

CHAPTER

8

Using Version 8 Cross-Environment Data Access (CEDA)

<i>SAS/CONNECT Licensing Requirement</i>	65
<i>Version 8 Hosts That Support CEDA</i>	65
<i>Using CEDA to Create and Read Version 8 Supported Member Types</i>	65
<i>Supported Member Types</i>	66
<i>How CEDA Works</i>	66
<i>Advantages of CEDA</i>	66
<i>Limitations of CEDA</i>	67
<i>Using CEDA to Create a File</i>	67
<i>Using the OUTREP= Option in the LIBNAME Statement</i>	68
<i>Using the OUTREP= Option in the DATA Step</i>	68
<i>Host Format Values</i>	69
<i>Making Accessible a File That Was Created with CEDA</i>	69
<i>Reading a File That Was Created with CEDA</i>	69
<i>Finding Out if You Can Update a File That Was Created with CEDA</i>	70
<i>Accessing Files That Were Created with CEDA on Personal Computer FAT Drives</i>	71

SAS/CONNECT Licensing Requirement

Because CEDA (Cross-Environment Data Access) is bundled with SAS/CONNECT, you must have a license for Version 7 or Version 8 SAS/CONNECT software in order to use the features of CEDA.

Version 8 Hosts That Support CEDA

CAUTION:

CEDA is supported on all Version 8 hosts except the CMS Version 8 operating environment.

△

For a complete list of Version 8 hosts, see “Host Types Supported According to SAS Release” on page 4.

Using CEDA to Create and Read Version 8 Supported Member Types

CEDA is a Version 8 facility that allows you to create and to read supported member types in all Version 8 operating environments, except CMS.

An alternative to traditional transport methods, CEDA eliminates the steps that have been required to move a SAS file between the native internal representations of

incompatible hosts. This quick file-access feature of CEDA is one aspect of the Version 8 vision of the computing environment in which different computers share access to common files in a network system.

Supported Member Types

CEDA supports these SAS member types:

- Data set
- PROC SQL view¹
- MDDDB (Multi-dimensional Data Base)¹

¹ CEDA supports PROC SQL views and MDDDBs for read-only access.

How CEDA Works

You use CEDA to create a file in order to designate a file format to be other than the format of the source host. This process assigns a non-native (or foreign) format to the file being created. For example, from a UNIX host, you can create a SAS data set in Windows file format.

At the beginning of each SAS data set that is created by using Version 8 is a *universal header*. This header contains host-related information that is used, such as number size, number alignment, data representation (such as IEEE floating-point), and character set (such as ASCII). In an NFS (Network File System) environment, the accessing host uses the universal header of the file to determine how the data is represented.

If the file data representation and the accessing host data representation are the same (for example, the file is represented in Windows format and the accessing host is a Windows host), then the accessing host can read the file and write to the file.

If the file data representation and the accessing host representation are different, then the accessing host can dynamically translate the data file into the accessing host native format for read-only access. For example, an OpenVMS Alpha host can access an NFS-mounted file that is in Windows format. Using the universal header, the OpenVMS Alpha host can translate each observation that it reads into a format that it understands. The accessing host translates only the observation that it retrieves for reading.

Advantages of CEDA

CEDA offers these advantages:

- The user can transparently read a data set with no conscious concern or knowledge about the native format of the file.
- System performance is maximized because a read operation requires a single translation between native and non-native formats versus a translation from native format to transport format to native format.
- No interim transport files are created.
- CEDA eliminates the need to perform explicit steps in order to access the file.
- CEDA does not require a dedicated server as does SAS/SHARE or an explicit sign on to a remote host as does SAS/CONNECT.
- The internal numeric representation provided by CEDA is more precise than that provided by the XPORT engine with PROC COPY. CEDA uses a one-step

translation from the native format of the source host to the native format of the target host, whereas the XPORT engine uses a two-step transformation from a file's native format to the target host format by means of transport format.

Limitations of CEDA

CEDA does not replace the traditional file transport methods (PROC CPORT and PROC CIMPORT or the XPORT engine with PROC COPY or the DATA step), PROC UPLOAD and PROC DOWNLOAD in SAS/CONNECT, or server processing in SAS/SHARE because of these limitations:

- CEDA features are implemented for Version 8 SAS data sets, PROC SQL views, and MDDBs (for read-only access). CEDA does not support Version 8 stored programs, catalogs, DMDBs, or FDBs, nor does it support any pre-Version 7 member types.
- CEDA does not support update access.
- CEDA does not support subsetting by means of an index.
- CEDA is available for hosts that use directory-based file structures. On OS/390, CEDA is available only for SAS data sets that reside in a UNIX System Services Directory. Bound libraries that are traditionally used on the OS/390 host do not implement CEDA.

If your application can operate within these limitations, then CEDA offers the simplest strategy for file access across a network. If your needs exceed these limitations, then you still have other methods available.

Using CEDA to Create a File

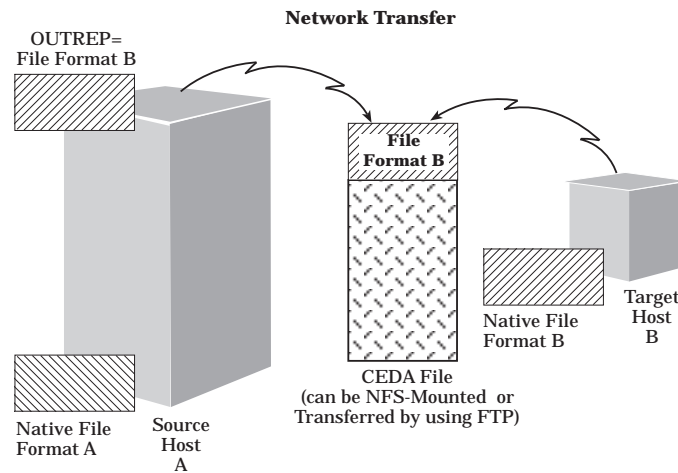
In order to create a file in a non-native (or foreign) format for a supported member type, use the OUTREP= option in the LIBNAME statement or in the DATA step.

Specified in the LIBNAME statement, the OUTREP= option applies the designated format to all files that are created in the specified library. However, specified in the DATA step, the designated host format is limited to the specified data set.

Note: The OUTREP= option in the LIBNAME statement or in the DATA statement does not support MDDB. CEDA supports MDDB files for read-only access. △

The illustration below shows how CEDA is used to create a file.

Figure 8.1 Using CEDA to Create a File in a Foreign Format for Read, Write, and Update Access



This figure shows the designation of a file in a foreign file format for access by a specific remote host. Source host A uses the OUTREP= option in either a LIBNAME or a DATA statement to create a file in foreign format B. Doing so allows remote host B to quickly and efficiently access the network-mounted file (for read, write, and update operations) because no internal format translation is required.

Using the OUTREP= Option in the LIBNAME Statement

As an example, suppose that you use a UNIX host to create data sets that contain academic grades. However, the data entry and processing personnel who read and write these data sets use Windows hosts. In order to accommodate the needs of Windows users, you designate a non-native format for all data sets that are created and will be created in a SAS library by using a LIBNAME statement. Here is an example:

```
libname grades '/dev/app/unc' outrep=windows;
```

The libref GRADES points to the location for the application and its data sets, which will be created in WINDOWS format.

See “Host Format Values” on page 69 for supported values for the OUTREP= option.

For complete details about the LIBNAME statement, see *SAS Language Reference: Dictionary* and the companion documentation that is appropriate to your operating environment.

Using the OUTREP= Option in the DATA Step

The difference between using OUTREP= as an option in the LIBNAME statement and as an option in the DATA step is that the DATA step restricts the host file format only to the specified data set.

In this example, data set CHEM.GRADES is created in Windows format.

```
data chem.grades (outrep=windows);
  input student $ test1 test2 final;
  datalines;
Fred 66 80 70
Wilma 97 91 98
```

```
run;
```

Refer to the following section for supported values for the OUTREP= option.
For complete details about the DATA step, see *SAS Language Reference: Dictionary*.

Host Format Values

Supported values for *host-format* are:

ALPHA_OSF

ALPHA_VMS

HP_UX

MVS (specified for files in a Version 8 UNIX System Services Directory on an OS/390 host)

OS2

RS_6000_AIX

SOLARIS

VAX_VMS

WINDOWS

MIPS_ABI (Introduced in Version 8, includes HP_UX, RS_6000_AIX, and SOLARIS)

INTEL_ABI (Introduced in Version 8, Silicon Graphics (SGI) only, which was formerly included in MIPS ABI for SAS Release 6.11)

Making Accessible a File That Was Created with CEDA

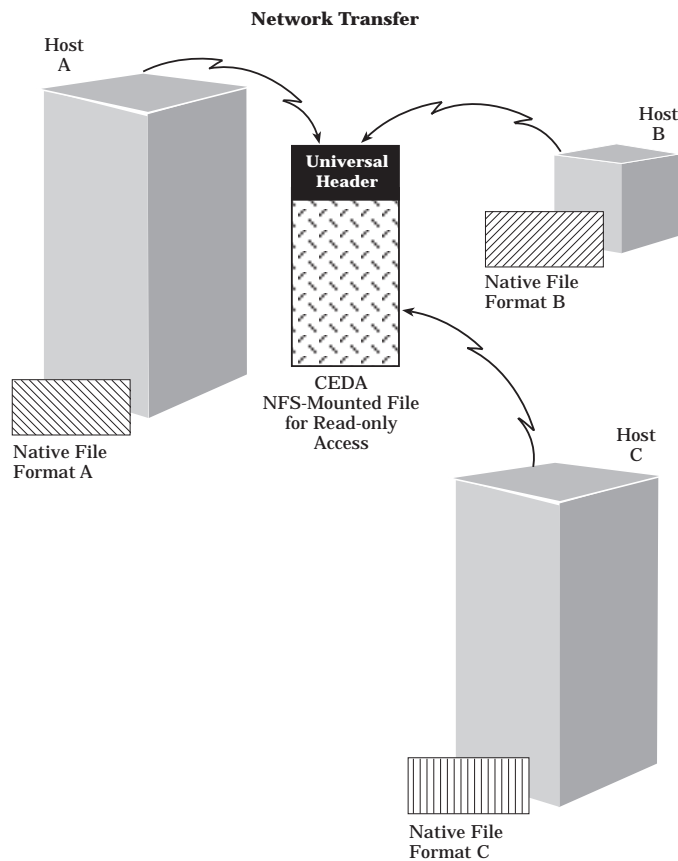
You make a file available for access through either of these methods:

- Use NFS (Network File Services) to mount the file on the network for host access. See the documentation for NFS and for your operating system.
- Use FTP (File Transfer Protocol) services to copy a file in binary format to a specific target host. See “Example of Using FTP to Transfer Transport Files” on page 34 for details about FTP.

Note: A file that is created with CEDA must be transferred in BINARY format. △

Reading a File That Was Created with CEDA

After a file has been mounted by using NFS on a network, regardless of the format identified in the OUTREP= option, any supported Version 8 host, except CMS, that is attached to the network can access the file. If the host and the file do not use the same format, CEDA is used for reading the file in read-only mode.

Figure 8.2 Reading a File That Was Created with CEDA

Remote hosts transparently access the file in read-only mode, unaware of how the file was created.

Finding Out if You Can Update a File That Was Created with CEDA

If the file is represented in the same format as that of your accessing host, then you can update it. For example, if you use a Windows host and the file was created by a Windows host, or if the `OUTREP=` option was used to designate a file formatted for Windows, then you can update the file. Otherwise, you can only read the file.

Here are two ways to find out if you can update a file that was created with CEDA.

- Try it. If you are not allowed to update the file, you receive this error message:

```
ERROR: Updating not allowed for file CHEM.CMVS because
it was created for a different operating system.
```

where CHEM is a libref and CMVS is a member name.

It is assumed that the file format is foreign to the format of the accessing host.

- Use the `CONTENTS` procedure to find out the data representation of the file. If the output includes this field entry:

```
Data Representation: FOREIGN
```

then you cannot update the file. Otherwise, there is no specification of data representation, which implies that the format of the file and the native format of the accessing host are the same.

Accessing Files That Were Created with CEDA on Personal Computer FAT Drives

CEDA creates and recognizes only Version 8 files, which are designated with an eight-character filename extension. For example, a Version 8 filename extension for a data set on the UNIX, OpenVMS, OS/2, and Windows hosts is `.sas7bdat`.

Note: Version 7 and Version 8 filename extensions are identical. △

However, the Version 8 default eight-character filename extension requirement conflicts with the three-character filename extension that is used on FAT- (File Allocation Table) style disk devices, which are used exclusively in OS/2 and Windows environments.

To solve the Version 8 long, filename-extension conflict with FAT device short filename requirements, set the `SHORTFILEEXT` option in the `LIBNAME` statement. Setting `SHORTFILEEXT` overrides the default `LONGFILEEXT`, allowing Version 8 to recognize valid three-character filename extension equivalents. An example of a short extension for a Version 8 data set is `.sd7`.

Consequently, in a Windows or an OS/2 environment, you cannot create, read from, or update a file that was created with CEDA on a FAT- (File Allocation Table) style disk device, which recognizes only files that have a three-character filename extension. For a comprehensive list of Version 8 filename extensions, see “Using Filename Extensions to Determine Which SAS Engine and Host Were Used to Create a SAS File” on page 125.

Here is an example of setting the `SHORTFILEEXT` option in the `LIBNAME` statement on an OS/2 or a Windows host:

```
libname grades 'c:\chemistry' shortfileext;
```

The `SHORTFILEEXT` option allows Version 8 to recognize short filename extensions, which enables you to create, read from, and update Version 8 files on a FAT device. For more information about the `SHORTFILEEXT` option, see *SAS Companion for the OS/2 Environment* or *SAS Companion for the Microsoft Windows Environment*, whichever is applicable.

Because it is unlikely that all users will be aware of file-naming restrictions imposed by some hardware, the need to set the `SHORTFILEEXT` option in the `LIBNAME` statement may not be apparent. To warn of this problem, you will receive this error message:

```
ERROR: File library.member.DATA has too long a
member name for the FAT drive
```

The recovery action is to retry the `LIBNAME` statement and use the `SHORTFILEEXT` option.

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.