

## CHAPTER

## 6

## Using SAS/CONNECT to Transport Files between Hosts

<i>Supported Members and Files</i>	45
<i>Uploading and Downloading SAS Files</i>	45
<i>Connecting Local and Remote Sessions</i>	46
<i>Managing Incompatible Local and Remote Host Types</i>	46
<i>Transporting SAS Files</i>	47
<i>Transporting Individual Data Sets</i>	47
<i>Transporting Members in a Data Library</i>	48
<i>Transporting Catalog Entries</i>	48
<i>Transporting SAS Files between Versions</i>	49
<i>Cross-Version Compatibility</i>	49
<i>Truncating Version 8 Data Set Names for Version 6 Access</i>	49
<i>Version 6 Local Host Accessing a Version 8 Remote Host</i>	50
<i>SAS Data Files</i>	50
<i>SAS SQL Views</i>	50
<i>SAS Catalogs</i>	50
<i>Version 8 Local Host Accessing a Version 6 Remote Host</i>	51
<i>SAS Data Sets</i>	51
<i>SAS SQL Views</i>	51
<i>SAS Catalogs</i>	51
<i>Version 6-Version 8 Compatible Catalog Entry Types</i>	51

### Supported Members and Files

You can use SAS/CONNECT to move any of these SAS members and files:

- Data set
- Catalog or entry
- PROC SQL view
- MDDB (multi-dimensional database) file
- DMDB (data mining database) file
- FDB (financial database) file.

### Uploading and Downloading SAS Files

You can use the data transfer services provided by SAS/CONNECT to move a copy of data between hosts. You can position data and applications to best use the CPU and disk resources on your network. SAS/CONNECT provides two procedures to support data transfer services:

**PROC UPLOAD**

sends one or more files from a *local* host to a *remote* host.

**PROC DOWNLOAD**

sends one or more files from a *remote* host to a *local* host.

PROC UPLOAD and PROC DOWNLOAD enable you to transfer an entire library, selected members of a library (by using SELECT or EXCLUDE statements), or selected data sets with WHERE processing and SAS data set options. Furthermore, you can move SAS files across SAS releases as well as across hosts; for example, from OS/390 Release 6.09 to Windows Version 8. For complete details about SAS/CONNECT, see *SAS/CONNECT User's Guide*.

---

## Connecting Local and Remote Sessions

A local host sign on to a remote host establishes a connection between your local SAS session and a remote SAS session. You invoke SAS/CONNECT on a remote host by means of a TELNET daemon or a spawner program. The spawner program runs on the remote host and listens for client requests for connection to the remote host.

Here is a typical sign on:

```
options comamid=access-method remote=host-id
      cscript=script-file;
signon;
```

where:

**COMAMID=**

specifies the communication access method identifier that is used for passing data between the two hosts.

**REMOTE=**

designates the name of the remote host that you want to access.

**CSCRIPT=**

optionally specifies the script file to be used for sign on. The need for a sign-on script is dependent on the access method that is used to communicate between hosts and the method that is used to invoke SAS/CONNECT. The same script that is used to sign on is usually also used at sign off.

*Note:* Omit the CSCRIPT= option when using a spawner program to invoke SAS/CONNECT.  $\Delta$

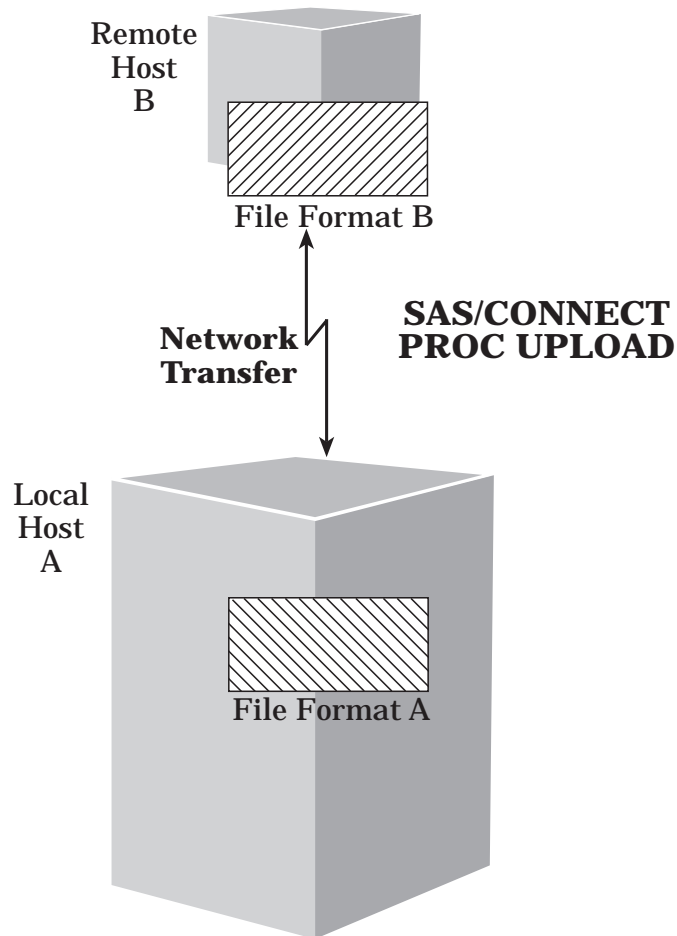
---

## Managing Incompatible Local and Remote Host Types

SAS/CONNECT dynamically translates incompatible file formats between the local host and the remote host for each file upload or file download transaction, bypassing the explicit creation of a transport file.

Figure 6.1 on page 47 shows the transfer of data between a local host and a remote host that are incompatible with each other. PROC UPLOAD translates the file from the local host file format to the remote host file format.

**Figure 6.1** Using SAS/CONNECT PROC UPLOAD to Transport a SAS File for Remote Host Processing




---

## Transporting SAS Files

---

### Transporting Individual Data Sets

You use `DATA=` and `OUT=` options in `PROC UPLOAD` and `PROC DOWNLOAD` to transport data sets between hosts. The `DATA=` option identifies the source data set, and the `OUT=` option identifies the target name and destination.

This example shows how to copy the data set `GRADES` from the library `SCHOOL` on the local host to the temporary directory `GRADES_COPY` on the remote host.

```
proc upload data=school.grades out=grades_copy;
run;
```

You can use the `WHERE` statement to further subset the data based on the characteristics of the observations within the data set. The resulting data set that is transferred contains only those observations that satisfy the `WHERE` expression.

---

## Transporting Members in a Data Library

You use the INLIB= and OUTLIB= options to PROC UPLOAD or PROC DOWNLOAD to transfer an entire library or selected members of a library. You use the MEMTYPE= option for specifying which member types to transfer. Valid values for the MEMTYPE= option are:

- DATA (data sets)
- VIEW (PROC SQL views)
- CATALOG (catalog files)
- MDDB (multi-dimensional database files)
- DMDB (data mining database files)
- FDB (financial database files)
- ALL (all member types in the library).

This example transports all data sets and catalogs in the library REMLIB on the remote host and stores them in the library LOCLIB on the local host:

```
proc download inlib=remlib outlib=loclib
  memtype=(data catalog);
run;
```

You can also use the SELECT or EXCLUDE statement to specify certain members within a library for transfer. Because the statements are mutually exclusive, they cannot be used together in the same procedure.

---

## Transporting Catalog Entries

You use the INCAT= and OUTCAT= options in PROC UPLOAD or PROC DOWNLOAD to transfer a catalog. You can also use the SELECT or EXCLUDE statement to specify certain entries within the catalog or the ET= option to select a specific entry type within the catalog for transfer.

This example transfers all the SCL catalog entries from the catalog REMLIB.GENERAL-TOOLS on the remote host to the catalog LOCLIB.MY-TOOLS on the local host:

```
proc download incat=remlib.general-tools
  outcat=loclib.my-tools
  et=scl;
run;
```

You can use SELECT statements to identify specific catalog entries for transfer. This example builds upon the preceding example.

```
proc download incat=remlib.general-tools
  outcat=loclib.my-tools
  select grades.format scores.infmt;
  select midterm final / et=scl;
run;
```

The SELECT statements transferred these catalog entries:  
GRADES of entry type FORMAT

SCORES of entry type INFMT  
 MIDTERM of entry type SCL  
 FINAL of entry type SCL.

---

## Transporting SAS Files between Versions

The data transfer services in SAS/CONNECT also moves data between versions of SAS software as well as across hosts. Many organizations may find it necessary to operate in a mixed-version environment while making a transition from Version 6 to Version 8. This environment includes a mixture of local or remote hosts running either a Version 6 session or a Version 8 session.

Whenever possible, it is generally recommended that you keep Version 6 and Version 8 SAS files in separate physical locations. This will avoid any potential confusion as to which file is being accessed. If mixed versions are stored in the same library, the appropriate engine should be specified when accessing the SAS data files. For more details about operating in a mixed library environment, see Chapter 7, “Using SAS/SHARE to Access Remote Files,” on page 55.

---

### Cross-Version Compatibility

Version 8 offers several new features that make Version 8 data files incompatible with Version 6 data files. However, for new features that involve increased character length, SAS/CONNECT truncates the character length in an attempt to produce a Version 6 compatible data file. This allows you to take advantage of some of the new Version 8 features while operating in a mixed-version environment. The Version 8 features that SAS/CONNECT is able to truncate are:

- long data set labels
- long variable labels
- long variable names.

Other new Version 8 features cannot be modified to make data files compatible with Version 6. Therefore, any attempt to transfer data files that contain these features will fail. Some of the new features that cannot be modified to be Version 6 compatible include:

- data set generations
- integrity constraints.

### Truncating Version 8 Data Set Names for Version 6 Access

Version 8 supports longer names and labels than those supported in Version 6. The longer names and labels are stored in Version 8 data files, which makes them incompatible with Version 6. SAS/CONNECT implements a set of truncation rules to convert a Version 8 data file that contains long names and labels back into a Version 6 data file.

SAS/CONNECT applies the truncation rules during file transfer

- from a Version 8 session to a Version 6 session.
- between two Version 8 sessions (including a Version 7 session and a Version 8 session) to produce a Version 6 data file.

*Note:* In order to produce a Version 6 data file, you set the SAS option and value VALIDVARNAME=V6 in the session that creates the data file. With VALIDVARNAME=V6 set, the truncation is performed when running the V8 engine or the V6 engine.  $\Delta$

In either case, the following rules are applied for data sets that have long data set labels, long variable labels, and/or long variable names. In each case, the length is truncated to the length that is supported by Version 6.

- Data set labels are truncated to 40 characters.
- Variable labels are truncated to 40 characters.
- Variable names are truncated to 8 characters.

*Note:* If the variable label field is blank, the long variable name is copied to the label field.  $\Delta$

The truncation algorithm that is used to produce the 8-character variable name also resolves conflicting names.

- 1 The first name that is greater than 8 characters is truncated to 8 characters. For example, STOCKNUMBER53 is truncated to STOCKNUM.
- 2 The next name that is greater than 8 characters is truncated to 8 characters. If it conflicts with an existing variable name, it is truncated to 7 characters, and a suffix of 2 is added. For example, STOCKNUMBER54 is truncated to STOCKNU2.
- 3 The suffix is increased by 1 for each truncated name that conflicts with an existing name. If the suffix reaches 9, the next conflicting variable name is truncated to 6 characters, and a suffix of 10 is appended. For example, STOCKNUMBER63 is truncated to STOCKN10.

---

## Version 6 Local Host Accessing a Version 8 Remote Host

This section summarizes the limitations of a Version 6 local host that accesses files on a Version 8 remote host in a cross-version environment.

### SAS Data Files

All data format conversions are performed when transferring a data file and do not require any user invention. PROC UPLOAD produces a Version 8 data file when transferring data files from a Version 6 local host. PROC DOWNLOAD provides full support for the transfer of Version 6 data files residing on the remote host. However, Version 8 data files can be downloaded to a Version 6 local host and will follow the truncation rules provided that the data files do not contain any other Version 8 specific features.

### SAS SQL Views

In order to upload or download PROC SQL views, a Version 6 local host can use the DATA= option in the procedure to specify the data file that is associated with the view to be transferred. Alternatively, the Version 6 local host can use the INLIB= option in the procedure to specify the library for transfer.

### SAS Catalogs

A Version 6 local host can upload a Version 6 catalog, but the resulting catalog is in Version 8 format. A Version 6 host can download Version 6 or Version 8 catalogs as long as the entry type is backwards compatible.

---

## Version 8 Local Host Accessing a Version 6 Remote Host

This section summarizes the limitations of a Version 8 local host that accesses files on a Version 6 remote host in a cross-version environment.

### SAS Data Sets

A Version 8 data set can be uploaded to a Version 6 remote host and will follow the truncation rules describe earlier, provided that the data sets do not contain any other Version 8 specific features. PROC DOWNLOAD transfers Version 6 data sets to the Version 8 local host.

### SAS SQL Views

A Version 8 local host can upload data that is associated with a view to a Version 6 remote host and will follow the truncation rules.

### SAS Catalogs

A Version 8 local host can download a Version 6 catalog from a Version 6 remote host. A Version 6 catalog can also be uploaded from the Version 8 local host if the entry type is backward compatible.

*Note:* The Version 8 local host cannot use PROC UPLOAD to create a Version 6 catalog entry because Version 6 catalogs are read-only in Version 8. △

### Version 6-Version 8 Compatible Catalog Entry Types

*Note:* Version 8 cannot read Version 6 catalogs on an AIX UNIX host. In order to move Version 6 catalogs to Version 8 format on AIX hosts, you must use PROC CPORT and PROC CIMPORT. △

SAS/CONNECT supports moving some catalog entries between Version 8 and Version 6. Version 8 entry types that are backward compatible with Version 6 are:

DEVMAP	FONT	FONTLIST
KEYMAP	KEYS	LOG
OUTPUT	SOURCE	TEMPLATE
TRANTAB		





The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *Moving and Accessing SAS Files across Operating Environments, Version 8*, Cary, NC: SAS Institute Inc., 1999. 186 pages.

**Moving and Accessing SAS Files across Operating Environments, Version 8**

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

ISBN 1-58025-480-2

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, October 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®, AIX®, DB2®, OS/2®, OS/390®, and System/390® are registered trademarks or trademarks of International Business Machines Corporation. ORACLE® is a registered trademark or trademark of Oracle 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.