



CHAPTER 32

SAS Catalogs

<i>Definition</i>	479
<i>Catalog Names</i>	479
<i>Parts of a Catalog Name</i>	479
<i>Accessing Information in Catalogs</i>	480
<i>Tools for Managing Catalogs</i>	480
<i>User Profile Catalog</i>	481
<i>Definition</i>	481
<i>How the Information Is Used</i>	481
<i>How SASUSER.PROFILE Is Created</i>	481
<i>Default Settings</i>	481
<i>Catalog Concatenation</i>	481
<i>Definitions</i>	481
<i>Example 1: Implicit Concatenation</i>	482
<i>Example 2: Explicit Concatenation</i>	483
<i>Rules for Catalog Concatenation</i>	484

Definition

SAS catalogs are special SAS files that store many different kinds of information in smaller units called catalog entries. Each entry has an entry type that identifies its purpose to the SAS System. A single SAS catalog can contain several different types of catalog entries. Some catalog entries contain system information such as key definitions. Other catalog entries contain application information such as window definitions, help windows, formats, informats, macros, or graphics output. You can list the contents of a catalog using various SAS features, such as SAS Explorer and PROC CATALOG.

Catalog Names

Parts of a Catalog Name

SAS catalog entries are fully identified by a four-level name in the following form:

libref.catalog.entry-name.entry-type

You commonly specify the two-level name for an entire catalog, as follows:

libref.catalog

libref

is the logical name of the SAS data library to which the catalog belongs.

catalog

is a valid SAS name for the file.

The entry name and entry type are required by some SAS procedures. If the entry type has been specified elsewhere or if it can be determined from context, you can use the entry name alone. To specify entry names and entry types, use this form:

entry-name.entry-type

entry-name

is a valid SAS name for the catalog entry.

entry-type

is assigned by SAS when the entry is created.

Accessing Information in Catalogs

In base SAS software, SAS catalog entries are generally accessed automatically by SAS when the information stored in them is required for processing. In other SAS software products, you must specify the catalog entry in various procedures. Because the requirements differ with the SAS procedure or software product, refer to the appropriate procedure or product documentation for details.

Tools for Managing Catalogs

There are several SAS features to help you manage the entries in catalogs. The CATALOG procedure and the CEXIST function are two features of base SAS software. Another tool is SAS Explorer, which lets you view the contents of SAS catalogs. A number of interactive windowing procedures contain a catalog directory window for managing entries. The following list summarizes the tools available for managing catalogs:

CATALOG procedure

is similar to the DATASETS procedure. Use the CATALOG procedure to copy, delete, list, and rename entries in catalogs.

CEXIST function

allows you to verify the existence of a SAS catalog or catalog entry. See the CEXIST function in the *SAS Language Reference: Dictionary* for more information.

CATALOG window

is a window you can bring up at any time in an interactive windowing environment. It displays the name, type, description, and date of last update for each entry in the specified catalog. CATALOG window commands enable you to edit catalog entries. You can also view and edit catalog entries after double-clicking on a catalog file in SAS Explorer.

catalog directory windows

are available in some procedures in SAS/AF, SAS/FSP, and SAS/GRAPH software. A catalog directory window lists the same kind of information that the CATALOG window provides: entry name, type, description, and date of last update. See the description of each interactive windowing procedure for details about the catalog directory window for that procedure.

User Profile Catalog

Definition

user profile catalog (SASUSER.PROFILE)

is a catalog available for customizing the way you work with SAS. SAS uses this catalog to store function key definitions, fonts for graphics applications, window attributes, and other information from interactive windowing procedures.

How the Information Is Used

The information in the SASUSER.PROFILE catalog is accessed automatically by SAS when you need it for processing. For example, each time you enter the KEYS window and change the settings, SAS stores the new settings with the KEYS entry type. Similarly, if you change and save the attributes for interactive window procedures, the changes are stored under the appropriate entry name and type. When you use the window or procedure, SAS then looks for information in the user profile catalog.

How SASUSER.PROFILE Is Created

SAS creates the user profile catalog the first time it tries to refer to it and discovers that it does not exist. If you are using an interactive windowing environment, this occurs during system initialization in your first SAS session. If you use one of the other modes of execution, the user profile catalog is created the first time you execute a SAS procedure that requires it.

Operating Environment Information: The SASUSER library is implemented differently in various operating environments. See the SAS documentation for your host system for more information on how and when it is created. △

Default Settings

The default settings for your SAS session are stored in several catalogs in the SASHELP installation library. If you do not make any changes to key settings or other options, SAS uses the default settings. If you make changes, the new information is stored in your user profile catalog. To restore the original default settings, use the CATALOG procedure or CATALOG window to delete the appropriate entries from your user profile catalog. By default, SAS then uses the corresponding entry from the SASHELP library.

Catalog Concatenation

Definitions

You can logically combine two or more SAS catalogs by concatenating them. This allows you to access the contents of several catalogs, using one catalog name. There are two types of concatenation, *explicit* and *implicit*.

Implicit catalog concatenation

results from a concatenation of libraries through a LIBNAME statement. When two or more libraries are logically combined through concatenation, any catalogs with the same name in each library become logically combined as well.

Explicit catalog concatenation

is a concatenation that is specified by the global CATNAME statement in which the catalogs to be concatenated are specifically (or explicitly) named. During explicit catalog concatenation, the CATNAME statement sets up a logical catalog in memory.

Example 1: Implicit Concatenation

This LIBNAME statement concatenates the two SAS data libraries:

```
libname both ('SAS-Data-library 1' 'SAS-Data-library 2');
```

If library1 contains these members ...	and library2 contains these members ...
MYCAT.CATALOG	MYCAT.CATALOG
TABLE 1.DATA	MYCAT2.CATALOG
TABLE3.DATA	TABLE1.DATA
	TABLE1.INDEX
	TABLE2.DATA
	TABLE2.INDEX

The concatenated libref BOTH would have the following:

Concatenated libref BOTH

MYCAT.CATALOG (from path 1 and 2)
 MYCAT2.CATALOG (from path 2)
 TABLE1.DATA (from path 1)
 TABLE2.DATA (from path2)
 TABLE2.INDEX (from path2)
 TABLE3.DATA (from path1)

Notice that TABLE1.INDEX does not show up in the concatenation but TABLE2.INDEX does. SAS suppresses listing the index when its associated data file is not part of the concatenation.

So what happened to the catalogs when the libraries were concatenated? A resulting catalog now exists logically in memory, with the full name BOTH.MYCAT.CATALOG. This catalog combines each of the two physical catalogs residing in 'library 1' and 'library2', called MYCAT.CATALOG.

To understand the contents of the concatenation BOTH.MYCAT, first look at the contents of both parts of the concatenation. Assume that the two original MYCAT.CATALOG files contain the following:

MYCAT.CATALOG in library1 contains ...	MYCAT.CATALOG in library 2 contains ...
A.FRAME	A.GRSEG
C.FRAME	B.FRAME
	C.FRAME

The combined catalog BOTH.MYCAT contains:

BOTH.MYCAT
A.GRSEG (from path 2)
A.FRAME (from path 1)
B.FRAME (from path 2)
C.FRAME (from path 1)

Example 2: Explicit Concatenation

The syntax of the CATNAME statement is:

```
CATNAME libref.catref
      (libref-1.catalog--1 (ACCESS=READONLY)
      libref-n.catalog-n (ACCESS=READONLY));
```

To disassociate a concatenated catalog the syntax is:

```
CATNAME libref.catref | _ALL_ clear;
```

In the following example, there must be a libref that is defined and named CATDOG. The libref catdog establishes the scope for the explicit concatenation definition.

Note: If a file in catdog named COMBINED.CATALOG already exists, it becomes unaccessible until the explicit concatenation CATDOG.COMBINED is cleared.

△

If library1 contains these members ...	and library2 contains these members ...
MYCAT.CATALOG	MYDOG.CATALOG
TABLE1.DATA	MYCAT2.CATALOG
TABLE3.DATA	TABLE1.DATA
	TABLE1.INDEX
	TABLE2.DATA
	TABLE2.INDEX

and if we issue the following statement,

```
CATNAME catdog.combined
      (library1.mycat (ACCESS=READONLY)
      library2.mydog (ACCESS=READONLY));
```

then the concatenated catalog CATDOG.COMBINED would combine the following catalogs:

 Concatenated catalog CATALOG.COMBINED

MYCAT.CATALOG (from library 1)

MYDOG.CATALOG (from library 2)

Note: In explicit concatenation only the named catalogs are combined. In implicit concatenation, any catalogs that happen to have the same name in their respective libraries are concatenated when those libraries are concatenated. Δ

The previous CATNAME statement creates a catalog that exists logically in memory, named CATDOG.COMBINED.CATALOG, which combines the two physical catalogs residing in library1 and library2, called MYCAT.CATALOG and MYDOG.CATALOG.

To understand the contents of the concatenation COMBINED.CATALOG, first look at the contents of both parts of the concatenation. The two original catalog files contain the following entries:

MYCAT.CATALOG	MYDOG.CATALOG
library 1	library 2
A.FRAME	A.GRSEG
C.FRAME	B.FRAME
	C.FRAME

The concatenated catalog COMBINED contains:

 COMBINED.CATALOG contents

A.GRSEG (from MYDOG)

A.FRAME (from MYCAT)

B.FRAME (from MYDOG)

C.FRAME (from MYCAT)

Rules for Catalog Concatenation

The rules for catalog concatenation are the same, whether the the catalogs are implicitly or explicitly concatenated.

- When a catalog entry is open for input or update, the parts are searched and the first occurrence of the specified entry is used.
- When an item is open for output, it will be created in the first catalog listed in the concatenation.

Note: A new catalog entry is created in the first catalog even if there is an item with the same name in another part of the concatenation. Δ

Note: If the first catalog in a concatenation that is opened for update does not exist, the item will be written to the next catalog that exists in the concatenation. Δ

- When you want to delete or rename a catalog entry, only the first occurrence of the entry is affected.

- Any time a list of catalog entries is displayed, only one occurrence of the catalog entry is shown.

Note: Even if a catalog entry occurs multiple times in the concatenation, only the first occurrence is shown. △

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.