CHAPTER 1

Introducing SAS/ACCESS Software

Using This Book

SAS/ACCESS Software for Relational Databases: Reference, First Edition provides reference and usage information about SAS/ACCESS software for relational database management systems (DBMSs). This book applies to Version 7 and subsequent releases as well as SAS/ACCESS interfaces that have been available since Release 6.06 of the SAS System.

Introductory, example, and reference information is presented in this book. Part 1, the introductory portion, provides an overview of SAS/ACCESS software. Part 2, the reference portion, describes all the SAS options and statements that can be used with the SAS/ACCESS LIBNAME statement, the ACCESS and DBLOAD procedures, and with the SQL Procedure Pass-Through Facility. Part 4, the examples section, explains how you can use SAS/ACCESS software to accomplish some common tasks. Part 6, the last section of this book, lists the sample data that is used in the book’s examples.

The information in this book applies generally to all the relational database management systems that are supported by SAS/ACCESS software. Information on using individual DBMSs is included in a separate chapter in Part 5 of this book. Features of each SAS/ACCESS interface, such as DBMS-specific procedure statements, options, and system options, are described in each DBMS specific chapter. Appendix 2 also includes a brief overview of DBMSs and how your database administrator (DBA) can best run SAS/ACCESS software with the DBMS at your site.

This book is intended for applications programmers and users who know how to use their host operating system, but who may not be familiar with the SAS System, the DBMS, or with the DBMS-specific SQL (Structured Query Language) that the DBMS
uses. Database administrators might also want to read this book to understand how the interface is implemented and how they should administer it.

---

SAS/ACCESS Changes and Enhancements for Version 7

SAS/ACCESS software provides an interface between the SAS System and relational database management systems (DBMSs). In Version 6, the SAS/ACCESS interface consisted of one or more of the following: the ACCESS and DBLOAD procedures, a SAS/ACCESS interface view engine, and the SQL Procedure Pass-Through Facility. These features continue to be supported in Version 7, in addition to the following new features that are available in Version 7:

- **SAS Explorer window**: Enables you to navigate within the SAS System by using a new, point-and-click, graphical interface. This window replaces the ACCESS window in earlier releases of SAS/ACCESS software.
- **SAS/ACCESS LIBNAME statement**: Enables you to assign a SAS libref directly to a DBMS or objects in a DBMS, such as tables and views. There is no longer a need to create access and view descriptors, although all previously created access and view descriptors are fully supported in Version 7.
- **SAS/ACCESS data set options**: Enable you to use many new SAS/ACCESS LIBNAME options to specify how objects in your DBMS data are processed by SAS.
- **Enhanced naming conventions and support**: Enable you to use 32-byte names for SAS data set and variable names and for DBMS table and column names when you use the new SAS/ACCESS LIBNAME statement.
- **Performance optimization**: Passes joins directly to the DBMS for processing.
- **New macro variables**: Enable you to use two new macro variables, SYSDBRC and SYSDBMSG, for reporting DBMS specific errors.
- **DBMS support in PROC SQL**: Enables you to access DBMS data directly by using PROC SQL.
- **enables you to update and delete DBMS data directly through SQL or through a simple PROC SQL view that references the underlying data.**
Introducing SAS/ACCESS Software

- Enables you to join DBMS data so that the DBMS processes the join more efficiently.
- Enables you to use the new DBCONDITION=data set option to subset DBMS data within a PROC SQL view so that the DBMS processes the subsetting conditions.
- Enables you to store embedded LIBNAME statements in PROC SQL views by using the new using libname syntax.
- Windowing interface not supported in Version 7
  - The Access and View Descriptor windows for creating and editing descriptors are not supported in Version 7. If the ACCESS procedure was available in Version 6 for your DBMS, then you can use ACCESS procedure statements to perform these tasks. Similarly, the DBLOAD windows are no longer supported. If the DBLOAD Procedure was available in Version 6 for your DBMS, then you can use DBLOAD procedure statements to perform the same tasks.
- The UPDATE statement has been added to the ACCESS procedure so that you can update descriptor files.
- Full Version 6 compatibility for existing SAS programs that use descriptors, the ACCESS procedure, the DBLOAD procedure, and the SQL Procedure Pass-Through Facility

SAS Explorer Window

The new graphical interface to the SAS System, the SAS Explorer window, enables you to manage your SAS files. By selecting New from the File pull-down menu then selecting Library, or by using the LIBASSIGN command, you can open the SAS New Library window to assign new librefs.

SAS/ACCESS LIBNAME Statement

SAS/ACCESS engines now provide direct and dynamic access to your DBMS data without using access descriptors or view descriptors. You can now assign a SAS libref directly to a DBMS and then use this libref to identify a table or view in that DBMS. This libref.dbms_table behaves like a SAS data set and, therefore, can be used in the DATA step and in SAS procedures just like a SAS data set. You can read, update, insert, or delete DBMS data, as well as create SAS data sets or SAS data views from the DBMS data. In addition, you can use many new options in the LIBNAME statement to specify how to connect to your DBMS and how to access and process your DBMS data.

SAS/ACCESS Data Set Options

Because the new SAS/ACCESS LIBNAME statement enables you to access DBMS tables and views through a SAS libref, you can use them as data sets. You can use many new SAS/ACCESS data set options, as well as several existing SAS data set options, to specify how your data set is processed by SAS. The SAS/ACCESS data set
options provide greater flexibility and control over row and table locking, buffering data, indexing, the use of DBMS data types, and other DBMS functionality.

Note: Control over locking and buffering might not be available for your DBMS. See your DBMS chapter for details.

---

Enhanced Naming Conventions and Support

SAS software and SAS/ACCESS software now support the following names up to 32 bytes: SAS data set, SAS variable, DBMS table name, and DBMS column name. You can also use mixed case and special characters when you specify a SAS or DBMS table or column name. Several new options - including the SAS/ACCESS LIBNAME options PRESERVE_TAB_NAMES= and PRESERVE_COL_NAMES=, the PROC SQL DQUOTE= option, and the VALIDVARNAME= system option - enable you to support any names in your DBMS even if the name does not conform to SAS naming conventions. See your DBMS chapter for details on the availability of each of these options. The SQL Procedure Pass-Through Facility fully supports Version 7 long names as well.

---

Performance Optimization

SAS/ACCESS provides several enhancements in the areas of indexing, error handling, support of DBMS objects such as triggers and stored procedures, and passing joins to the DBMS. Whenever possible, SAS software now optimizes queries to use indexes. You can also use new LIBNAME and data set options to take advantage of DBMS indexes.

The SAS/ACCESS engines support DBMS triggers and stored procedures in SAS. It also enables you to pass a join to the DBMS for more efficient processing.

In addition, you can list your DBMS tables by using the SAS Explorer window or PROC DATASETS, and you can list the SAS metadata for each DBMS table or view by using the Explorer window or PROC CONTENTS.

---

New Macro Variables

Two new macro variables, SYSDBMSG and SYSDBRC, provide DBMS error message and return code information. These variables are similar to the SQLXMSG and SQLXRC macro variables that are available in the SQL Procedure Pass-Through Facility.

---

DBMS Support in the SQL Procedure

The SQL procedure has several enhancements in Version 7 and many of the enhancements interact with SAS/ACCESS software. For a complete description of PROC SQL, see the SAS Procedures Guide.

After you assign a libref to a DBMS, you can reference this new data set in a PROC SQL statement in order to access, update, or delete DBMS data. You can also embed
LIBNAME information in a PROC SQL view; therefore, every time the PROC SQL view is processed, you automatically connect to the DBMS and you can access its data.

Similarly, you can use the new DBCONDITION= data set option to subset DBMS data within a PROC SQL view so that the DBMS processes the subsetting conditions. You can specify a simple PROC SQL view—that is, a view based on one table—in a PROC SQL UPDATE or DATA step UPDATE statement to update the view's underlying data.

Joins are passed to the DBMS to process whenever possible. The DBMS server performs the join and returns only the results of the join to your SAS session. This provides a major performance enhancement for many of your SAS programs that perform joins.

Finally, you can still retrieve DBMS data from or send DBMS statements to a DBMS through the Pass-Through facility.

---------------

**ACCESS Procedure UPDATE Statement**

You can now use the ACCESS procedure's UPDATE statement to update existing access descriptors and view descriptors that you have created by using the ACCESS procedure. This new statement is provided because the ACCESS procedure windowing interface is not supported in Version 7.

---

**Assigning a Libref Interactively**

One of the easiest ways to associate a libref with a DBMS or SAS data library is to use the New Library window. To open this window, issue the LIBASSIGN command from your SAS session’s command box or command line, or select **New** from the SAS Explorer window’s File pull-down menu and then select **Library**. In Display 1.1 on page 7, the user TESTUSER assigns a libref MYDBLIB to a DB2 database named TESTDB.

**Display 1.1  New Library Window**

![New Library Window](image)

The following list describes how to use the New Library window:

- **Name**: enter the libref that you want to associate with a SAS data library or a DBMS.
- **Engine**: enter the name of your SAS engine or SAS/ACCESS engine. Or, click the down arrow to select a name from the pull-down listing.
Assigning a Libref with the LIBNAME Statement

The global SAS statement, LIBNAME, has been enhanced to enable you to assign a libref to a DBMS. This feature lets you specify a DBMS table or view directly in a DATA step or SAS procedure without using descriptors.

In the following example, the SAS libref MYDBLIB is associated with the ORACLE database named AIRDB_REMOTE. You specify LIBNAME options that enable you to connect to a particular database. (If you set certain environment variables or system options—depending on your operating environment and DBMS—you can omit the connection options.)

The AIRDB_REMOTE database contains a number of DBMS objects, including several tables, such as STAFF. By assigning a libref, the table can be used like a SAS data set and can be specified in any DATA step or SAS procedure.

```sas
libname mydblib oracle user=testuser
    password=testpass path=airdb_remote;

proc sql;
    select idnum, lname
    from mydblib.staff
    where state='NY'
    order by lname;
```

You can use the DBMS data to create a SAS data set, as in this example.

```sas
data newds;
    set mydblib.staff;
    keep=idnum lname fname;
run;
```

You can also use the libref and data set with any other SAS procedure.

```sas
proc print data=mydblib.staff;
run;
```

```sas
proc datasets library=mydblib;
quit;
```

SAS/ACCESS and DBMS Product Support

In Version 7 the SAS/ACCESS interface consists of one or more of the following: the SAS/ACCESS LIBNAME statement, the ACCESS and DBLOAD procedures, a SAS/ACCESS interface engine, the SQL Procedure Pass-Through Facility, and several new options. Not all of these components are in every DBMS specific SAS/ACCESS

Some DBMS products have changed their names since the previous release:

**INGRES**

is now CA-OpenIngres. You use the same alias to access your database data (for example, DBMS=INGRES).

**DB2 for UNIX (AIX) and DB2 for OS/2**

are now both supported in the SAS/ACCESS Interface to DB2. You can use previous aliases to access the data or you can use the new alias DBMS=DB2.

**DEC Rdb**

is now Oracle Rdb. You use the same alias to access your database data (DBMS=RDB).

The SAS/ACCESS Interface to ODBC now enables you to access AS/400 data and SQL Server data. You can no longer access these databases directly. A new conversion procedure, PROC CV2ODBC, enables you to convert Version 6 descriptors (SAS/ACCESS to AS/400 or SAS/ACCESS to SQL Server) and PROC SQL views that are based on Pass-Through code to Version 7 SQL views for compatibility. See your ODBC chapter for more information on PROC CV2ODBC.

---

### Sample Data in This Book

This book uses two sets of tables in order to show you how to use SAS/ACCESS software. See “DBMS Tables (Textile Data)” on page 445. One set of tables was created for a fictitious international textile manufacturer. The second set of tables was created for a fictitious international airline. All the data in the tables is fictitious.

The tables are designed to show how the SAS/ACCESS interface treats data that is stored in DBMS tables. They are not meant as examples for you to follow in designing tables for any purpose.
The following list describes the files that contain examples and data that are used in this book. These files are packaged with your SAS software; see your site administrator for details.

**GENSQL.SAS**
contains the SAS code that creates the DBMS tables used in the Version 7 examples. These tables contain the international airline data.

**GENDBL.SAS**
contains the SAS code that creates the DBMS tables for the Version 6 compatibility examples. These tables contain the international textile manufacturer data.

**GENSAMP.SAS**
contains the SAS code used in the examples chapters.

**DBMACS.SAS**
contains macros that enable any SAS/ACCESS interface to a relational DBMS to create the database connection statements that are used by the DBLOAD procedure in the GENDBL.SAS and GENSAMP.SAS files.

**DBPTMACS.SAS**
contains macros that enable any SAS/ACCESS interface to a relational database to create the database connection statements that are used by the SQL Procedure Pass-Through Facility in the GENDBL.SAS and GENSAMP.SAS files.