access the files, you assign different librefs with different engines to the single directory. For example, the following statements assign two librefs to the same directory: one for the V6 compatibility engine and the other for the V8 engine:

```
libname v6files v6 "path-to-SAS-data-library";
libname v8files v8 "path-to-SAS-data-library";
```

To access the files, you reference the appropriate libref. For example, to print a Version 6 data set, you would issue:

```
proc print data=v6files.member1;
run;
```

To print a Version 8 data set, you would issue:

```
proc print data=v8files.member2;
run;
```

*Note:* If you combine Version 7 and Version 8 files in the same directory, note that the file extensions (and the file formats) are the same in both releases. Therefore, a

Version 7 file will be overwritten by a Version 8 file of the same name stored in the same directory.  $\triangle$

---

## Accessing SAS Data Files

---

### Using Version 8 to Access Data Files from Earlier Releases without Converting

A Version 8 SAS session can read and update a Version 6 data file or a Version 7 data file without converting the file as long as the features included in the data file are supported by the file format's version. That is, you cannot use Version 8 features for a Version 6 data file.

In general, you can use Version 8 to manipulate Version 6 data files, using the V6 compatibility engine as long as needed. However, you will not be able to maximize the potential of Version 8 until you convert Version 6 data files to Version 8 format.

---

### Using Version 6 to Access Version 8 Data Files

Version 6 cannot access a Version 8 data file due to differences in the file format, except with SAS/SHARE or SAS/CONNECT software.

---

### Converting Version 6 Data Files to Version 8 Format

Even though you can use Version 8 with Version 6 data files, in order to use Version 8 features such as long variable names, integrity constraints, and generation data sets, you must convert existing data to Version 8 format.

To convert a Version 6 data file to Version 8 format, you can use one of the following methods:

- use the V6 compatibility engine and the COPY procedure. In the following example, the first LIBNAME statement specifies the V6 compatibility engine and the libref OLD, which points to the library containing the Version 6 data files. The second LIBNAME statement specifies the V8 engine and the libref NEW, which points to the new library to which the data will be copied. PROC COPY reads the data files identified by the IN= option with the V6 engine and writes them to the new library identified in the OUT= option with the V8 engine. Note that the engine names are specified in the LIBNAME statements for clarity; specifying the engine name is optional.

```
libname old v6 "v6-SAS-data-library";
libname new v8 "v8-SAS-data-library";

proc copy in=old out=new memtype=data;
run;
```

- use the V6 compatibility engine and a DATA step. This technique works well if you want to convert only one or two data files:

```
libname old v6 "v6-SAS-data-library";
libname new v8 "v8-SAS-data-library";

data new.data;
```

```

    set old.data;
run;

```

- use the CPORT and CIMPORT transport procedures. The following program uses PROC CPORT to create a transport file from the Release 6.12 data file V6LIB.MYDATA:

```

/* Release 6.12 SAS program */

libname old "v6-SAS-data-library";

proc cport cat=old.mydat file='myxpt.xpt';
run;

```

The next program then uses PROC CIMPORT to convert the transport file to the Version 8 data file NEW.MYDAT:

```

/* Version 8 SAS program */

libname new "v8-SAS-data-library";

proc cimport cat=new.mydat file='myxpt.xpt';
run;

```

*Note:* Depending on your operating environment, PROC CPORT and PROC CIMPORT may require different syntax. See the SAS documentation for your operating environment.  $\Delta$

---

## Creating Version 6 Data Files in Version 8

You may need to create a Version 6 data file in a Version 8 session, for example, if you are sharing data with a Version 6 application. To do so, you use the V6 compatibility engine. For example, the following statements use the V6 engine to create a SAS data file named QTR1. The raw data is read from the external file associated with the fileref MYFILE:

```

libname newdata v6 "SAS-data-library";

filename myfile "external-file";

data newdata.qtr1;
    infile myfile;
    input saledata amount;
run;

```

You may also need to create a Version 6 data file from a Version 8 data file. However, because the Version 8 file could contain features like long variable names that are not compatible with Version 6, you would need to remove Version 8 features. In Version 8, the COPY procedure can automatically truncate long variable names if you specify the VALIDVARNAME=V6 system option.

For example, assume that a Version 8 SAS data file named V8LIB.EMPLOYEE contains the variables LASTNAME, FIRSTNAME, and EMPLOYEEID. By issuing the following PROC COPY with the VALIDVARNAME=V6 system option, the resulting Version 6 SAS data file V6LIB.EMPLOYEE contains the variables LASTNAME, FIRSTNAM, and EMPLOYEE:

```

libname v8lib "v8-SAS-data-library";

```

```

libname v6lib "v6-SAS-data-library";

options validvarname=v6;

proc copy in=v8lib out=v6lib;
    select=employee;
run;

```

---

## Accessing SAS Views

---

### Using Version 8 to Access Views from Earlier Releases without Converting

Version 8 can read all types of Version 6 and Version 7 SAS views. That is, Version 8 can read Version 6 and Version 7 DATA step views, SAS/ACCESS views, and PROC SQL views.

In addition, Version 8 can use Version 6 and Version 7 SAS/ACCESS and PROC SQL views to update data.

---

### Using Version 6 to Access Version 8 Views

Version 6 cannot access SAS views from later releases because of differences in the file format, except with SAS/SHARE or SAS/CONNECT software.

---

### Converting Version 6 Views to Version 8 Format

Converting Version 6 SAS views to Version 8 format depends on the following:

- DATA step views can be converted if the data files or views that the DATA step view accesses are available.
- PROC SQL views can be converted if they are views of SAS data files; PROC SQL views cannot be converted if they are views to other SAS views.
- SAS/ACCESS views can be converted if the database product is available.

To convert a Version 6 view to Version 8 format, you can use the COPY procedure.

In the following example, the first LIBNAME statement specifies the V6 compatibility engine and the libref OLD, which points to the library containing the Version 6 views. The second LIBNAME statement specifies the V8 engine and the libref NEW, which points to a Version 8 library to which the views will be copied. PROC COPY reads the data files that is identified by the IN= option with the V6 engine, and then writes them to the new library that is identified in the OUT= option with the V8 engine.

```

libname old v6 "v6-SAS-data-library";
libname new v8 "v8-SAS-data-library";

proc copy in=old out=new memtype=view;
run;

```

---

### Creating Views from Earlier Releases in Version 8

In Version 8, the ability to create SAS views for earlier releases depends on the type of view:



- Version 8 cannot create Version 6 or Version 7 DATA step views.
- Version 8 can create Version 6 or Version 7 SAS/ACCESS views if you use the V6 compatibility engine.
- Version 8 cannot create Version 6 PROC SQL views.

---

## Accessing SAS Catalogs

*Note:* The engine that creates a SAS catalog determines its format, which is different in Version 6 and Version 8 and therefore not compatible. However, the format of a SAS catalog entry is determined by the SAS program or application that creates it and may or may not be compatible between versions.  $\Delta$

---

### Using Version 8 to Access Version 6 Catalogs without Converting

Version 8 can read a Version 6 catalog. Therefore, if you do not need to update a Version 6 catalog, then you do not need to convert it.

In general, Version 8 cannot write to a Version 6 catalog. However, you can use the COPY procedure to write a Version 6 catalog from a Version 6 library to another Version 6 library.

Version 8 cannot create new entries or update existing entries in a Version 6 catalog. You must convert the catalog to Version 8 format.

---

### Using Version 6 to Access Version 8 Catalogs

Version 6 cannot access Version 8 catalogs, because the file formats are not compatible.

---

### Converting Version 6 Catalogs to Version 8 Format

For a Version 6 catalog, to create new entries or to update existing ones, you must convert the catalog to Version 8 format. Two methods are available, which produce different results regarding catalog entries:

- You can use the COPY procedure to convert a Version 6 catalog to Version 8 format. However, the resulting catalog entries are in Version 6 format, because the application or SAS program that originally created them was a Version 6 application or program. As those entries are updated, they are changed to Version 8 format; entries never updated are not changed. New entries are, of course, in Version 8 format. For example:

```
libname old v6 "v6-SAS-data-library";
libname new v8 "v8-SAS-data-library";

proc copy in=old out=new memtype=catalog;
run;
```

- The CPORT and CIMPORT transport procedures can produce an output Version 8 catalog. Unlike PROC COPY, the resulting catalog entries are in Version 8 format.

For example, the following program uses PROC CPORT to place the contents of the Release 6.12 catalog OLD.MYCAT in a transport file.

```
/* Release 6.12 SAS program */  
  
libname old "v6-SAS-data-library";  
  
proc cport cat=old.mycat file='myxpt.xpt';  
run;
```

Then, the following program uses PROC CIMPORT to convert the transport file to the Version 8 catalog NEW.MYCAT:

```
/* Version 8 SAS program */  
  
libname new "v8-SAS-data-library";  
  
proc cimport cat=new.mycat file='myxpt.xpt';  
run;
```

---

## Creating Version 6 SAS Catalogs in Version 8

You cannot create a Version 6 SAS catalog in Version 8, except with SAS/SHARE and SAS/CONNECT software.

---

## Accessing Stored Compiled DATA Step Programs

Version 8 can access Version 6 and Version 7 stored compiled DATA step programs. However, Version 6 cannot access any Version 8 stored compiled DATA step program.

---

## Accessing MDDB Files

Version 8 can access Version 6 and Version 7 MDDB files. You can also use PROC COPY to convert an MDDB from Version 6 to Version 8 format.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS Language Reference: Concepts*, Cary, NC: SAS Institute Inc., 1999. 554 pages.

**SAS Language Reference: Concepts**

Copyright © 1999 SAS Institute Inc., Cary, NC, USA.

ISBN 1-58025-441-1

All rights reserved. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, by any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute, Inc.

**U.S. Government Restricted Rights Notice.** Use, duplication, or disclosure of the software by the government is subject to restrictions as set forth in FAR 52.227-19 Commercial Computer Software-Restricted Rights (June 1987).

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

1st printing, November 1999

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries.® indicates USA registration.

IBM, ACF/VTAM, AIX, APPN, MVS/ESA, OS/2, OS/390, VM/ESA, and VTAM are registered trademarks or trademarks of International Business Machines Corporation. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

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