

McGraw-Hill Series on Data Warehousing and Data Management

Roger E. Sanders

ODBC 3.5

Developer's Guide

interfacing



DISK
Included

Oracle Ex. 1030, p. 1
Oracle v. DeBellis
IPR2024-00221

Library of Congress Cataloging-in-Publication Data

Sanders, Roger E.

ODBC 3.5 developer's guide / Roger Sanders.

p. cm.

Includes bibliographical references and index.

ISBN 0-07-058087-1

1. Relational databases. 2. Database design. 3. Microsoft ODBC.
4. Application software—Development. I. Title.

QA76.9.D3S253723 1998

005.75'6—dc21

98-24785

CIP

McGraw-Hill



A Division of The McGraw-Hill Companies

Copyright © 1999 by Roger E. Sanders. All Rights Reserved. Printed in the United States of America. Except as permitted under the United States Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

2 3 4 5 6 7 8 9 0 1FGR/1FGR 9 0 3 2 1 0 9 8

P/N 134226-5

PART OF ISBN 0-07-058087-1

The sponsoring editor for this book was Simon Yates and the production supervisor was Tina Cameron. It was set in Century Schoolbook by Douglas & Gayle Limited.

Printed and bound by Quebecor/Fairfield.

McGraw-Hill books are available at special quantity discounts to use as premiums and sales promotions, or for use in corporate training programs. For more information, please write to Director of Special Sales, McGraw-Hill, 11 West 19th Street, New York, NY 10011. Or contact your local bookstore.

Information contained in this work has been obtained by The McGraw-Hill Companies, Inc. ("McGraw-Hill") from sources believed to be reliable. However, neither McGraw-Hill nor its authors guarantees the accuracy or completeness of any information published herein and neither McGraw-Hill nor its authors shall be responsible for any errors, omissions, or damages arising out of use of this information. This work is published with the understanding that McGraw-Hill and its authors are supplying information but are not attempting to render engineering or other professional services. If such services are required, the assistance of an appropriate professional should be sought.



This book is printed on recycled, acid-free paper containing a minimum of 50% recycled de-inked fiber.

Oracle Ex. 1030, p. 2
Oracle v. DeBellis
IPR2024-00221

applications. Drivers are also responsible for interacting with software that is needed to access a specific data source (for example, the software that interfaces to underlying networks or file systems).

- **The Driver Manager**—The Driver Manager loads and calls one or more drivers on behalf of an application.
- **Data Sources**—Data sources consist of sets of data and their associated environments, which include operating system, DBMS, and network platform used (if any) to access the DBMS that an application wants to access. The term data source is used rather loosely to describe any set of data that has a supporting ODBC driver.

Figure 2-2 shows how these four components work with both file-oriented and client/server DBMS data sources. Notice how in ODBC architecture, the Database API component of client/server architecture (shown in Figure 2-1) has been divided into two components—the ODBC Driver and the ODBC Driver Manager.

Applications

Applications are executable programs that call ODBC API functions to access data in one or more data sources. Because the majority of data access work is done with SQL, applications usually use ODBC API function calls to submit SQL statements to data sources and to retrieve the results (if any) generated when those statements are executed. In addition to calling ODBC API functions, applications perform all other work external to the ODBC interface.

Although many kinds of applications exist, most can be classified as one of the following types:

- **Generic**
- **Vertical**
- **Custom**

GENERIC APPLICATIONS Generic applications, often referred to as shrink-wrapped applications or off-the-shelf applications, are designed to work with a variety of different data sources. Examples include a spreadsheet or statistics package that uses ODBC to import data for further analysis, and a word processing package that uses ODBC to obtain a mailing list

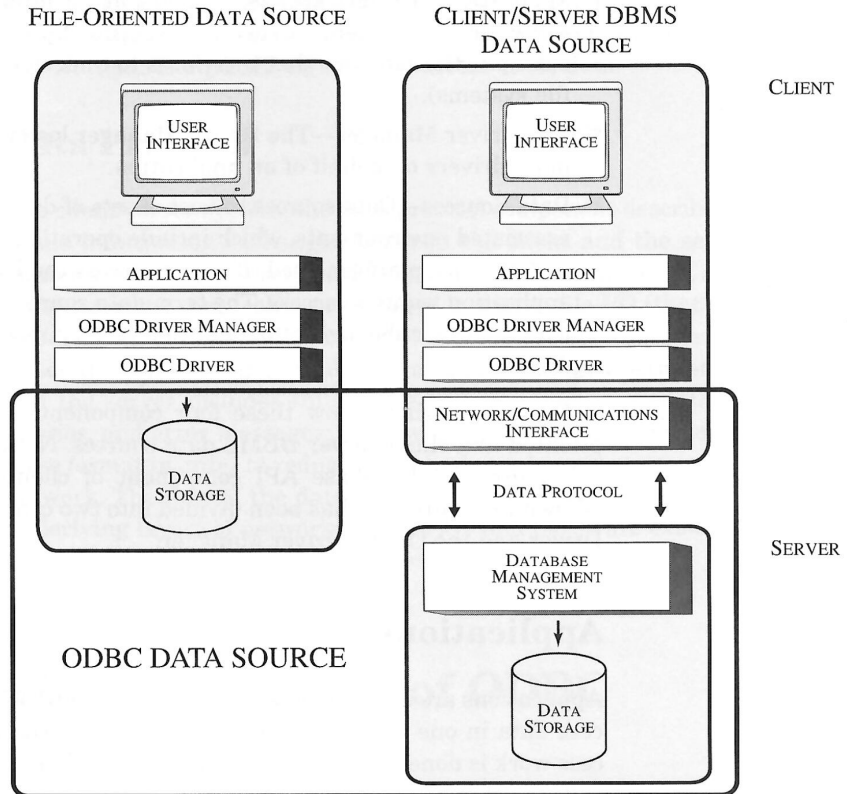


Figure 2-2 How the four components of ODBC architecture recognize and work with file-oriented and client/server DBMS data sources.

from a database. The one thing all generic applications have in common is that they are highly interoperable among data sources and/or DBMS products and they tend to use ODBC in a relatively generic manner.

Interactive application development environments (IDEs), such as PowerBuilder, Microsoft Visual Basic, and Microsoft Visual C++ make up an important subcategory of generic applications. Even though applications constructed with these IDEs may only work with a single data source, the IDE itself needs to be able to work with many different data sources.

Data Sources

A *data source* is simply the source of the data. It can be a flat-file, a particular database in a DBMS, or even a spreadsheet file. For example, a data source might be an Oracle DBMS running on Windows NT, accessed by Novell Netware; an IBM DB2 DBMS running on MVS, accessed through a gateway; a collection of dBASE files in a server directory; or a local Microsoft Access database file.

In ODBC, the term *data source* refers to the actual location of the data as well as all the technical information needed to access the data: the driver name, network address, network software, and so on. Specifically, the term data source is used in ODBC for two purposes:

- As a conceptual term defining the kind of data the end user wants to access (for example, “Payroll,” “Sales,” or “Personnel”). In practice, the data source includes the DBMS software running on a physical machine with some operating system on some hardware platform and the networking software required to access the machine (if appropriate).
- As the actual name an end user or a system administrator assigns via an ODBC utility to describe a particular collection of software components (such as an ODBC driver, a network library, a server name or address, a DBMS, etc.). For example, when an ODBC driver is installed on an end user’s machine, the ODBC installation routine guides the user through the process of creating a data source name for the driver and the target location that identifies where the data the driver will access resides (that is, the directory name on a local machine or the network address of a server).

TYPES OF DATA SOURCES There are two types of data sources in ODBC: machine data sources and file data sources. Although both types contain similar information about the actual location of data and the technical information needed to get to it, they differ how they store this information. Because of this difference, they are used in a somewhat different manner.

Machine data sources are stored on the system with a user-defined name and all the information the Driver Manager and driver need to connect to the actual data source.

With file data sources, all the information the Driver Manager and driver need to connect to the actual data source are stored in a flat file with the .DSN extension. File data sources do not have user-defined data source names like machine data sources do, and they are not registered to any

from these calls are then used with the `SQLConnect()`, `SQLDriverConnect()`, or the `SQLBrowseConnect()` function to establish the desired data source connections. Figure 6-1 illustrates how a connection data storage area is allocated during the initialization task of an application, and how its handle is then passed to the Driver Manager with subsequent ODBC API function calls.

The use of separate connection handles ensures that multi-threaded applications can use one connection per thread without encountering concurrency problems.

Declaring the Application's ODBC Version

A special set of ODBC API function calls can be used to control the attributes associated with an environment and/or a connection. Some of these attributes must be set before a connection to a data source is made while others can only be set afterwards.

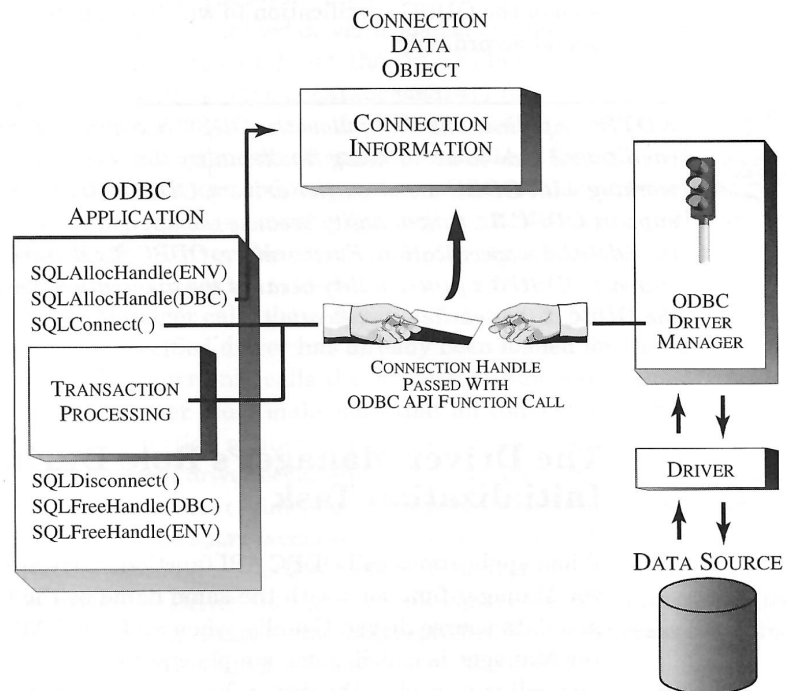


Figure 6-1 How ODBC connection handles are established and used.

- `SQLGetConnectOption()`—ODBC 2.0 or earlier
- `SQLSetConnectOption()`—ODBC 2.0 or earlier
- `SQLGetInfo()` with the `SQL_ODBC_VER` option specified.

If any other ODBC API functions are called, `SQLSTATE 08003` (Connection not open) is returned.

Connecting to a Data Source or Driver

Once a connection handle has been allocated and its attributes/options have been set, an application can use it to connect to a data source or driver. Three different ODBC API functions can be used to establish a connection. They are:

- `SQLConnect()`
- `SQLDriverConnect()`
- `SQLBrowseConnect()`

Applications can use any combination of these functions to connect to any number of data sources and drivers—at the same time if necessary. Additionally, an application's connections can consist of a variety of drivers and data sources, a single driver and a variety of data sources, or even multiple connections to the same driver and data source.



NOTE: Some drivers limit the number of active connections they support. An application can call the `SQLGetInfo()` function with the `SQL_MAX_DRIVER_CONNECTIONS` option specified to determine how many active connections a particular driver supports.

Connecting with `SQLConnect()`

The simplest ODBC API connection function available, `SQLConnect()`, assumes that the only information needed to establish a connection is a data source name and, optionally, a user (authorization) ID and password. This function works well for applications that need to connect to data sources that may or may not require a user ID and/or password. It also

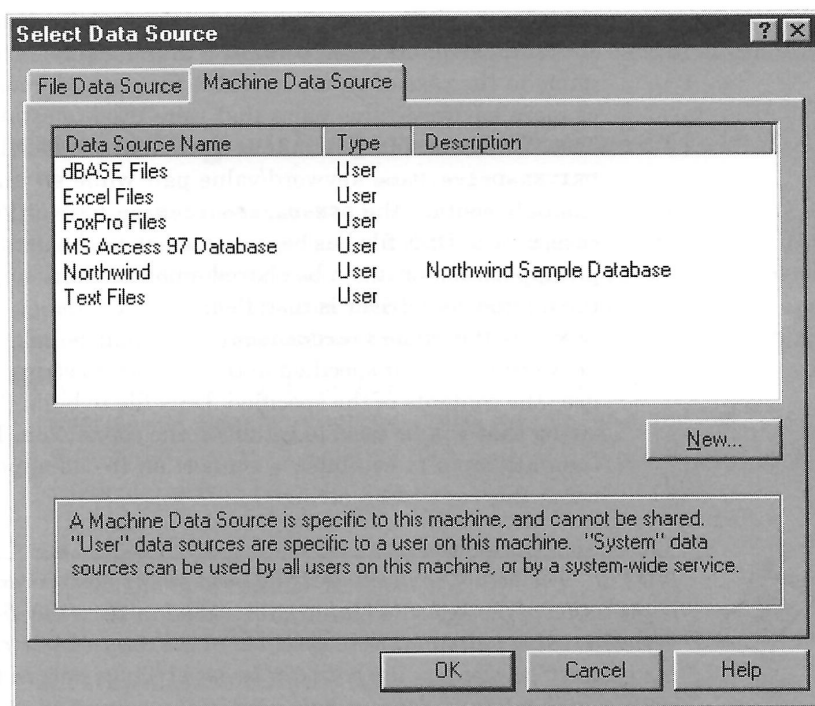


Figure 7-1 The ODBC 3.0 connection information dialog used by `SQLDriverConnect()` to obtain connection information from a user.

Depending on how the `SQLDriverConnect()` function is used, an application can let ODBC prompt the user for everything needed to establish a connection, it can provide the name of a data source and let ODBC prompt the user for any additional connection information needed, or it can provide everything needed to establish a connection without having ODBC to prompt the user for anything.

CONNECTING USING FILE DATA SOURCES Applications that repeatedly build the same connection string to connect to a specific data source can streamline the connection process by using a file data source. With file data sources, all the information needed to establish a connection is stored in a special file, that has a .DSN extension. File data sources are usually created by the `SQLDriverConnect()` function; if `SQLDriverConnect()` is called with a connection string containing the **SAVEFILE** keyword/value pair, the ODBC Driver Manager saves the

See Also

`SQLFreeConnect()`, `SQLAllocHandle()`, `SQLAllocEnv()`,
`SQLConnect()`, `SQLDriverConnect()`, `SQLBrowseConnect()`,
`SQLGetConnectAttr()`, `SQLSetConnectAttr()`, `SQLGetConnectOption()`,
`SQLSetConnectOption()`

Example

See the example provided for the `SQLAllocEnv()` function on page 147.

SQLConnect

COMPATABILITY					
X/OPEN 95 CLI	ISO/IEC 92 CLI	ODBC 1.0	ODBC 2.0	ODBC 3.0	ODBC 3.5
☒	☒	☒	☒	☒	☒
API CONFORMANCE LEVEL				CORE	

Purpose

The `SQLConnect()` function is used to establish a connection to a specified driver and its underlying data source.

Syntax

`SQLRETURN SQLConnect (SQLHDBC ConnectionHandle,`
`SQLCHAR *DSName,`
`SQLSMALLINT DSNameSize,`
`SQLCHAR *UserID,`
`SQLSMALLINT UserIDSize,`
`SQLCHAR *Password,`
`SQLSMALLINT PasswordSize);`

Parameters

- ConnectionHandle*

A data source connection handle that refers to a previously allocated connection information storage buffer (data structure).
- DSName*

A pointer to a location in memory where the name of the data source to connect to is stored.
- DSNameSize*

The length of the data source name value stored in the *DSName* parameter.
- UserID*

A pointer to a location in memory where the user's authorization name (user identifier) is stored.

**Concise yet complete – the developer's definitive guide
to ODBC command structure and syntax**

While other guides may overload you with everything you could possibly know about ODBC (Open Database Connectivity), this one delivers exactly the information you need—no more, no less. Perfect for intermediate- to advanced-level developers, it's your complete, time-saving, at-a-glance guide to mastering ODBC 3.5, the Call Level Interface standard for communicating between relational and object-relational database systems in the Windows environment.

Inside, you'll find full coverage of core concepts, including a frank discussion of the advantages and drawbacks of using ODBC instead of embedded SQL. And you'll get a step-by-step overview of the fundamentals of ODBC architecture, transactions, and concurrency control, the basic types of SQL statements used, and the ODBC components that interact with applications. This one-of-a-kind guide will also get you up to speed on the basic steps of building, testing, and debugging executable applications that maximize ODBC's legendary flexibility. Then it delves deeply into the ODBC API functions and offers tested programming tips that will help you maximize the power of ODBC—without sacrificing hours of trial-and-error programming.

And because several versions of ODBC drivers and Driver Managers are so widely available, this handbook also helps you sort out the numerous differences between versions of ODBC and the international standards upon which ODBC is based. Rich in practical examples and techniques you can put to work right away, the *ODBC 3.5 Developer's Guide* is your shortcut to a new understanding of this incredibly useful interface.

About the Author

Roger E. Sanders is an Educational Multimedia Assets Specialist with the Secondary Education Development Group at SAS Institute, Inc. He has been designing and programming software for IBM PCs and mainframes for more than 15 years. Sanders writes for trade journals such as *DB2 Magazine* and is also the author of McGraw-Hill's *The Developer's Handbook to DB2 for Common Servers*. He is based in Fuquay Varina, North Carolina.

UNIVERSITY BOOKSTORE

P

PRICE \$59.99

SANDERS

ODBC 3.5 DEVELOPE

586 0-07-058087-1

9

3377473

0707040-0200

39785 30624 5

Cover Design & Illustration: DesignConcept

www.books.mcgraw-hill.com

McGraw-Hill

A Division of The McGraw-Hill Companies

