



CHAPTER

3

Using the SAS ODBC Driver

<i>Introduction</i>	25
<i>Accessing Your Data Sources (Overview)</i>	25
<i>Prerequisites for Communications</i>	26
<i>Accessing Local SAS Data Sources</i>	26
<i>Starting a SAS ODBC Server</i>	26
<i>Terminating the SAS ODBC Server</i>	27
<i>Accessing Remote (SAS/SHARE) Data Sources</i>	27
<i>The TCP/IP Services File</i>	28
<i>Editing the TCP/IP Services File</i>	28
<i>Other Important Usage Information</i>	28
<i>Using Data Sets That Have One-Level Names</i>	29
<i>Updating Attached Tables</i>	29
<i>Using SQL Statements to Access SAS Data Sources</i>	29
<i>Accessing the SAS Libraries MAPS, SASUSER, and SASHELP</i>	29
<i>Return Codes and Associated Messages</i>	30

Introduction

This chapter provides an overview of how to use the SAS ODBC driver to access your SAS data sources, as well as information that may be important to some users. It also provides information about the SAS servers and the communications software that are used by the driver and about SQL syntax and return codes that are supported by the driver.

Accessing Your Data Sources (Overview)

The details of how you access your SAS data sources depend on which ODBC-compliant Windows application you are using (for example, Microsoft Access or Excel), and on whether you are accessing local or remote data. However, here is an overview of the procedure that you follow:

- 1 Install the SAS ODBC driver. (Refer to the installation instructions that are shipped with the driver.) The installation program creates an ODBC icon in the Windows Control Panel if one does not already exist.
- 2 Double-click on the ODBC icon to access the ODBC and SAS ODBC dialogs.
- 3 Use these dialogs to provide the SAS ODBC driver with the necessary information about your SAS data source(s). (See Chapter 2 for instructions.) In addition to

SAS data files, SAS data sources can include DATA step views, PROC SQL views, or SAS/ACCESS views, all of which contain definitions of data that is stored elsewhere rather than the physical data itself. (See “What Kinds of Data Can I Access with the SAS ODBC Driver?” on page 3 and “SAS Data Sets” on page 5.)

- 4 The next step depends on whether you want to access local data or remote data:

Note: TCP/IP (Winsock) is required for accessing local and/or remote data sources. △

- To access local data see “Accessing Local SAS Data Sources” on page 26.
 - To access remote data sources, see “Accessing Remote (SAS/SHARE) Data Sources” on page 27.
- 5 Consult the documentation for your Windows application for instructions on how to access or import data from other sources. From the list of available data sources, select or specify the name that you assigned to your SAS data source.
 - 6 Select or enter the name of the desired SAS data library (if required by your application).
 - 7 Select or enter the name of the desired SAS data file or view.

For information about configuring file DSNs for ODBC-compliant applications that only support the use of file DSNs (such as Microsoft Excel '97), see the SAS ODBC Driver online help.

Prerequisites for Communications

The SAS ODBC driver interacts with other software to provide you with access to your data sources. Certain prerequisites apply, depending on whether you want to access local data or remote data. These prerequisites are described in the following sections.

Accessing Local SAS Data Sources

To access local SAS data sources, the SAS ODBC driver uses a SAS ODBC server in conjunction with TCP/IP. You must edit your TCP/IP Services file to define your servers before starting the SAS ODBC Server. It is not necessary for the server to be running when you *define* your data sources. However, the server must be running on your PC in order for you to *access* your SAS data sources. For information about editing the TCP/IP Services file, see “The TCP/IP Services File” on page 28. For more information about SAS servers, see “SAS Servers” on page 7.

Starting a SAS ODBC Server

If there isn't already a SAS ODBC server running on your PC, the SAS ODBC driver uses the information that you supplied in the Local Options dialog (described in Display 2.5 on page 17) to automatically start one for you. That is, you do not need to take any action to start the server.

If you already have a SAS session running on your PC, then you can start the SAS ODBC server in that session by submitting the following statements:

```
options com
amid=tcp;
proc odbcserv id=server-name authenticate=optional;
```

```
run;
```

The *server-name* must be the same as the name you specified in the Servers page when you defined your local data source, as explained in “Defining Servers” on page 15.

Alternatively, you can terminate your SAS session so that the SAS ODBC driver can start a SAS ODBC server for you in a new SAS session.*

Note: When the server is running in the SAS session, the SAS session does not accept user input from the keyboard. Δ

If the SAS session cannot be started before the **SAS Timeout** value that you specified in the Local Options dialog is reached, a timeout error is returned to your ODBC client application. An error message is also returned to the client if the SAS session was started but PROC ODBCSESV could not execute.

Terminating the SAS ODBC Server

When you are finished using your ODBC client application to access your local SAS data sources, the SAS ODBC server continues to execute in case you want to access additional SAS data sources. To terminate the server, do either of the following:

- Open the Microsoft Windows Task List window and press the **Ctrl**, **Alt** and **Delete** keys simultaneously. (For Windows NT, use **Ctrl**, **Shift** and **Escape** keys). Select **SAS** from the list of tasks, and then select **End Task**.
- Bring the SAS session into focus (make it the active window), and then press the **Ctrl** and **Break** keys simultaneously. The Task Manager appears. Select **ODBCSESV** from the list of tasks and click **OK**. Click **OK** again to halt the procedure.

Accessing Remote (SAS/SHARE) Data Sources

To access remote data sources, the SAS ODBC driver uses a SAS/SHARE server. (See “SAS Servers” on page 7 for more information. See *SAS/SHARE User's Guide* for complete information about SAS/SHARE.) It is not necessary for the server to be running when you *define* your data sources. However, the server must be running on your remote host in order for you to *access your data sources*.

Because a SAS/SHARE server is used by multiple users, it is usually invoked on the remote host at system startup time. Therefore, end users generally do not need to take any local action to invoke the server.

Whoever is responsible for starting the SAS/SHARE server (you or your server administrator) must do the following:

- 1 Define the server(s) in the TCP/IP Services file for the *client* machine(s) that are running the server(s). * (For more information, see “The TCP/IP Services File” on page 28.)
- 2 Specify **COMAMID=TCP** in an OPTIONS statement when you invoke PROC SERVER, as in the following example:

```
options comamid=tcp;
proc server id=server-name authenticate=optional;
run;
```

* Under Windows NT, it is not necessary to terminate the SAS session because you can run multiple SAS sessions in that environment.

* Refer to the documentation for your TCP/IP software to find the path name for the Services file.

The TCP/IP Services File

The *TCP/IP Services file* contains information about the names of the available services on the machine, along with the port number, protocol name, and any aliases corresponding to each service. For the purposes of the SAS ODBC driver, each entry in this file associates a SAS Server (service name) with the port number and protocol used by that service. The location of the Services file varies on different platforms. In order to configure the SAS ODBC driver appropriately, you must locate the Services file for your platform. Common locations for the Services file are:

Windows 95	C:\WINDOWS\SERVICES
Windows NT	C:\WINNT\SYSTEM32\DRIVERS\ETC\SERVICES
Unix	/etc/services

Entries in the services file take the following general form:

```
<official service name> <port number/protocol name> <aliases> #
<comments>
```

Editing the TCP/IP Services File

To configure your Services file for use with the SAS ODBC driver, you must add an entry to the services file for each SAS server (either local or remote) that you have configured using the ODBC Administrator.

The port number that you use should be an unused port number in this file (for larger networks, contact your network administrator to obtain an available port number). The port number must be above 1024, as any port number equal to or less than 1024 is reserved. The protocol must always be **tcp**. The server name must be one to eight characters long, and it is generally case-sensitive. The first character must be a letter or an underscore; the remaining seven characters can include letters, digits, underscores, the dollar (\$) sign, or the at (@) sign. You specify the same server name in the Servers page when you define your data source as explained in “Defining Servers” on page 15.

For example, if, in the ODBC Administrator, you have configured a local ODBC server named **shr1** and a remote SAS/SHARE server named **machine.shr2** you would add entries to the Services file similar to the following (substituting the appropriate port numbers):

```
shr1    5010/tcp    #Local ODBC Server
shr2    5011/tcp    #Remote SAS/SHARE Server
```

Note: In the case of **shr2**, the administrator of the remote system named **machine** should have already edited the Services file on that system to include the same **shr2** entry and should have started the SAS/SHARE server. Δ

Other Important Usage Information

The information in this section may be important to some users. You should skim through this section to determine whether any of this information is relevant to you.

Using Data Sets That Have One-Level Names

If you use an ODBC application such as Microsoft Access that exports databases using one-level names, you should use the ODBC administrator to define a USER library. The SAS System normally places any data set that has a one-level name into the WORK library, which is deleted at the end of the SAS session. But if a USER library has been defined, SAS places all one-level name data sets into the USER library, which is saved at the end of the SAS session. In a multi-user environment, multiple client connections to a SAS System server can each have their own USER library defined.

Updating Attached Tables

Some Microsoft products that are based on the JET engine (such as Microsoft Access) have certain requirements in order to be able to update database tables. This may be true of other ODBC applications as well. These requirements may make it necessary for you to specify two SQL options when you define your SAS data sources.

- The attached table must have a unique primary key that is not a floating-point value. You can use the Data Source SQL Option **Infer INTEGER from FORMAT** to indicate that SAS numeric fields without fractional parts (for example, **FORMAT(*n*,0)** where *n* is less than 12) are actually integer values that can be used to index the table.
- All of the values in a row may be used to uniquely select the row for updating. This can be a problem in rows that contain floating-point fields (SAS numerics). Insignificant differences in values can be caused either by differences in floating-point representation on different machines or by conversion between character and binary formats. By specifying the Data Source SQL Option **Fuzz Numbers at 12 places**, you can cause WHERE clauses to select values that are "acceptably close" rather than requiring exact comparisons.

See "Naming Your Data Source and Specifying SQL Options" on page 14 and "User-Specified SQL Options" on page 36 for more information about these SQL options.

Using SQL Statements to Access SAS Data Sources

All ODBC-compliant applications use a variety of the Structured Query Language (SQL) to access and manipulate data. However, most of these applications transform user actions into SQL statements so that users themselves do not need to know anything about SQL.

If your application requires you to use SQL statements, or if you use SQL out of personal preference, then you should refer to the chapter "The SQL Procedure" in the *SAS Procedures Guide*. The elements of SQL grammar that are supported by the SAS ODBC driver are the same as those described in that book.

Accessing the SAS Libraries MAPS, SASUSER, and SASHELP

By default, every SAS session (including SAS server sessions) provides access to the SAS libraries MAPS, SASUSER, and SASHELP. However, because these libraries contain sample data sets and other files that are generally not of interest to ODBC users, the SAS ODBC driver does not report the contents of these libraries when it invokes a SAS ODBC server. (From a programming standpoint, when SQLTables,

SQLStatistics, or SQLColumns is called, the result set that is returned does not include rows for the SAS libraries MAPS, SASUSER, or SASHELP.) If you want information from these libraries, you can do either of the following:

- Use a LIBNAME statement to define the desired SAS library to the server.
- Use the SAS ODBC Driver Configuration dialogs to define the desired library to your ODBC applications.

In both cases you must use a different name (that is, not MAPS, SASUSER, or SASHELP) as your libref or library name.

Return Codes and Associated Messages

The SAS ODBC driver uses standard ODBC return codes to notify you of any errors and to provide additional information or warnings. The associated message texts may be generated by the driver itself, by the SAS server, or by your communications access method. See the *Microsoft ODBC 2.0 Programmer's Reference and SDK Guide* and the Appendix in this book for explanations of these return codes and their associated texts.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS[®] ODBC Driver User's Guide and Programmer's Reference, Version 8*, Cary, NC: SAS Institute Inc., 1999.

SAS[®] ODBC Driver User's Guide and Programmer's Reference, Version 8

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

ISBN 1-58025-505-1

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.

AS/400[®], DB2[®], OS/2[®], and OS/390[®] are registered trademarks or trademarks of IBM Corporation. ORACLE[®] and Oracle Rdb[™] are trademarks or registered trademarks of Oracle Corporation. ® 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.