

PROC DATASETS

This is general utility procedure is useful for List, Copy, Rename, Delete, Append, Exchange, Repairing SAS member types.

Syntax:-

```
Proc datasets library= libname Memtype = member type list <options>;  
Change oldname=newname memtype=member type;  
Exchange name=othername memtype=member type;  
Copy member_ list memtype=member type;  
Delete member_ list memtype=member type;  
Repair member_ list memtype=member type;  
Modify member_ list memtype=member type;  
Append member_ list memtype=member type;
```

Options:-

Library -

Specifies the SAS Library. . If a library is not specified, SAS looks at the last (_last_) library referenced. If no library has been specified, the default is the work library

```
Proc datasets lib=work;
```

```
Run;
```

```
Quit;
```

Details/Nodetails -

To report descriptive information for required Library.

Details options can be used to report descriptive information for required Library.

```
Proc datasets lib=work Details;
```

```
Run;
```

```
Quit;
```

List/Nolist -

suppresses printing of directory of members. The default is LIST.

```
Proc datasets lib=work Nolist;
```

```
Run;
```

```
Quit;
```

Force -

Used to force the proc to continue even though an error has been found.

```
Proc datasets lib=work Force;
```

```
/*Commands of proc datasets*/
```

```
Run;
```

```
Quit;
```

Memtype = member type -

Specify the member types from the SAS library that need to be processed by proc datasets. Can be used to copy the required sas file type or required memtype. Values of memtypes are All, Data, View and Catalog.

```
Proc datasets Lib=Work Memtype=Data;
```

```
/*Commands of proc datasets*/
```

```
Run;
```

```
Quit;
```

Nowarn -

Suppresses warning messages.

```
Proc datasets Lib=Work Nowarn;
```

```
/*Commands of proc datasets*/
```

```
Run;
```

```
Quit;
```

Kill-

Delete all members in the sas data library.

```
Proc datasets library = work kill;
```

```
Run;
```

```
Quit;
```

When we have large number of SAS datasets and system have space issues then this proc is useful to kill all the datasets from library.

Commands:-

Change -

Rename SAS data sets

```
Data Emp;
```

```
Infile Datalines;
```

```
Input Eid Sal Jod: ddmmyy10.;
```

```
Cards;
```

```
100 2000 30-01-2010
```

```
101 3000 23-02-2009
```

```
103 3500 25-03-2008
```

```
;
```

```
Run;
```

```
Proc datasets Lib=Work;
```

```
Change Emp=Employee;
```

```
Run;
```

```
Quit;
```

Exchange -

The names of datasets will be exchanged for the specified member type

```
Data Emp1;
Infile Datalines;
Input Eid Sal Jod:ddmmyy10.;
Cards;
100 2000 30-01-2010
101 3000 23-02-2009
103 3500 25-03-2008
;
Run;
Data Emp2;
Infile Datalines;
Input Eid Sal Jod:ddmmyy10.;
Cards;
104 5000 30-01-2010
105 2500 23-02-2009
106 4500 25-03-2008
;
Run;
Proc datasets Lib=Work;
Exchange Emp1=Emp2;
Run;
Quit;
```

Copy -

Used to copy the SAS datasets between the libraries

```
Proc Datasets Lib=Sashelp;
Copy In=Sashelp Out=Work;
Run;
Quit;

Proc Datasets Lib=Sashelp;
Copy In=Sashelp Out=Work;
Select Class Shoes;
/*Exclude Class Shoes;*/
Run;
Quit;

Proc Datasets Lib=Sashelp;
Copy In=Sashelp Out=Work Memtype=data;
/*Copy In=Sashelp Out=Work Memtype=view;*/
/*Copy In=Sashelp Out=Work Memtype=catalog;*/
```

```
/*Copy In=Sashelp Out=Work Memtype=all;*/
```

```
Run;
```

```
Quit;
```

If we are not mentioning member type system will take default member type All.

Modify -

Modify statement can be used to change the formats.

```
Proc Datasets Lib=Work;
```

```
Modify Emp;
```

```
Format Doj Worddate20.;
```

```
Run;
```

```
Quit;
```

Working on dataset passwords

```
/* Apply password */
```

```
Proc Datasets Lib=Work;
```

```
Modify Emp(pw=krisna);
```

```
Run;
```

```
Quit;
```

```
/* Alter password */
```

```
Proc Datasets Lib=Work;
```

```
Modify Emp(pw=krisna/stansys);
```

```
Run;
```

```
Quit;
```

```
/* Delete password */
```

```
Proc Datasets Lib=Work;
```

```
Modify Emp(pw=stansys/);
```

```
Run;
```

```
Quit;
```

Rename -

Renaming column names.

```
Proc Datasets Lib=Work;
```

```
Modify Emp;
```

```
Rename Doj=Joindate;
```

```
Run;
```

```
Quit;
```

Appending -

Using append statement we can append the values or load the values from one dataset to another dataset. Append statement can be used only in data set procedure

```
Data Emp1;
```

```
Infile Datalines;
```

```
Input Eid Sal Jod:ddmmyy10.;  
Cards;  
100 2000 30-01-2010  
101 3000 23-02-2009  
103 3500 25-03-2008
```

```
;
```

```
Run;
```

```
Data Emp2;
```

```
Infile Datalines;
```

```
Input Eid Sal Jod:ddmmyy10.;
```

```
Cards;
```

```
104 5000 30-01-2010  
105 2500 23-02-2009  
106 4500 25-03-2008
```

```
;
```

```
Run;
```

```
Proc Datasets lib=work;
```

```
Append Base=Emp1 Data=Emp2;
```

```
Quit;
```

We can use **Force** option in proc datasets also like Proc append.

See proc append for force option.

Contents -

To report descriptive information for required dataset in particular library.

```
Proc datasets Lib=Work;
```

```
Contents Data=Emp;
```

```
Quit;
```

Delete-

Delete required dataset from library.

```
Proc datasets Lib=Work;
```

```
Delete Emp;
```

```
Quit;
```

Repair -

Repairs corrupted SAS datasets.

```
Proc datasets lib=work;
```

```
Repair Emp;
```

```
Run;
```

NOTE: Repairing WORK.EMP (memtype=DATA).

ERROR: Unable to repair WORK.EMP.DATA because it is in a scratch library.

We can repair SAS datasets which we have in permanent library.

Proc datasets lib=Krishna;

Repair Emp;

Run;

Why Proc datasets not used more?

How Proc datasets is efficient than older procedures like contents, copy & append etc... ?

One reason is that three of the older SAS procedures – Copy, Contents, and Append were around before proc datasets. People learned those three and stuck with them. Why use proc datasets instead of several procedures? Proc datasets is more efficient, more versatile, and has features far beyond copy, contents, and append. Using Proc datasets lets you remember, become proficient, and rely on one proc rather than waste brain storage and time working with three.

PROC DATASETS reads and changes information about SAS datasets without actually changing the data. This makes PROC DATASETS much more efficient than the data step which must read data in, whether or not it will be changed, in order to make changes to the descriptor information.

PROC DATASETS allows run group processing. Meaning you can start a procedure and have the commands execute without ending it. Commands can be stacked up and executed together. An error in one command stops processing for all remaining statements. You can use the FORCE option to force the proc to continue, but my advice is to not use FORCE. I would rather find my error, correct it, and, rerun the entire package. Then I know what to expect in subsequent steps and procedures. The FORCE command

does have other, very powerful, uses as we will see.