



CHAPTER

3

Defining SAS/ACCESS Descriptor Files

Introduction 19

Understanding SAS/ACCESS Descriptor Files 19

Creating Descriptor Files 20

Creating Access and View Descriptors in One PROC Step 20

Creating Access and View Descriptors in Separate PROC Steps 22

Introduction

To use the SAS/ACCESS interface to SYSTEM 2000 software, you must define special files that describe the SYSTEM 2000 database and data to the SAS System. These files are called *SAS/ACCESS descriptor files*. This chapter is a tutorial and uses examples to illustrate creating these files. (For complete reference information on the ACCESS procedure, see Chapter 7, “ACCESS Procedure Reference,” on page 73.)

The examples are based on the SYSTEM 2000 database named EMPLOYEE. (See Appendix 3, “Example Data,” on page 143 to review the definition of the EMPLOYEE database and see the data accessed by the examples.) You create an access descriptor file named MYLIB.EMPLOYEE for that database. Then, you create two view descriptor files named VLIB.EMPPOS and VLIB.EMPSKIL based on the access descriptor. Before you create these files, though, the next sections provide some background on access and view descriptor files.

Understanding SAS/ACCESS Descriptor Files

The SAS System interacts with SYSTEM 2000 software through an interface view engine that uses SAS/ACCESS descriptor files. There are two types of descriptor files:

- access descriptor files (member type ACCESS)
- view descriptor files (member type VIEW)

An *access descriptor* contains information about the SYSTEM 2000 database you want to use. The information includes the database name, the item names, and their item types. You use the access descriptor to create view descriptors. Think of an access descriptor as being a *master* descriptor file for a single SYSTEM 2000 database because it usually contains a complete description of that database.

A view descriptor defines a subset of the data described by an access descriptor. You choose this subset by selecting particular items, and you can specify selection criteria that the data must meet. For example, you may want to select two items, LAST NAME and CITY-STATE, and specify that the value stored in item CITY-STATE must be AUSTIN TX. You can also specify a sequence order for the data.

After you create your view descriptor, you can use it in a SAS program to read data directly from the SYSTEM 2000 database or to extract the data and place them in a SAS data file. Typically, for each access descriptor that you define, you have several view descriptors, each selecting different subsets of data.

Creating Descriptor Files

You define descriptor files with the ACCESS procedure. You can define access descriptor files and view descriptor files in the same PROC ACCESS execution or in separate executions. Within an execution, you can define multiple descriptor files of the same and different types.

Creating Access and View Descriptors in One PROC Step

Perhaps the most common way to use the ACCESS procedure statements, especially in batch mode, is to create an access descriptor and one or more view descriptors based on this access descriptor in a single PROC ACCESS execution. The following example shows how to do this. First, an access descriptor is created (MYLIB.EMPLOYEE). Then two view descriptors are created (VLIB.EMPPOS and VLIB.EMPSKIL). Each statement is then explained in the order it appears in the example program.

```
proc access dbms=s2k;
  create mylib.employe.access;
    database=employee s2kpw=demo mode=multi;
    assign=yes;
    drop c110 c120;
    rename forename=firstnme office_e=phone
           yearsofe=years gender=sex
           degree_c=degree;
    length firstnme=13 lastname=13 c101=16;
    list all;

  create vlib.emppos.view;
    select lastname firstnme position departme
           manager;
    subset "order by lastname";
    list all;

  create vlib.empskil.view;
    select c2 c3 c201 c203;
    subset "ob skilltyp";
    s2kpw=demo mode=multi;
    list view;
run;
```

Here is an explanation of the statements in this example:

proc access dbms=s2k;
invokes the ACCESS procedure for the SAS/ACCESS interface to SYSTEM 2000 software.

create mylib.employe.access;

identifies the access descriptor, MYLIB.EMPLOYE, that you want to create. The MYLIB libref must be associated with the SAS data library before you can specify it in this statement.

database=employee s2kpw=demo mode=multi;

indicates the access descriptor is for the EMPLOYEE database, specifies the password, DEMO, required to access the database definition, and indicates the database is in the Multi-User environment.

assign=yes;

generates unique SAS variable names based on the first eight non-blank characters of the item names. Variable names and attributes can be changed in this access descriptor but not in any view descriptors created from this access descriptor.

drop c110 c120;

marks the records associated with C-numbers C110 and C120 as non-display. Because these two C-numbers indicate records, all the items in each record are marked as non-display. Therefore, all the items in the two records associated with these numbers do not appear in any view descriptor created from this access descriptor.

rename forename=firstnme office_e=phone yearsofe=years gender=sex degree_c=degree;

renames the default SAS variable names associated with the FORENAME, OFFICE_E, YEARSOFE, GENDER, and DEGREE_C SAS names. Specify the default SAS variable name on the left side of the equal sign (=) and the new name on the right. Because the ASSIGN=YES statement is specified, any view descriptors created from this access descriptor automatically use the new SAS variable names.

length firstnme=13 lastname=13 c101=16;

changes the field width for the items associated with FIRSTNME and LASTNAME to 13 characters and the field width for the item associated with C-number C101 (the POSITION SAS name) to 16 characters.

list all;

lists the access descriptor's item identifier numbers, C-numbers, SAS variable names, SAS formats, SAS informats, and SAS variable lengths. The list includes any associated BY key information and is written to the SAS log. Any items that have been dropped from display (using the DROP statement) have *NON-DISPLAY* next to them.

create vlib.emppos.view;

writes the access descriptor to the library associated with MYLIB and identifies the view descriptor, VLIB.EMPPOS, that you want to create. The VLIB libref must be associated with a SAS data library before you can specify it in this statement.

select lastname firstnme position departme manager;

selects the items associated with the LASTNAME, FIRSTNME, POSITION, DEPARTME, and MANAGER SAS names for inclusion in the view descriptor. The SELECT statement is required to create the view unless a RENAME, FORMAT, INFORMAT, LENGTH, or BYKEY statement is specified.

subset "order by lastname";

specifies you want the names in the output to be ordered by last name. Using the word WHERE is optional. Use the SAS/ACCESS interface to SYSTEM 2000 syntax in the SUBSET statement.

list all;

lists all the available item identifier numbers, C-numbers, SAS variable names, SAS formats, SAS informats, and SAS variable lengths on which the view descriptor is based; items that have been dropped from the display have *NON-DISPLAY* next to them. Any associated BY key information is also listed. Selection criteria specified in the view descriptor are listed. Items that have been selected for the view have *SELECTED* next to them. The list is written to the SAS log.

create vlib.empskil.view;

writes the first view descriptor to the library associated with VLIB and identifies the next view descriptor, VLIB.EMPSKIL, that you want to create.

select c2 c3 c201 c203;

selects the four items associated with C-numbers C2, C3, C201 and C203 for inclusion in the view descriptor. The SELECT statement is required to create the view unless a RENAME, FORMAT, INFORMAT, LENGTH, or BYKEY statement is specified.

subset "ob skilltyp";

specifies you want the observations to be sorted by skill type. Refer to "SUBSET" on page 87 for syntax information.

s2kpw=demo mode=multi;

specifies the password required to access the data and indicates the database is in the Multi-User environment. This information is stored in the view descriptor. To override this password, or to specify a SYSTEM 2000 password for view descriptor VLIB.EMPPPOS, which omits the S2KPW statement, you can use the S2KPW data set option. For more information, see "Data Set Options" on page 128.

list view;

lists the item identifier numbers, the C-numbers, the SAS variable names, the SAS formats, the SAS informats, and the SAS variable lengths that have been selected for the view descriptor. Any associated BY key information is also listed. Selection criteria specified in the view descriptor are listed. The list is written to the SAS log.

run;

writes the last view descriptor when the RUN statement is processed.

Creating Access and View Descriptors in Separate PROC Steps

Examples of how to create the MYLIB.EMPLOYEE access descriptor and VLIB.EMPOS and VLIB.EMPSKIL view descriptors in separate PROC ACCESS executions are provided in Appendix 3, "Example Data," on page 143.

When you use a separate PROC ACCESS execution to create a view descriptor, note that you must use the ACCDESC= option to specify an existing access descriptor from which the view descriptor will be derived.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/ACCESS® Interface to SYSTEM 2000® Data Management Software: Reference, Version 8*, Cary, NC: SAS Institute Inc., 1999.

**SAS/ACCESS® Interface to SYSTEM 2000® Data Management Software:
Reference, Version 8**

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

ISBN 1-58025-549-3

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, October 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.

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.