SAS® 9.4 DATASETS Procedure Tip Sheet

PROC DATASETS Examples

Example 1: Remove Labels and Formats

Within PROC DATASETS, all labels and formats are removed by using the MODIFY statement and the ATTRIB option. The CONTENTS statement within PROC DATASETS displays the contents of the data set.

```
proc datasets lib=work memtype=data;
   modify class;
     attrib _all_ label=' ';
     attrib _all_ format=;
   run;
   contents data=class;
   run;
quit;
```

Example 2: Manipulate SAS Files

This example 1 deletes the Tension data set and the A2 catalog, 2 changes the name of the A1 catalog (only member in the library that is named A1) to Postdrug, 3 exchanges the names of the Weight and Bodyfat data sets, 4 restricts processing to SAS views and requests views to be moved from the Health library to the Dest1 library, 5 moves the SAS view Spdata, 6 moves the catalogs ETest1 through ETest5, and 7 excludes from the COPY operation all SAS files that begin with the letter D, Oxygen, Vision, and Weight.

```
libname dest1 'SAS-library-1';
libname dest2 'SAS-library-2';
libname health 'SAS-library-3';
proc datasets library=health details;
   delete tension a2(mt=catalog);1
   change a1=postdrug;2
   exchange weight=bodyfat;3
   copy out=dest1 move memtype=view;4
      select spdata;5
      select etest1-etest5/memtype=catalog;6
   copy out=dest2;
      exclude d: oxygen vision weight;7
quit;
```

PROC DATASETS Examples
(Continued)

Example 3: Create ODS Output SAS Data Set

This example creates an ODS output SAS data set that contains directory information about all of the SAS data sets in a SAS library. The output data set can be used for additional processing.

```
libname source 'SAS-library';
ods output members=memout;
proc datasets lib=source memtype=data;
   contents data=_all_ nods;
run;
quit;
proc print data=memout label;
run;
```

Example 4: Modify SAS Data Sets

This example modifies two SAS data sets. For the Subjects data set, the MODIFY statement and subordinate statements 1 label the data set, 2 assign a Read password, 3 specify how the data is sorted, 4 create a composite index on variables Birth and Salary, 5 exclude from the index all observations with missing values, 6 specify that the combination of values of the index variables must be unique, 7 assign an informat and format to the Birth variable, and 8 label the variable Salary. For the Oxygen data set, the MODIFY statement and subordinate statements 9 rename the variable Oxygen to Intake and 10 label the variable Intake.

```
libname health 'SAS-library';
proc datasets library=health nolist;
   modify subjects (label='Test Subjects'1
         read=green2 sortedby=lname)3;
      index create vital=(birth salary)4 /
         nomiss5 unique6;
      informat birth date7.;7
      format birth date7.;7
      label salary='current salary8
         excluding bonus';
   modify oxygen;
      rename oxygen=intake;9
      label intake='Intake Measurement';10
quit;
```

PROC DATASETS Tips

* When you start the procedure, specify the input library in the PROC DATASETS statement. If you omit an input library, the procedure processes the current default SAS library (usually the WORK library).

* Statements execute in the order in which they are written. To see the contents of a data set, copy a data set, and then visually compare the contents of the second data set with the first, the statements that perform those tasks must appear in that order (that is, CONTENTS, COPY, CONTENTS).

* PROC DATASETS supports RUN-group processing, which enables you to submit RUN groups without ending the procedure.

* Generally, if an error occurs in a statement, the RUN group containing the error does not execute. RUN groups preceding or following the one containing the error execute normally. The MODIFY RUN group is an exception. If a syntax error occurs in a statement subordinate to the MODIFY statement, only the statement containing the error fails. The other statements in the RUN group execute.

* To stop the procedure, issue a QUIT statement, a RUN CANCEL statement, a new PROC statement, or a DATA statement. Submitting a QUIT statement executes any statements that have not executed. Submitting a RUN CANCEL statement cancels any statements that have not executed.

For complete information, refer to the SAS 9.4 documentation at support.sas.com/documentation.

SAS Institute Inc. World Headquarters
+1 919 677 8000 www.sas.com/offices

SAS® 9.4 DATASETS Procedure Tip Sheet

This tip sheet places frequently used information in one place, on one sheet of paper, so you don't have to search throughout the documentation. This tip sheet presents SAS 9.4 information for the DATASETS procedure.

The DATASETS procedure is a utility procedure that manages your SAS libraries and helps you perform the following tasks:

- copy SAS files from one SAS library to another
- rename, repair, and delete SAS files
- list SAS files contained in a SAS library
- list attributes of a SAS data set
- manipulate passwords on SAS files
- append SAS data sets
- modify attributes of SAS data sets and variables
- create and delete indexes
- create and manage audit files
- create and delete integrity constraints
- create and manage extended attributes

SAS® 9.4 DATASETS Procedure Tip Sheet

PROC DATASETS Syntax

PROC DATASETS
<ALTER=*alter-password*>
<ENCRYPTKEY=*key-value*>
<FORCE>
<GENNUM=ALL|HIST|REVERT|*integer*>
<KILL>
<LIBRARY=*libref*>
<MEMTYPE=(*mtype(s)*)>
<NODETAILS|DETAILS>
<NOLIST>
<NOPRINT>
<NOWARN>
<PW=*password*>
<READ=*read-password*>;

AGE *current-name related-SAS-file(s)*
</ <ALTER=*alter-password*>
<MEMTYPE=*mtype*> <PW=*password*>>;

APPEND BASE=<*libref.*>*SAS-data-set*
<APPENDVER=V6>
<DATA=<*libref.*>*SAS-data-set*>
<ENCRYPTKEY=*key-value*>
<FORCE>
<GETSORT>
<NOWARN>;

AUDIT *SAS-file* <(<*SAS-password*>
<ENCRYPTKEY=*key-value*>
<GENNUM=*integer*>)>;
INITIATE <AUDIT_ALL=NO|YES>;
LOG <ADMIN_IMAGE=YES|NO>
<BEFORE_IMAGE=YES|NO>
<DATA_IMAGE=YES|NO>
<ERROR_IMAGE=YES|NO>;
RESUME|**SUSPEND**|**TERMINATE**;
USER_VAR *variable(s)*;

CHANGE *old-name=new-name**
</ <ALTER=*alter-password*>
<GENNUM=ALL|*integer*>
<MEMTYPE=*mtype*> <PW=*password*>>;

PROC DATASETS Syntax
(Continued)

CONTENTS
<CENTILES>
<DATA=*SAS-file-specification*>
<DIRECTORY>
<ENCRYPTKEY=*key-value*>
<FMTLEN>
<MEMTYPE=(*mtype(s)*)>
<NODETAILS|DETAILS>
<NODS>
<NOPRINT>
<ORDER=<COLLATE|CASECOLLATE
|IGNORECASE|VARNUM>
<OUT=*SAS-data-set*>
<OUT2=*SAS-data-set*>
<SHORT>
<VARNUM>;

COPY OUT=*libref-1*
<CLONE|NOCLONE>
<CONSTRAINT=NO|YES>
<DATECOPY>
<ENCRYPTKEY=*key-value*>
<FORCE>
<IN=*libref-2*>
<INDEX=YES|NO>
<MEMTYPE=(*mtype(s)*)>
<MOVE <ALTER=<*alter-password*>>
<OVERRIDE=(*ds_option=value**)>;
EXCLUDE *SAS-file(s)* </ MEMTYPE=*mtype*>;
SELECT *SAS-file(s)*
</ <ALTER=*alter-password*>
<ENCRYPTKEY=*key-value*>
<MEMTYPE=*mtype*> <PW=*password*>>;

DELETE *SAS-file(s)*
</ ALTER=*alter-password*>
<ENCRYPTKEY=*key-value*>
<GENNUM=ALL|HIST|REVERT|*integer*>
<MEMTYPE=*mtype*>
<PW=*password*>>;

PROC DATASETS Syntax
(Continued)

EXCHANGE *name=other-name**
</ <ALTER=*alter-password*>
<MEMTYPE=*mtype*> <PW=*password*>>;

MODIFY *SAS-file* <(<ALTER=*alter-password*>
<ENCRYPTKEY=*key-value*>
<GENMAX=*number-of-generations*>
<LABEL='*data-set-label*'| ' '>
<PW=*password*>
<READ=*read-password*>
<SORTEDBY=*sort-information*>
<TYPE=*data-set-type*>
<WRITE=*write-password*>)>
</ <CORRECTENCODING=*encoding-value*>
<DTC=*SAS-date-time*>
<GENNUM=*integer*>
<MEMTYPE=*mtype*>>;

ATTRIB *variable-list(s) attribute-list(s);*
FORMAT *variable* <*format*>*;
IC CREATE <*constraint-name*=> *constraint*
<MESSAGE='*message-string*'
<MSGTYPE=USER>>;
IC DELETE *constraint-name(s)* | _ALL_;
IC REACTIVATE *foreign-key-name*
REFERENCES *libref*;
INDEX CENTILES *index(s)*
</ <REFRESH>
<UPDATECENTILES=
ALWAYS|NEVER|*integer*>>;
INDEX CREATE *index-specification(s)*
</ <NOMISS> <UNIQUE>
<UPDATECENTILES=
ALWAYS|NEVER|*integer*>>;
INDEX DELETE *index(s)* | _ALL_;
INFORMAT *variable* <*informat*>*;
LABEL *variable*=<'*label*' | ' '>*;
RENAME *old-name=new-name**;
XATTR ADD DS *name=value**;
XATTR ADD VAR *variable* (*name=value**)*;
XATTR DELETE;

PROC DATASETS Syntax
(Continued)

XATTR OPTIONS <SEGLEN=*bytes*>;
XATTR REMOVE DS *name(s)*;
XATTR REMOVE VAR *variable* (*names(s)*)*;
XATTR SET DS *name=value**;
XATTR SET VAR *variable* (*name=value**)*;
XATTR UPDATE DS *name=value**;
XATTR UPDATE VAR *variable*
(*name=value**)*;

REBUILD *SAS-file*
</ ALTER=*alter-password*>
<ENCRYPTKEY=*key-value*>
<GENNUM=*integer*>
<MEMTYPE=*mtype*>
<NOINDEX>>;

REPAIR *SAS-file(s)*
</ <ALTER=*alter-password*>
<ENCRYPTKEY=*key-value*>
<GENNUM=*integer*>
<MEMTYPE=*mtype*>
<PW=*password*>>;

SAVE *SAS-file(s)* </ MEMTYPE=*mtype*>;

Syntax Conventions

- An asterisk (*) indicates that the group of arguments can be repeated any number of times.

- Default values are underscored.

- The options ALTER=, ENCRYPTKEY=, MEMTYPE=, GENNUM=, and PW= can be specified in parentheses after the SAS filename or, as shown in the syntax, after a forward slash (/).

- The APPEND, CONTENTS, and COPY statements have corresponding procedures.