CHAPTER

35

File Protection

Definitions

SAS software enables you to restrict access to members of SAS data libraries by assigning passwords to them. You can assign passwords to all member types except catalogs. You can specify three levels of protection: read, write, and alter. When a password is assigned, it appears as uppercase Xs in the log.

Note: This document uses the terms *SAS data file* and *SAS data view* to distinguish between the two types of SAS data sets. Passwords work differently for type VIEW than they do for type DATA. The term "SAS data set" is used when the distinction is not necessary. △

read	protects against reading the file.
write	protects against changing the data in the file. For SAS data files, write protection prevents adding, modifying, or deleting observations.
alter	protects against deleting or replacing the entire file. For SAS data files, alter protection also prevents modifying variable attributes and creating or deleting indexes.

Alter protection does not require a password for read or write access; write protection does not require a password for read access. For example, you can read an alter-protected or write-protected SAS data file without knowing the alter or write password. Conversely, read and write protection do not prevent any operation that requires alter protection. For example, you can delete a SAS data set that is only read- or write-protected without knowing the read or write password.

To protect a file from being read, written to, deleted or replaced by anyone who does not have the proper authority, assign read, write and alter protection. To allow others to read the file without knowing the password, but not change its data or delete it, assign just write and alter protection. To completely protect a file with one password, use the PW= data set option. See "Assigning Complete Protection with the PW= Data Set Option" on page 507 for details.

Note: Because of the way SAS opens files, you must specify the read password to update a SAS data set that is only read-protected. △

Note: The levels of protection differ somewhat for the member type VIEW. See "Using Passwords with Views" on page 508. △

Assigning Passwords

Syntax

To set a password, first specify a SAS data set in one of the following:

- a DATA statement
- the MODIFY statement of the DATASETS procedure
- an OUT = statement in PROC SQL
- the CREATE VIEW statement in PROC SQL
- the ToolBox.

Then assign one or more password types to the data set. The data set may already exist, or the data set may be one that you create. An example of syntax follows:

password-type=password <... password-type=password>)

where *password* is a valid eight-character SAS name and *password-type* can be one of the following SAS data set options:

ALTER=

PW=

READ=

WRITE=

CAUTION:

Keep a record of any passwords you assign! If you forget or do not know the password, you cannot get the password from SAS. △

Assigning a Password with a DATA Step

You can use data set options to assign passwords to unprotected members in the DATA step when you create a new SAS file.

This example prevents deletion or modification of the data set without a password.

```
   /* assign a write and an alter password to MYLIBNAME.STUDENTS */
data mylibname.students(write=yellow alter=red);
   input name $ sex $ age;
   datalines;
Amy f 25
… more data lines …
;
```

This example prevents reading or deleting a stored program without a password and also prevents changing the source program.

```
   /* assign a read and an alter password to the view ROSTER */
data mylibname.roster(read=green alter=red) /
     view=mylibname.roster;
   set mylibname.students;
run;

.

libname stored 'SAS-data-library-2';

   /* assign a read and alter password to the program file SOURCE */
data mylibname.schedule / pgm=stored.source(read=green alter=red);
   … DATA step statements …
run;
```

Assigning a Password to an Existing Data Set

You can use the MODIFY statement in the DATASET procedure to assign passwords to unprotected members if the SAS data file already exists.

```
   /* assign an alter password to STUDENTS */
proc datasets library=mylibname;
   modify students(alter=red);
run;
```

Assigning a Password with a Procedure

You can assign a password after an OUT= data set specification in PROC SQL.

```
   /* assign a write and an alter password to SCORE */
proc sort data=mylibname.math
   out=mylibname.score(write=yellow alter=red);
   by number;
run;
```

You can use a CREATE VIEW statement in PROC SQL to assign a password.

```
   /* assign an alter password to the view BDAY */
proc sql;
   create view mylibname.bday(alter=red) as
      query-expression;
```

Assigning a Password with the SAS Windowing Environment

You can create or change passwords for any data file using the Password Widow in the SAS windowing environment. To invoke the Password Window from the ToolBox, use the global command SETPASSWORD followed by the file name. This opens the password window for the specified data file.

Removing or Changing Passwords

To remove or change a password, use the MODIFY statement in the DATASETS procedure. See the *SAS Procedures Guide* for more information on PROC DATASETS.

Using Password-Protected SAS Files in DATA and PROC Steps

To access password-protected files, use the same data set options that you use to assign protection.

□

```
   /* Assign a read and alter password
   /* to the stored program file*/ /*STORED.SOURCE */
   data mylibname.schedule / pgm=stored.source
      (read=green alter=red);     <… more data step statements …>
run;
```

```
   /*Access password-protected file*/
proc sort data=mylibname.score(write=yellow alter=red);
   by number;
run;
```

□

```
   /* Print read-protected data set MYLIBNAME.AUTOS */
      proc print data=mylibname.autos(read=green); run;
```

□

```
   /* Append ANIMALS to the write-protected */
   /* data set ZOO */
proc append base=mylibname.zoo(write=yellow)
            data=mylibname.animals;
run;
```

□

```
   /* Delete alter-protected data set MYLIBNAME.BOTANY */
proc datasets library=mylibname;
   delete botany(alter=red);
run;
```

Passwords are hierarchical in terms of gaining access. For example, specifying the ALTER password gives you read and write access. The following example creates the data set STATES, with three different passwords, and then reads the data set to produce a plot:

```
data mylibname.states(read=green write=yellow alter=red);
   input density crime name $;
   datalines;
151.4 6451.3 Colorado
… more data lines …
;

proc plot data=mylibname.states(alter=red);
   plot crime*density;
run;
```

How SAS Handles Incorrect Passwords

If you are using the SAS windowing environment and you try to access a password-protected member without specifying the correct password, you receive a requestor window that prompts you for the appropriate password. The text you enter in this window is not displayed. You can use the PWREQ= data set option to control whether a requestor window appears after a user enters a missing or incorrect password. PWREQ= is most useful in SCL applications.

If you are using batch or noninteractive mode, you receive an error message in the SAS log if you try to access a password-protected member without specifying the correct password.

If you are using interactive line mode, you are also prompted for the password if you do not specify the correct password. When you enter the password and press ENTER, processing continues. If you cannot give the correct password, you receive an error message in the SAS log.

Assigning Complete Protection with the PW= Data Set Option

The PW= data set option assigns the same password for each level of protection. This data set option is convenient for thoroughly protecting a member with just one password. If you use the PW= data set option, those who have access only need to remember one password for total access.

- To access a member whose password is assigned using the PW= data set option, use the PW= data set option or the data set option that equates to the specific level of access you need:

```
   /* create a data set using PW=,
      then use READ= to print the data set */
data mylibname.states(pw=orange);
   input density crime name $;
   datalines;
151.4 6451.3 Colorado
… more data lines …
;

proc print data=mylibname.states(read=orange);
run;
```

- PW= can be an alias for other password options:

```
   /* Use PW= as an alias for ALTER=. */
data mylibname.college(alter=red);
   input name $ 1-10 location $ 12-25;
   datalines;
Vanderbilt Nashville
Rice       Houston
Duke       Durham
Tulane     New Orleans
… more data lines …
;

proc datasets library=mylibname;
   delete college(pw=red);
run;
```

Using Passwords with Views

How the Level of Protection Differs from SAS Views

The levels of protection for views and stored programs differ slightly from other types of SAS files. Passwords affect the actual view definition or view descriptor as well as the underlying data. Unless otherwise noted, the term "view" can refer to any type of view. Also, the term "underlying data" refers to the data accessed by the view:

read	□ protects against reading the view's underlying data. □ allows source statements to be written to the SAS log, using DESCRIBE. □ allows replacement of the view.
write	does not protect underlying data associated with a view.
alter	□ protects against reading the view's underlying data. □ protects against source statements being written to the SAS log, using DESCRIBE. □ protects against replacement of the view.

A key difference between views and other types of SAS files is that you need alter access to read (browse) an alter-protected view. For example, to use an alter-protected PROC SQL view in a DESCRIBE VIEW statement, you must specify the alter password.

In most DATA and PROC steps, the way you use password-protected views is consistent with the way you use other types of password-protected SAS files. For example, the following PROC PRINT step prints a read-protected view:

```
proc print data=mylibname.grade(read=green);
run;
```

PROC SQL Views

Typically, when you create a PROC SQL view from a password-protected SAS data set, you specify the password in the FROM clause in the CREATE VIEW statement using a data set option. In this way, when you use the view later, you can access the underlying data without re-specifying the password. For example, the following

statements create a PROC SQL view from a read-protected SAS data set, and drop a sensitive variable:

```
proc sql;
   create view mylibname.emp as
      select * from mylibname.employee(pw=orange drop=salary);
quit;
```

Note: If you create a PROC SQL view from password-protected SAS data sets without specifying their passwords, when you try to use the view you are prompted for the passwords of the SAS data sets named in the FROM clause. If you are running SAS in batch or noninteractive mode, you receive an error message. △

SAS/ACCESS Views

SAS/ACCESS software enables you to edit Version 6 view descriptors and, in some interfaces, the underlying data. To prevent someone from editing or reading (browsing) the view descriptor, assign alter protection to the view. To prevent someone from updating the underlying data, assign write protection to the view. For more information, see the SAS/ACCESS documentation for your DBMS.

DATA Step Views

When you create a DATA step view using a password-protected SAS data set, specify the password in the view definition. In this way, when you use the view, you can access the underlying data without respecifying the password.

The following statements create a DATA step view using a password-protected SAS data set, and drop a sensitive variable:

```
data mylibname.emp / view=mylibname.emp;
   set mylibname.employee(pw=orange drop=salary);
run;
```

Note that you can use the view without a password, but access to the underlying data requires a password. This is one way to protect a particular column of data. In the above example, **`proc print data=mylibname.emp;`** will execute, but **`proc print data=mylibname.employee;`** will fail without the password.

SAS Data File Encryption

SAS passwords restrict access to SAS data files within SAS, but SAS passwords cannot prevent SAS data files from being viewed at the operating environment system level or from being read by an external program.

Encryption provides security of your SAS data outside the SAS System by writing to disk the encrypted data that represents the SAS data. The data is decrypted as it is read from the disk.

Encryption does not affect file access. However, SAS honors all host security mechanisms that control file access. You can use encryption and host security mechanisms together.

Encryption is implemented with the ENCRYPT= data set option. You can use the ENCRYPT= data set option only when you are creating a SAS data file. You must also assign a password when encrypting a file. At a minimum, you must specify the READ=

or the PW= data set option at the same time you specify ENCRYPT=YES. Because passwords are used in the encryption method, you cannot change *any* password on an encrypted data set without re-creating the data set.

The following rules apply to data file encryption:

- In order to copy an encrypted SAS data file, the output engine must support encryption. Otherwise, the data file is not copied.
- Previous releases of SAS cannot use an encrypted SAS data file. Encrypted files work only in Release 6.11 or in later releases of SAS.
- You cannot encrypt SAS data views because they contain no data.
- If the data file is encrypted, all associated indexes are also encrypted.
- Encryption requires roughly the same amount of CPU resources as compression.
- You cannot use PROC CPORT on encrypted SAS data files.

Example

This example creates an encrypted SAS data set:

```
data salary(encrypt=yes read=green);
   input name $ yrsal bonuspct;
   datalines;
Muriel    34567  3.2
Bjorn     74644  2.5
Freda     38755  4.1
Benny     29855  3.5
Agnetha   70998  4.1
;
```

To print this data set, specify the read password:

```
proc print data=salary(read=green);
run;
```

Passwords and Encryption with Generation Data Sets, Audit Trails, Indexes and Copies

SAS extends password protection and encryption to other files associated with the original protected file. This includes generation data sets, indexes, audit trails and copies. When accessing protected or encrypted generation data sets, indexes audit trails and copies of the original file, the same rules, syntax and behavior for invoking the original password protected or encrypted files apply. Data views can not have generation data sets, indexes and audit trails.

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

ISBN 1–58025–441–1

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

1st printing, November 1999