CHAPTER

3

Defining SAS/ACCESS Descriptor Files

Introduction

To use the SAS/ACCESS to IMS-DL/I interface view engine, you must define special files that describe IMS-DL/I databases and data to the SAS System. These files are called SAS/ACCESS descriptor files. This chapter is a tutorial and uses examples to show you how to create and edit descriptor files.

The examples in this chapter are based on the IMS-DL/I database WIRETRN. Complete information on the WIRETRN database is provided later in this chapter. From this database, you will create an access descriptor. Then, you will create a view descriptor based on the access descriptor. For complete reference information on the ACCESS procedure, see Chapter 6, "ACCESS Procedure Reference," on page 93.

Understanding SAS/ACCESS Descriptor Files

One way that the SAS System interacts with IMS-DL/I databases is through an interface view engine that makes use of SAS/ACCESS descriptor files created with the ACCESS procedure. There are two types of descriptor files:

- access descriptors
- view descriptors.

An *access descriptor* contains information about the IMS-DL/I database you want to use. The information includes the IMS-DL/I database name, the IMS-DL/I field names and their default SAS formats, database formats, segment names and lengths, and key fields. An access descriptor also contains any special handling considerations for a field and indicates if an item occurs multiple times in a database segment. You use the access descriptor to create view descriptors. An access descriptor is like a master descriptor file for a single IMS-DL/I database because it contains a complete description of that database (if you choose to enter all the data). Because IMS-DL/I does not store descriptive information about a database, you must enter the database definition in the access descriptor.

A *view descriptor* defines a subset of the data described by an access descriptor. *This subset must contain data from only one path in the database on which the access descriptor is based.* You choose this subset by selecting particular items and specifying criteria that the data must meet. For example, you may want to select two items, CUSTOMER_NAME and STATE, and specify that the value stored in item STATE must equal **NC**.

A view descriptor is a SAS data set of member type VIEW. After you create your view descriptors, you can use them in a SAS program to read or write the data directly from and to an IMS-DL/I database, or you can extract IMS-DL/I data and place them in a SAS data file. Typically, you have several view descriptors (each selecting a different path of data in the database) for each access descriptor that you have defined.

Creating and Using Descriptor Files

You can use batch mode to create the access descriptor MYLIB.WIRETRN and the view descriptor VLIB.WIREDATA. Because IMS-DL/I does not have a dictionary or store descriptive information about IMS-DL/I databases, you must provide the database definition in the SAS statements following the procedure statement. You can also create view descriptors in the same PROC ACCESS execution after the access descriptor statements are entered. (See Chapter 6, "ACCESS Procedure Reference," on page 93 for a list of valid options that you can use with PROC ACCESS.) Here is the general format for creating descriptors:

```
proc access options;
   statements;
   run;
```

Creating Access and View Descriptors in One PROC Step

Perhaps the most common way to use the ACCESS procedure is to create an access descriptor and one or more view descriptors during a single PROC ACCESS execution.

The following example shows how to create the access descriptor MYLIB.WIRETRN based on the IMS-DL/I database WIRETRN. The view descriptor VLIB.WIREDATA is based on this access descriptor. After the following example, each SAS/ACCESS statement is explained in the order of appearance in the program:

```
JCL statements;

libname mylib 'access-descriptor libref';
libname vlib 'view-descriptor libref';

  proc access dbms=ims;
    create mylib.wiretrn.access;
      database=wiretrn dbtype=hdam;
      record='wire transaction' segment=wiretran
                                seglng=100;
        item='ssn - account' level=2 dbformat=$23.
                             search=ssnacc key=y;
        item='account type'  level=2 dbformat=$1.
                             search=accttype;
        item='wire date'     level=2 dbformat=$8.
                             search=wiredate;
        item='wire time'     level=2 dbformat=$8.
                             search=wiretime;
        item='wire amount'   level=2 dbformat=pd5.2
                             search=wireammt
                             dbcontent=1;
        item='wire descript' level=2 dbformat=$40.
                             search=wiredesc;
    an=y;
    list all;

    create vlib.wiredata.view psbname=acctsam
      pcbindex=5;
      select 'wire transaction';
    list view;
  run;
```

Here is an explanation of the statements in this example. See "Procedure Statements" on page 98 for complete reference information on these statements.

JCL statements;
: include for batch and noninteractive line modes.

`libname mylib='libref.access-descriptor';`
`libname vlib='libref.view-descriptor';`
: use LIBNAME statements to reference the SAS data library in which you will store the access descriptor (MYLIB) and the SAS data library in which you will store the view descriptors (VLIB). You must associate a libref with its data library before you can use it in another SAS statement or procedure.

`proc access dbms=ims;`
: invokes the ACCESS procedure for the SAS/ACCESS interface to IMS-DL/I.

`create mylib.wiretrn.access;`
: identifies the access descriptor, MYLIB.WIRETRN, that you want to create.

`database=wiretrn dbtype=hdam;`
: specifies the IMS-DL/I database named WIRETRN on which the access descriptor is to be created. The database type is HDAM.

`record='wire transaction' segment=wiretran seglng=100;`
: specifies the user-specified record name, as well as the segment name and segment length, as specified in the IMS DBD for the WIRETRN database.

`item='ssn - account' level=2 dbformat=$23.  search=ssnacc key=y;`
: identifies the item SSN - ACCOUNT. It has an internal format of type character, length 23 bytes. SSNACC is specified as a search field name. KEY=Y indicates that SSN - ACCOUNT is listed in the DBD as a key field for the WIRETRAN segment.

`item='account type' level=2 dbformat=$1.  search=accttype;`
: identifies the item ACCOUNT TYPE with an internal format of type character, length 1 byte. ACCTTYPE is specified as a search field in the DBD.

`item='wire date' level=2 dbformat=$8.  search=wiredate;`
: identifies the item WIRE DATE with an internal format of type character, length 8 bytes. The search field WIREDATE is specified.

`item='wire time' level=2 dbformat=$8.  search=wiretime;`
: identifies the item WIRE TIME with the same attributes as WIRE DATE except it has the search field name WIRETIME.

`item='wire amount' level=2 dbformat=pd5.2 search=wireammt dbcontent=l;`
identifies item WIRE AMOUNT with a packed decimal database format of 5 bytes with 2 decimal places. DBCONTENT=L indicates that SAS should display a missing value when it finds low values (hexadecimal zeros) for this item. The search field is WIREAMMT.

`item='wire descript' level=2 dbformat=$40.  search=wiredesc;`
identifies the item WIRE DESCRIPT with an internal format of type character, length 40 bytes. The search field is WIREDESC.

`an=y;`
generates unique SAS variable names and default formats based on the name of the IMS-DL/I item and its DBFORMAT= value. Using AN=Y in an access descriptor means no changes can be made to the SAS names and formats in any view descriptors that use the access descriptor.

`list all;`
lists all the items in the access descriptor and SAS information for each item. The output is displayed in the SAS log.

`create vlib.wiredata.view psbname=acctsam pcbindex=5;`
creates a view descriptor called WIREDATA which references PSB ACCTSAM. The PCBINDEX=5 statement refers to the specific PCB in the PSB to be used at execution time.

`select 'wire transaction';`
selects the WIRETRAN segment of the IMS-DL/I database to be included in the view, as defined in the access descriptor.

`list view;`
lists SAS information on the record WIRE TRANSACTION you selected for this view. Output from this statement is shown in the SAS log.

`run;`
forces execution of the ACCESS procedure.

Creating Access and View Descriptors in Separate PROC Steps

You can create view descriptors and access descriptors in separate PROC ACCESS steps. In the first PROC ACCESS step in the following example, you create the access descriptor MYLIB.WIRETRN, which is based on the WIRETRN database. In the second PROC ACCESS step, you create a view descriptor, VLIB.WIREDATA, which is based on the access descriptor MYLIB.WIRETRN.

```
proc access dbms=ims;
    create mylib.wiretrn.access;
      database=wiretrn dbtype=hdam;
      record='wire transaction' segment=wiretran
                                seglng=100;
        item='ssn - account' level=2 dbformat=$23.
                             search=ssnacc
                             key=y;
        item='account type'  level=2 dbformat=$1.
                             search=accttype;
        item='wire date'     level=2 dbformat=$8.
                             search=wiredate;
        item='wire time'     level=2 dbformat=$8.
                             search=wiretime;
        item='wire amount'   level=2 dbformat=pd5.2
                             search=wireammt
                             dbcontent=1;
        item='wire descript' level=2 dbformat=$40.
                             search=wiredesc;
    an=y;
    list all;
  run;

  proc access dbms=ims accdesc=mylib.wiretrn;
    create vlib.wiredata.view psbname=acctsam
        pcbindex=5;
      select 'wire transaction';
    list view;
  run;
```

Note that the statement **`proc access dbms=ims`** is repeated in this example. See "Creating Access and View Descriptors in One PROC Step" on page 40 for complete reference information on this statement.

Using View Descriptors in SAS Programs

You can use a view descriptor in any SAS procedure in which you could use other SAS data sets. The next two examples include printing and reviewing variables for a view descriptor.

Printing Data

Printing IMS-DL/I data described by a view descriptor is like printing any other SAS data set, as shown in the following example:

```
options nodate linesize=120;

proc print data=vlib.wiredata;
   title2 'Wire Transactions';
run;
```

Output 3.1 on page 44 shows the output for the VLIB.WIREDATA view descriptor.

Output 3.1 Results of the PRINT Procedure

```
                                         The SAS System
                                       Wire Transactions

 OBS  SSN_ACCOUNT             ACCOUNT_TYPE  WIRE_DATE WIRE_TIME   WIRE_AMOUNT  WIRE_DESCRIPT

   1  335-45-3451345620145345    C          03/31/95  15:42:43    1563.23      BAD CUST_SSN
   2  434-62-1224345656336366    L          03/30/95  23:45:32    424.87       WIRED FROM SCNB 37262849393
   3  156-45-5672345689435776    S          04/06/95  12:23:42    -150.00      WIRED TO BOA 9383627274
   4  456-45-3462345620134522    C          04/06/95  13:12:34    -245.73      WIRED TO WELLS FARGO CHICAGO
   5  234-74-4612345689413263    S          04/06/95  15:45:42    -238.73      WIRED TO WELLS FARGO SAN FRANCISCO
   6  667-73-8275345620154633    S          03/31/95  15:42:43    1563.23      BAD ACCT_NUM
   7  234-74-4612345620113263    C          04/06/95  11:12:42    1175.00      WIRED FROM SCNB 73653728343
   8  156-45-5672345620123456    C          04/06/94  10:23:53    -136.29      WIRED TO SCNB 53472019836
   9  156-45-5672345620123456    C          04/06/95   9:35:53    1923.87      WIRED FROM CIBN 37284839328
  10  434-62-1224345620134564    C          04/06/95  13:23:52    -284.42      WIRED TO TVNB 837362636438
  11  667-73-8275345689454633    C          03/28/95  15:42:43    1563.23      BAD ACCT_NUM
```

When you use the PRINT procedure, you may want to take advantage of the OBS= and FIRSTOBS= data set options. The OBS= option enables you to specify the last observation to be processed; the FIRSTOBS= option enables you to specify the first. The options are not valid with any form of the WHERE expression. The OBS= option improves performance when the view descriptor describes a large amount of data and you just want to see an example of the output. Because each record must still be read and its position calculated, using the FIRSTOBS= option does not improve performance significantly. The POINT= and KEY= options of the MODIFY and SET statements are not currently supported by the IMS-DL/I engine.

The following example uses the OBS= data set option to print the first five observations of data described by the view descriptor VLIB.WIREDATA, which describes the WIRETRAN segment of the IMS-DL/I database WIRETRN:

```
options nodate linesize=120;

proc print data=vlib.wiredata(obs=5);
   title2 'First Five Observations Described by
          VLIB.WIREDATA';
run;
```

Output 3.2 on page 45 shows the result of this example.

Output 3.2 Results of Using the FIRSTOBS= Option

```
                                              The SAS System
                             First Five Observations Described by VLIB.WIREDATA

OBS  SSN_ACCOUNT              ACCOUNT_TYPE  WIRE_DATE  WIRE_TIME   WIRE_AMOUNT  WIRE_DESCRIPT

  1  335-45-3451345620145345    C           03/31/95   15:42:43    1563.23      BAD CUST_SSN
  2  434-62-1224345656336366    L           03/30/95   23:45:32     424.87      WIRED FROM SCNB 37262849393
  3  156-45-5672345689435776    S           04/06/95   12:23:42    -150.00      WIRED TO BOA 9383627274
  4  456-45-3462345620134522    C           04/06/95   13:12:34    -245.73      WIRED TO WELLS FARGO CHICAGO
  5  234-74-4612345689413263    S           04/06/95   15:45:42    -238.73      WIRED TO WELLS FARGO SAN FRANCISCO
```

For more information on the PRINT procedure, see *SAS Language Reference: Concepts* and *SAS Procedures Guide*. For more information on the OBS= and FIRSTOBS= options, see *SAS Language Reference: Dictionary*.

Reviewing Variables

If you want to use IMS-DL/I data described by a view descriptor in your SAS program, you can use the CONTENTS or DATASETS procedure to display the view's variable and format information. You use these procedures with view descriptors in the same way you use them with other SAS data sets.

The following example uses the DATASETS procedure to give you information on the view descriptor VLIB.WIREDATA, which describes the data in the WIRETRAN segment of the IMS-DL/I database WIRETRN:

```
options nodate linesize=132;

proc datasets library=vlib memtype=view;
   contents data=wiredata;
   title2 ' ';
run;
```

Output 3.3 on page 46 shows the first display of the information for this example.

Output 3.3 Using the DATASETS Procedure with a View Descriptor

```
                              DATASETS PROCEDURE

         Data Set Name: VLIB.WIREDATA          Observations:         .
         Member Type:   VIEW                   Variables:            6
         Engine:        SASIOIMS               Indexes:              0
         Created:       .                      Observation Length:   88
         Last Modified: .                      Deleted Observations: 0
         Protection:                           Compressed:           NO
         Data Set Type:                        Sorted:               NO
         Label:

                  -----Engine/Host Dependent Information-----


              -----Alphabetic List of Variables and Attributes-----

   #    Variable       Type    Len    Pos    Format    Informat    Label
   -------------------------------------------------------------------------
   2    ACCOUNT_TYPE   Char      1     23    $1.       $1.         ACCOUNT TYPE
   1    SSN_ACCOUNT    Char     23      0    $23.      $23.        SSN - ACCOUNT
   5    WIRE_AMOUNT    Num       8     40    12.2      12.2        WIRE AMOUNT
   3    WIRE_DATE      Char      8     24    $8.       $8.         WIRE DATE
   6    WIRE_DESCRIPT  Char     40     48    $40.      $40.        WIRE DESCRIPT
   4    WIRE_TIME      Char      8     32    $8.       $8.         WIRE TIME
```

As you can see from the output produced by the DATASETS procedure, the VLIB.WIREDATA view descriptor has six variables: ACCOUNT_TYPE, SSN_ACCOUNT, WIRE_AMOUNT, WIRE_DATE, WIRE_DESCRIPT, and WIRE_TIME. The variables are listed in alphabetic order, and the column labeled with a # (pound sign) in the listing shows the order of each variable as it appears in the WIRETRAN database segment. You cannot change a view descriptor's variable labels using the DATASETS procedure. The labels are generated from the IMS-DL/I item names when the view descriptor is created.

For more information on the DATASETS procedure, see *SAS Language Reference: Concepts* and the *SAS Procedures Guide*.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/ ACCESS® Interface to IMS-DL/I Software: Reference, Version 8*, Cary, NC: SAS Institute Inc., 1999. 316 pp.

SAS/ACCESS® Interface to IMS-DL/I Software: Reference, Version 8

ISBN 1–58025–548–5

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

1st printing, October 1999