CHAPTER 6

The CATALOG Procedure

Overview **155**
Procedure Syntax **156**
PROC CATALOG Statement **157**
CHANGE Statement **158**
CONTENTS Statement **159**
COPY Statement **160**
DELETE Statement **162**
EXCHANGE Statement **162**
EXCLUDE Statement **163**
MODIFY Statement **164**
SAVE Statement **164**
SELECT Statement **165**
Concepts **166**
Interactive Processing with RUN Groups **166**
Definition **166**
How to End a PROC CATALOG Step **166**
Error Handling and RUN Groups **166**
Specifying an Entry Type **167**
Four Ways to Supply an Entry Type **167**
Why Use the ENTRYTYPE= Option? **167**
Avoid a Common Error **167**
The ENTRYTYPE= Option **168**
Catalog Concatenation **169**
Restrictions **169**
Results **170**
Examples **170**
Example 1: Copying, Deleting, and Moving Catalog Entries from Multiple Catalogs **170**
Example 2: Displaying Contents, Changing Names, and Changing a Description **174**

Overview

The CATALOG procedure manages entries in SAS catalogs. PROC CATALOG is an interactive, statement-driven procedure that enables you to

- □ create a listing of the contents of a catalog
- □ copy a catalog or selected entries within a catalog
- □ rename, exchange, or delete entries within a catalog
- □ change the name of a catalog entry
- □ modify, by changing or deleting, the description of a catalog entry.

For more information on SAS data libraries and catalogs, refer to *SAS Language Reference: Concepts*.

To learn how to use the SAS windowing environment to manage entries in a SAS catalog, see the SAS online Help for the SAS Explorer window. You may prefer to use the Explorer window instead of using PROC CATALOG. The window can do most of what the procedure does.

Procedure Syntax

Tip: Supports RUN-group processing

Tip: Supports the Output Delivery System. See Chapter 2, "Fundamental Concepts for Using Base SAS Procedures. "

Reminder: You can perform similar functions with the SAS Explorer window and with dictionary tables in the SQL procedure. For information on the Explorer window, see the online Help. For information on PROC SQL, see Chapter 34, "The SQL Procedure," on page 1021.

PROC CATALOG CATALOG=<*libref.*>*catalog* <ENTRYTYPE=*etype*> <FORCE>
    <KILL>;
  CONTENTS <OUT=*SAS-data-set*> <FILE=*fileref*>;
  COPY OUT=<*libref.*>*catalog* <*options*>;
    SELECT *entry(s)* </ ENTRYTYPE=*etype*>;
    EXCLUDE *entry(s)* </ ENTRYTYPE=*etype*>;
  CHANGE *old-name-1=new-name-1*
      <*…old-name-n=new-name-n*>
      </ ENTRYTYPE=*etype*>;
  EXCHANGE *name-1=other-name-1*
      <*…name-n=other-name-n*>
      </ ENTRYTYPE=*etype*>;
  DELETE *entry(s)*
      </ ENTRYTYPE=*etype*>;
  MODIFY *entry*
      (DESCRIPTION=<<'>*entry-description*<'>>)
      </ ENTRYTYPE=*etype*>;
  SAVE *entry(s)* </ ENTRYTYPE=*etype*>;

To do this	Use this statement
Copy entries from one SAS catalog to another	
Copy or move all entries	COPY (with MOVE option)
Copy entries to a new catalog (overwriting the catalog if it already exists)	COPY (with NEW option)
Copy only selected entries	COPY, SELECT
Copy all *except* the entries specified	COPY, EXCLUDE
Delete entries from a SAS catalog	
Delete *all* entries	PROC CATALOG (with KILL option)

To do this	Use this statement
Delete *all* entries in catalog opened by another process	PROC CATALOG (with FORCE and KILL options)
Delete specified entries	DELETE
Delete all *except* the entries specified	SAVE
Alter names and descriptions	
Change the names of catalog entries	CHANGE
Switch the names of two catalog entries	EXCHANGE
Change the description of a catalog entry	MODIFY
Print	
Print the contents of a catalog	CONTENTS

PROC CATALOG Statement

PROC CATALOG CATALOG=<*libref.*>*catalog* <ENTRYTYPE=*etype*> <FORCE> <KILL>;

To do this	Use this option
Restrict processing to one entry type	ENTRYTYPE=
Delete all catalog entries	KILL
Force certain statements to execute on a catalog opened by another process	FORCE

Required Arguments

CATALOG=<*libref.*>*catalog*
specifies the SAS catalog to process.

Alias: CAT=, C=

Default: If ENTRYTYPE= is not specified, PROC CATALOG processes all entries in the catalog.

Options

ENTRYTYPE=*etype*
restricts processing of the current PROC CATALOG step to one entry type.

Alias: ET=

Default: If you omit ENTRYTYPE=, PROC CATALOG processes all entries in a catalog.

Interaction: The specified entry type applies to any one-level entry names used in a subordinate statement. You cannot override this specification in a subordinate statement.

Interaction: ENTRYTYPE= does not restrict the effects of the KILL option.

Tip: In order to process multiple entry types in a single PROC CATALOG step, use ENTRYTYPE= in a subordinate statement, not in the PROC CATALOG statement.

See also: "Specifying an Entry Type" on page 167.

Featured in: Example 1 on page 170 and Example 2 on page 174

FORCE
forces statements to execute on a catalog opened by another process.

Some CATALOG statements require exclusive access to the catalog they operate on if the statement can radically change the contents of a catalog. If exclusive access cannot be obtained, the action fails. The statements and the catalogs that are affected are

KILL	affects the specified catalog
COPY	affects the OUT= catalog
COPY MOVE	affects the IN= and the OUT= catalogs
SAVE	affects the specified catalog.

Tip: Use FORCE to execute the statement, even if exclusive access cannot be obtained.

KILL
deletes all entries in a SAS catalog.

Interaction: The KILL option deletes all catalog entries even when ENTRYTYPE= is specified.

Interaction: The SAVE statement has no effect because the KILL option deletes all entries in a SAS catalog before any other statements are processed.

Tip: KILL deletes all entries but does not remove an empty catalog from the SAS data library. You must use another method, such as PROC DATASETS or the DIR window to delete an empty SAS catalog.

CAUTION:
Do not attempt to limit the effects of the KILL option. This option deletes all entries in a SAS catalog before any option or other statement takes effect. △

CHANGE Statement

Renames one or more catalog entries.

Tip: You can change multiple names in a single CHANGE statement or use multiple CHANGE statements.

Featured in: Example 2 on page 174

CHANGE *old-name-1=new-name-1*
<*…old-name-n=new-name-n*>
</ ENTRYTYPE=*etype*>;

Required Arguments

old-name=new-name
specifies the current name of a catalog entry and the new name you want to assign to it. Specify any valid SAS name.

Restriction: You must designate the type of the entry, either with the name (*ename.etype*) or with the
ENTRYTYPE= option.

Options

ENTRYTYPE=*etype*
restricts processing to one entry type.

See: "The ENTRYTYPE= Option" on page 168

See also: "Specifying an Entry Type" on page 167

CONTENTS Statement

Lists the contents of a catalog in the procedure output or writes a list of the contents to a SAS data set, an external file, or both.

Featured in: Example 2 on page 174

CONTENTS <OUT=*SAS-data-set*> <FILE=*fileref*>;

Without Options

The output is sent to the procedure output.

Options

CATALOG=<*libref.*>*catalog*
specifies the SAS catalog to process.

Alias: CAT=, C=

Default: None

FILE=*fileref*
sends the contents to an external file, identified with a SAS fileref.

Interaction: If *fileref* has not been previously assigned to a file, then the file is created and named according to operating environment-dependent rules for external files.

OUT=*SAS-data-set*
sends the contents to a SAS data set. When the statement executes, a message on the SAS log reports that a data set has been created. The data set contains six variables in this order:

LIBNAME	the libref
MEMNAME	the catalog name
NAME	the names of entries
TYPE	the types of entries
DESC	the descriptions of entries
DATE	the dates entries were last modified.

COPY Statement

Copies some or all of the entries in one catalog to another catalog.

Restriction: A COPY statement's effect ends at a RUN statement or at the beginning of a statement other than the SELECT or EXCLUDE statement.

Tip: Use SELECT or EXCLUDE statements, but not both, after the COPY statement to limit which entries are copied.

Tip: You can copy entries from multiple catalogs in a single PROC step, not just the one specified in the PROC CATALOG statement.

Tip: The ENTRYTYPE= option does not require a forward slash (/) in this statement.

Featured in: Example 1 on page 170

COPY OUT=<*libref.*>*catalog* <*options*>;

To do this	Use this option
Restrict processing to one type of entry	ENTRYTYPE=
Copy from a different catalog in the same step	IN=
Move (copy and then delete) a catalog entry	MOVE
Copy entries to a new catalog (overwriting the catalog if it already exists)	NEW
Protect several types of SAS/AF entries from being edited with PROC BUILD	NOEDIT
Not copy source lines from a PROGRAM, FRAME, or SCL entry	NOSOURCE

Required Arguments

OUT=<*libref.*>*catalog*
names the catalog to which entries are copied.

Options

ENTRYTYPE=*etype*
restricts processing to one entry type for the current COPY statement and any subsequent SELECT or
EXCLUDE statements.

See: "The ENTRYTYPE= Option" on page 168

See also: "Specifying an Entry Type" on page 167

IN=<*libref.*>*catalog*
specifies the catalog to copy.

Interaction: The IN= option overrides a CATALOG= argument that was specified in the PROC CATALOG statement.

Featured in: Example 1 on page 170

MOVE
deletes the original catalog or entries after the new copy is made.

Interaction: When MOVE removes all entries from a catalog, the procedure deletes the catalog from the library.

NEW
overwrites the destination (specified by OUT=) if it already exists. If you omit NEW, PROC CATALOG updates the destination.

NOEDIT
prevents the copied version of the following SAS/AF entry types from being edited by the BUILD procedure:

CBT	PROGRAM
FRAME	SCL
HELP	SYSTEM
MENU	

Restriction: If you specify the NOEDIT option for an entry that is not one of these types, it is ignored.

Tip: When creating SAS/AF applications for other users, use NOEDIT to protect the application by preventing certain catalog entries from being altered.

Featured in: Example 1 on page 170

NOSOURCE
omits copying the source lines when you copy a
SAS/AF PROGRAM, FRAME, or SCL entry.

Alias: NOSRC

Restriction: If you specify this option for an entry other than a PROGRAM, FRAME, or SCL entry, it is ignored.

DELETE Statement

Deletes entries from a SAS catalog.

Tip: Use DELETE to delete only a few entries; use SAVE when it is more convenient to specify which entries *not* to delete.

Tip: You can specify multiple entries. You can also use multiple DELETE statements.

See also: "SAVE Statement" on page 164

Featured in: Example 1 on page 170

DELETE *entry(s)* </ ENTRYTYPE=*etype*>;

Required Arguments

entry(s)
specifies the name of one or more SAS catalog entries.

Restriction: You must designate the type of the entry, either with the name (*ename.etype*) or with the ENTRYTYPE= option.

Options

ENTRYTYPE=*etype*
restricts processing to one entry type.

See: "The ENTRYTYPE= Option" on page 168

See also: "Specifying an Entry Type" on page 167

EXCHANGE Statement

Switches the name of two catalog entries.

Restriction: The catalog entries must be of the same type.

EXCHANGE *name-1=other-name-1*
<*…name-n=other-name-n*>
</ ENTRYTYPE=*etype*>;

Required Arguments

name=other-name
specifies two catalog entry names that the procedure will switch.

Interaction: You can specify only the entry name without the entry type if you use the ENTRYTYPE= option on either the PROC CATALOG statement or the EXCHANGE statement.

See also: “Specifying an Entry Type” on page 167

Options

ENTRYTYPE=*etype*
restricts processing to one entry type.

See: “The ENTRYTYPE= Option” on page 168

See also: “Specifying an Entry Type” on page 167

EXCLUDE Statement

Specifies entries that the COPY statement does *not* copy.

Restriction: Requires the COPY statement.

Restriction: Do not use the EXCLUDE statement with the SELECT statement.

Tip: You can specify multiple entries in a single EXCLUDE statement.

Tip: You can use multiple EXCLUDE statements with a single COPY statement within a RUN group.

See also: “COPY Statement” on page 160 and “SELECT Statement” on page 165

Featured in: Example 1 on page 170

EXCLUDE *entry(s)* </ ENTRYTYPE=*etype*>;

Required Arguments

entry(s)
specifies the name of one or more SAS catalog entries.

Restriction: You must designate the type of the entry, either when you specify the name (*ename.etype*) or with the ENTRYTYPE= option.

See also: “Specifying an Entry Type” on page 167

Options

ENTRYTYPE=*etype*
restricts processing to one entry type.

See: “The ENTRYTYPE= Option” on page 168

See also: “Specifying an Entry Type” on page 167

MODIFY Statement

Changes the description of a catalog entry.

Featured in: Example 2 on page 174

MODIFY *entry*
 (DESCRIPTION=<<'>*entry-description*<'>>)
 </ ENTRYTYPE=*etype*>;

Required Arguments

entry
specifies the name of one SAS catalog entry. Optionally, you can specify the entry type with the name.

Restriction: You must designate the type of the entry, either when you specify the name (*ename.etype*) or with the ENTRYTYPE= option.

See also: "Specifying an Entry Type" on page 167

DESCRIPTION=<<'>*entry-description*<'>>
changes the description of a catalog entry by replacing it with a new description, up to 40 characters long, or by removing it altogether. Optionally, you can enclose the description in single or double quotes.

Alias: DESC

Tip: Use DESCRIPTION= with no text to remove the current description.

Options

ENTRYTYPE=*etype*
restricts processing to one entry type.

See: "The ENTRYTYPE= Option" on page 168

See also: "Specifying an Entry Type" on page 167

SAVE Statement

Specify entries *not* to delete from a SAS catalog.

Restriction: Cannot limit the effects of the KILL option.

Tip: Use SAVE to delete all but a few entries in a catalog. Use DELETE when it is more convenient to specify which entries to delete.

Tip: You can specify multiple entries and use multiple SAVE statements.

See also: "DELETE Statement" on page 162

SAVE *entry(s)* </ ENTRYTYPE=*etype*>;

Required Arguments

entry(s)
specifies the name of one or more SAS catalog entries.

Restriction: You must designate the type of the entry, either with the name (*ename.etype*) or with the
ENTRYTYPE= option.

Options

ENTRYTYPE=*etype*
restricts processing to one entry type.

See: "The ENTRYTYPE= Option" on page 168

See also: "Specifying an Entry Type" on page 167

SELECT Statement

Specifies entries that the COPY statement will copy.

Restriction: Requires the COPY statement.

Restriction: Cannot be used with an EXCLUDE statement.

Tip: You can specify multiple entries in a single SELECT statement.

Tip: You can use multiple SELECT statements with a single COPY statement within a RUN group.

See also: "COPY Statement" on page 160 and "EXCLUDE Statement" on page 163

Featured in: Example 1 on page 170

SELECT *entry(s)* </ ENTRYTYPE=*etype*>;

Required Arguments

entry(s)
specifies the name of one or more SAS catalog entries.

Restriction: You must designate the type of the entry, either when you specify the name (*ename.etype*) or with the ENTRYTYPE= option.

Options

ENTRYTYPE=*etype*
restricts processing to one entry type.

See: "The ENTRYTYPE= Option" on page 168.

See also: "Specifying an Entry Type" on page 167.

Concepts

Interactive Processing with RUN Groups

Definition

The CATALOG procedure is interactive. Once you submit a PROC CATALOG statement, you can continue to submit and execute statements or groups of statements without repeating the PROC CATALOG statement.

A set of procedure statements ending with a RUN statement is called a *RUN group*. The changes specified in a given group of statements take effect when a RUN statement is encountered.

How to End a PROC CATALOG Step

In the DATA step and most SAS procedures, a RUN statement is a step boundary and ends the step. A simple RUN statement does not, however, end an interactive procedure. To terminate a PROC CATALOG step, you can

- □ submit a QUIT statement
- □ submit a RUN statement with the CANCEL option
- □ submit another DATA or PROC statement
- □ end your SAS session.

Note: When you enter a QUIT, DATA, or PROC statement, any statements following the last RUN group execute before the CATALOG procedure terminates. If you enter a RUN statement with the CANCEL option, however, the remaining statements *do not execute* before the procedure ends. △

See Example 2 on page 174.

Error Handling and RUN Groups

Error handling is based in part on the division of statements into RUN groups. If a syntax error is encountered, *none* of the statements in the current RUN group execute, and execution proceeds to the next RUN group.

For example, the following statements contain a misspelled DELETE statement:

```
proc catalog catalog=misc entrytype=help;
   copy out=drink;
      select coffee tea;
   del juices;        /* INCORRECT!!! */
   exchange glass=plastic;
run;
   change calstats=nutri;
run;
```

Because the DELETE statement is incorrectly specified as DEL, no statements in that RUN group execute, *except* the PROC CATALOG statement itself. The CHANGE statement does execute, however, because it is in a different RUN group.

CAUTION:
Be careful when setting up batch jobs in which one RUN group's statements depend on the effects of a previous RUN group, especially when deleting and renaming entries. △

Specifying an Entry Type

Four Ways to Supply an Entry Type

There is no default entry type, so if you do not supply one, PROC CATALOG generates an error. You can supply an entry type in one of four ways. See Table 6.1 on page 167.

Table 6.1 Supplying an Entry Type

You can supply an entry type with...	Example
the entry name	`delete test1.program test1.log test2.log;`
ET= in parenthesis	`delete test1 (et=program);`
ET= *after* a slash[1]	`delete test1 (et=program) test1 test2 / et=log;`
ENTRYTYPE= *without* a slash[2]	`proc catalog catalog=mycat et=log; delete test1 test2;`

1 in a subordinate statement
2 in the PROC CATALOG or the COPY statement

All statements, except the CONTENTS statement, accept the ENTRYTYPE= (alias ET=) option.

Why Use the ENTRYTYPE= Option?

ENTRYTYPE= can save keystrokes when you are processing multiple entries of the same type.

To create a default for entry type for all statements in the current step, use ENTRYTYPE= in the PROC CATALOG statement. To set the default for only the current statement, use ENTRYTYPE= in a subordinate statement.

If many entries are of one type, but a few are of other types, you can use ENTRYTYPE= to specify a default and then override that for individual entries with (ENTRYTYPE=) *in parenthesis* after those entries.

Avoid a Common Error

You cannot specify the ENTRYTYPE= option in both the PROC CATALOG statement and a subordinate statement. For example, these statements generate an error and do not delete any entries because the ENTRYTYPE= specifications contradict each other:

```
/* THIS IS INCORRECT CODE. */
proc catalog cat=sample et=help;
   delete a b c / et=program;
run;
```

The ENTRYTYPE= Option

The ENTRYTYPE= option is available in every statement in the CATALOG procedure except CONTENTS.

ENTRYTYPE=*etype*
not in parenthesis, sets a default entry type for the entire PROC step when used in the PROC CATALOG statement. In all other statements, this option sets a default entry type for the *current* statement.

Alias: ET=

Default: If you omit ENTRYTYPE=, PROC CATALOG processes all entries in the catalog.

Interaction: If you specify ENTRYTYPE= in the PROC CATALOG statement, do not specify either ENTRYTYPE= or (ENTRYTYPE=) in a subordinate statement.

Interaction: (ENTRYTYPE=*etype*) *in parenthesis* immediately following an entry name overrides
ENTRYTYPE= *in that same statement*.

Tip: On all statements *except* the PROC CATALOG and COPY statements, this option follows a slash.

Tip: To process multiple entry types in a single PROC CATALOG step, use ENTRYTYPE= in a subordinate statement, not in the PROC CATALOG statement.

See also: "Specifying an Entry Type" on page 167.

Featured in: Example 1 on page 170

(ENTRYTYPE=*etype*)
in parenthesis, identifies the type of the entry just preceding it.

Alias: (ET=)

Restriction: (ENTRYTYPE=*etype*) immediately following an entry name in a subordinate statement *cannot override* an ENTRYTYPE= option *in the PROC CATALOG statement*. It generates a syntax error.

Interaction: (ENTRYTYPE=*etype*) immediately following an entry name overrides ENTRYTYPE= *in that same statement*.

Tip: This form is useful mainly for specifying exceptions to an ENTRYTYPE= option used in a subordinate statement. The following statement deletes A.HELP, B.FORMAT, and C.HELP:

```
delete a b (et=format) c / et=help;
```

Tip: For the CHANGE and EXCHANGE statements, specify (ENTRYTYPE=) *in parenthesis* only once for each pair of names following the second name in the pair. For example,

```
change old1=new1 (et=log)
       old1=new2 (et=help);
```

See also: "Specifying an Entry Type" on page 167

Featured in: Example 1 on page 170 and Example 2 on page 174

Catalog Concatenation

Catalog concatenation was not available prior to Version 7 of the SAS System . The CATALOG procedure supports both implicit and explicit concatenation of catalogs. All statements and options that can be used on single (unconcatenated) catalogs can be used on catalog concatenations.

Restrictions

When you use the CATALOG procedure to copy concatenated catalogs and you use the NEW option, the following rules apply:

1 If the input catalog is a concatenation and if the output catalog exists in any level of the input concatentation, the copy is not allowed.

2 If the output catalog is a concatenation and if the input catalog exists in the first level of the output concatenation, the copy is not allowed.

For example, the following code demonstrates these two rules, and the copy fails:

```
libname first 'path-name1';
libname second 'path-name2';
/* create contat.x */
libname concat (first second);

/* fails rule #1 */
proc catalog c=concat.x;
   copy out=first.x new;
run;
quit;

/* fails rule #2 */
proc catalog c=first.x;
   copy out=concat.x new;
run;
quit;
```

In summary, the following table shows when copies are allowed. In the table, A and B are libraries, and each contains catalog X. Catalog C is an implicit concatenation of A and B, and catalog D is an implicit concatenation of B and A.

Input catalog	Output catalog	Copy allowed?
C.X	B.X	No
C.X	D.X	No
D.X	C.X	No
A.X	A.X	No
A.X	B.X	Yes
B.X	A.X	Yes
C.X	A.X	No

B.X	C.X	Yes
A.X	C.X	No

Results

- *SAS Log:* In most cases, the CATALOG procedure writes the list of entries being processed to the SAS log.
- *Output from the CONTENTS statement:* By default, PROC CATALOG routes output produced by the CONTENTS statement to the procedure output listing. You can use the FILE= or OUT= options to redirect the output to an external file or to a SAS data set.

Examples

Example 1: Copying, Deleting, and Moving Catalog Entries from Multiple Catalogs

Procedure features:

PROC CATALOG statement:
  CATALOG= argument
COPY statement options:
  IN=
  MOVE
  NOEDIT
DELETE statement options:
  ENTRYTYPE= or ET=
EXCLUDE statement options:
  ENTRYTYPE= or ET=
  (ENTRYTYPE=) or (ET=)
QUIT statement
RUN statement
SELECT statement options:
  ENTRYTYPE= or ET=

This example

- copies entries by excluding a few entries
- copies entries by specifying a few entries
- protects entries from being edited
- moves entries
- deletes entries
- processes entries from multiple catalogs
- processes entries in multiple run groups.

Input Catalogs

The SAS catalog PERM.SAMPLE contains the following entries:

```
DEFAULT  FORM     Default form for printing  06/16/96
FSLETTER FORM     Standard form for letters  06/16/96
LOAN     FRAME    Loan analysis application  06/16/96
LOAN     HELP     Information about the app. 06/16/96
BUILD    KEYS     Function Key Definitions   06/16/96
LOAN     KEYS     Custom key def. for app.   06/16/96
CREDIT   LOG      credit application log     06/16/96
TEST1    LOG      Inventory program          06/16/96
TEST2    LOG      Inventory program          06/16/96
TEST3    LOG      Inventory program          06/16/96
LOAN     PMENU    Custom menu def. for app.  06/16/96
CREDIT   PROGRAM  credit application pgm     06/16/96
TEST1    PROGRAM  testing budget applic.     06/16/96
TEST2    PROGRAM  testing budget applic.     06/16/96
TEST3    PROGRAM  testing budget applic.     06/16/96
LOAN     SCL      loan analysis SCL code     06/16/96
PASSIST  SLIST    User profile               06/16/96
PRTINFO  XPRINTER Printing Parameters        06/16/96
```

The SAS catalog PERM.FORMATS contains the following entries:

```
REVENUE   FORMAT   FORMAT:MAXLEN=16,16,12   06/16/96
DEPT      FORMATC  FORMAT:MAXLEN=1,1,14     06/21/96
```

Program

The SAS system option SOURCE writes the source code to the log.

```
options nodate pageno=1 linesize=80 pagesize=60 source;
```

The LIBNAME statement assigns a libref to a SAS data library that contains a permanent SAS catalog.

```
libname perm 'SAS-data-library';
```

The DELETE statement deletes two entries from the PERM.SAMPLE catalog.

```
proc catalog cat=perm.sample;
   delete credit.program credit.log;
run;
```

The COPY statement copies all entries in the PERM.SAMPLE catalog to the WORK.TCATALL catalog.

```
   copy out=tcatall;
run;
```

The COPY and EXCLUDE statements copy everything except three LOG entries and PASSIST.SLIST from PERM.SAMPLE to WORK.TESTCAT. The EXCLUDE statement specifies which entries not to copy. ET= specifies a default type. (ET=) specifies an exception to the default type.

```
   copy out=testcat;
      exclude test1 test2 test3  passist (et=slist) / et=log;
run;
```

The COPY and SELECT statements and the MOVE option move three LOG entries from PERM.SAMPLE to WORK.LOGCAT. The SELECT statement specifies which entries to move. ET= restricts processing to LOG entries.

```
   copy out=logcat move;
      select test1 test2 test3 / et=log;
run;
```

The COPY and SELECT statements copy five SAS/AF software entries from PERM.SAMPLE to PERM.FINANCE. The NOEDIT option protects these entries in PERM.FINANCE from further editing with PROC BUILD.

```
   copy out=perm.finance noedit;
      select loan.frame loan.help loan.keys loan.pmenu;
run;
```

The COPY and SELECT statements copy two formats from PERM.FORMATS to PERM.FINANCE. The IN= option enables you to copy from a different catalog than the one specified in the PROC CATALOG statement. Note the entry types for numeric and character formats: REVENUE.FORMAT is a numeric format and DEPT.FORMATC is a character format. The COPY and SELECT statements execute before the QUIT statement ends the PROC CATALOG step.

```
   copy in=perm.formats out=perm.finance;
      select revenue.format dept.formatc;
quit;
```

Log

```
1    libname perm 'SAS-data-library';
NOTE: Directory for library PERM contains files of mixed engine types.
NOTE: Libref PERM was successfully assigned as follows:
      Engine:        V7
      Physical Name: 'SAS-data-library'
2    options nodate pageno=1 linesize=80 pagesize=60 source;
3   proc catalog cat=perm.sample;
4      delete credit.program credit.log;
5   run;
NOTE: Deleting entry CREDIT.PROGRAM in catalog PERM.SAMPLE.
NOTE: Deleting entry CREDIT.LOG in catalog PERM.SAMPLE.
6      copy out=tcatall;
7   run;
NOTE: Copying entry DEFAULT.FORM from catalog PERM.SAMPLE to catalog
      WORK.TCATALL.
NOTE: Copying entry FSLETTER.FORM from catalog PERM.SAMPLE to catalog
      WORK.TCATALL.
NOTE: Copying entry LOAN.FRAME from catalog PERM.SAMPLE to catalog WORK.TCATALL.
NOTE: Copying entry LOAN.HELP from catalog PERM.SAMPLE to catalog WORK.TCATALL.
NOTE: Copying entry BUILD.KEYS from catalog PERM.SAMPLE to catalog WORK.TCATALL.
NOTE: Copying entry LOAN.KEYS from catalog PERM.SAMPLE to catalog WORK.TCATALL.
NOTE: Copying entry TEST1.LOG from catalog PERM.SAMPLE to catalog WORK.TCATALL.
NOTE: Copying entry TEST2.LOG from catalog PERM.SAMPLE to catalog WORK.TCATALL.
NOTE: Copying entry TEST3.LOG from catalog PERM.SAMPLE to catalog WORK.TCATALL.
NOTE: Copying entry LOAN.PMENU from catalog PERM.SAMPLE to catalog WORK.TCATALL.
NOTE: Copying entry TEST1.PROGRAM from catalog PERM.SAMPLE to catalog
      WORK.TCATALL.
NOTE: Copying entry TEST2.PROGRAM from catalog PERM.SAMPLE to catalog
      WORK.TCATALL.
NOTE: Copying entry TEST3.PROGRAM from catalog PERM.SAMPLE to catalog
      WORK.TCATALL.
NOTE: Copying entry LOAN.SCL from catalog PERM.SAMPLE to catalog WORK.TCATALL.
NOTE: Copying entry PASSIST.SLIST from catalog PERM.SAMPLE to catalog
      WORK.TCATALL.
NOTE: Copying entry PRTINFO.XPRINTER from catalog PERM.SAMPLE to catalog
      WORK.TCATALL.
```

```
8      copy out=testcat;
9         exclude test1 test2 test3  passist (et=slist) / et=log;
10   run;
NOTE: Copying entry DEFAULT.FORM from catalog PERM.SAMPLE to catalog
      WORK.TESTCAT.
NOTE: Copying entry FSLETTER.FORM from catalog PERM.SAMPLE to catalog
      WORK.TESTCAT.
NOTE: Copying entry LOAN.FRAME from catalog PERM.SAMPLE to catalog WORK.TESTCAT.
NOTE: Copying entry LOAN.HELP from catalog PERM.SAMPLE to catalog WORK.TESTCAT.
NOTE: Copying entry BUILD.KEYS from catalog PERM.SAMPLE to catalog WORK.TESTCAT.
NOTE: Copying entry LOAN.KEYS from catalog PERM.SAMPLE to catalog WORK.TESTCAT.
NOTE: Copying entry LOAN.PMENU from catalog PERM.SAMPLE to catalog WORK.TESTCAT.
NOTE: Copying entry TEST1.PROGRAM from catalog PERM.SAMPLE to catalog
      WORK.TESTCAT.
NOTE: Copying entry TEST2.PROGRAM from catalog PERM.SAMPLE to catalog
      WORK.TESTCAT.
NOTE: Copying entry TEST3.PROGRAM from catalog PERM.SAMPLE to catalog
      WORK.TESTCAT.
NOTE: Copying entry LOAN.SCL from catalog PERM.SAMPLE to catalog WORK.TESTCAT.
NOTE: Copying entry PRTINFO.XPRINTER from catalog PERM.SAMPLE to catalog
      WORK.TESTCAT.
11      copy out=logcat move;
12         select test1 test2 test3 / et=log;
13   run;
NOTE: Moving entry TEST1.LOG from catalog PERM.SAMPLE to catalog WORK.LOGCAT.
NOTE: Moving entry TEST2.LOG from catalog PERM.SAMPLE to catalog WORK.LOGCAT.
NOTE: Moving entry TEST3.LOG from catalog PERM.SAMPLE to catalog WORK.LOGCAT.
14      copy out=perm.finance noedit;
15         select loan.frame loan.help loan.keys loan.pmenu;
16   run;
NOTE: Copying entry LOAN.FRAME from catalog PERM.SAMPLE to catalog PERM.FINANCE.
NOTE: Copying entry LOAN.HELP from catalog PERM.SAMPLE to catalog PERM.FINANCE.
NOTE: Copying entry LOAN.KEYS from catalog PERM.SAMPLE to catalog PERM.FINANCE.
NOTE: Copying entry LOAN.PMENU from catalog PERM.SAMPLE to catalog PERM.FINANCE.
17      copy in=perm.formats out=perm.finance;
18         select revenue.format dept.formatc;
19   quit;
NOTE: Copying entry REVENUE.FORMAT from catalog PERM.FORMATS to catalog
      PERM.FINANCE.
NOTE: Copying entry DEPT.FORMATC from catalog PERM.FORMATS to catalog
      PERM.FINANCE.
```

Example 2: Displaying Contents, Changing Names, and Changing a Description

Procedure features:

PROC CATALOG statement

CHANGE statement options:

  (ENTRYTYPE=) or (ET=)

CONTENTS statement options:

  FILE=

MODIFY statement

RUN statement

QUIT statement

This example

- lists the entries in a catalog and routes the output to a file

- changes entry names
- changes entry descriptions
- processes entries in multiple run groups.

Program

The SAS system option SOURCE writes the source code to the log.

```
options nodate pageno=1 linesize=80 pagesize=60 source;
```

The LIBNAME statement assigns a libref to the SAS data library that contains a permanent SAS catalog. The FILENAME statements assign filerefs to two external files.

```
libname perm 'SAS-data-library';
```

The CONTENTS statement creates a listing of the contents of the SAS catalog PERM.FINANCE and routes the output to a file.

```
proc catalog catalog=perm.finance;
   contents;
title1 'Contents of PERM.FINANCE before changes are made';
run;
```

The CHANGE statement changes the name of an entry that contains a user-written character format. (ET=) specifies the entry type.

```
   change dept=deptcode (et=formatc);
run;
```

The MODIFY statement changes the description of an entry. The CONTENTS statement creates a listing of the contents of PERM.FINANCE after all the changes have been applied. QUIT ends the procedure.

```
   modify loan.frame (description='Loan analysis app. - ver1');
   contents;
title1 'Contents of PERM.FINANCE after changes are made';
run;
quit;
```

Output

Output 6.1

```
                  Contents of PERM.FINANCE before changes are made            1

                          Contents of Catalog PERM.FINANCE

# Name    Type            Create Date       Modified Date Description
--------------------------------------------------------------------------------
1 REVENUE FORMAT   16OCT1996:13:48:11  16OCT1996:13:48:11 FORMAT:MAXLEN=16,16,12
2 DEPT    FORMATC  30OCT1996:13:40:42  30OCT1996:13:40:42 FORMAT:MAXLEN=1,1,14
3 LOAN    FRAME    30OCT1996:13:40:43  30OCT1996:13:40:43 Loan analysis
                                                          application
4 LOAN    HELP     16OCT1996:13:48:10  16OCT1996:13:48:10 Information about
                                                          the application
5 LOAN    KEYS     16OCT1996:13:48:10  16OCT1996:13:48:10 Custom key definitions
                                                          for application
6 LOAN    PMENU    16OCT1996:13:48:10  16OCT1996:13:48:10 Custom menu
                                                          definitions for
                                                          application
7 LOAN    SCL      16OCT1996:13:48:10  16OCT1996:13:48:10 SCL code for loan
                                                          analysis application
```

```
                  Contents of PERM.FINANCE after changes are made             2

                          Contents of Catalog PERM.FINANCE

# Name    Type            Create Date       Modified Date Description
--------------------------------------------------------------------------------
1 REVENUE FORMAT   16OCT1996:13:48:11  16OCT1996:13:48:11 FORMAT:MAXLEN=16,16,12
2 DEPT    FORMATC  30OCT1996:13:40:42  30OCT1996:13:40:42 FORMAT:MAXLEN=1,1,14
3 LOAN    FRAME    30OCT1996:13:40:43  23OCT1998:14:30:40 Loan analysis
                                                          app. - ver1
4 LOAN    HELP     16OCT1996:13:48:10  16OCT1996:13:48:10 Information about
                                                          the application
5 LOAN    KEYS     16OCT1996:13:48:10  16OCT1996:13:48:10 Custom key definitions
                                                          for application
6 LOAN    PMENU    16OCT1996:13:48:10  16OCT1996:13:48:10 Custom menu
                                                          definitions for
                                                          application
7 LOAN    SCL      16OCT1996:13:48:10  16OCT1996:13:48:10 SCL code for loan
                                                          analysis application
```

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS® Procedures Guide, Version 8,* Cary, NC: SAS Institute Inc., 1999. 1729 pp.

SAS® Procedures Guide, Version 8

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

ISBN 1–58025–482–9

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

1st printing, October 1999