

Progress
ODBC Driver Guide

© 2001 Progress Software Corporation. All rights reserved.

Progress® software products are copyrighted and all rights are reserved by Progress Software Corporation. This manual is also copyrighted and all rights are reserved. This manual may not, in whole or in part, be copied, photocopied, translated, or reduced to any electronic medium or machine-readable form without prior consent, in writing, from Progress Software Corporation.

The information in this manual is subject to change without notice, and Progress Software Corporation assumes no responsibility for any errors that may appear in this document.

The references in this manual to specific platforms supported are subject to change.

Progress, Progress Results, Provision and WebSpeed are registered trademarks of Progress Software Corporation in the United States and other countries. Apptivity, AppServer, ProVision Plus, SmartObjects, IntelliStream, and other Progress product names are trademarks of Progress Software Corporation.

SonicMQ is a trademark of Sonic Software Corporation in the United States and other countries.

Progress Software Corporation acknowledges the use of Raster Imaging Technology copyrighted by Snowbound Software 1993-1997 and the IBM XML Parser for Java Edition.

© IBM Corporation 1998-1999. All rights reserved. U.S. Government Users Restricted Rights — Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Progress is a registered trademark of Progress Software Corporation and is used by IBM Corporation in the mark Progress/400 under license. Progress/400 AND 400® are trademarks of IBM Corporation and are used by Progress Software Corporation under license.

Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Any other trademarks and/or service marks contained herein are the property of their respective owners.

.

May 2001



Product Code: 4510
Item Number: 81077;9.1C

Contents

Preface . . . . . **vii**
Purpose vii
Audience vii
Organization of This Manual vii
How to Use This Manual viii
Typographical Conventions viii
Syntax Notation ix
Other Useful Documentation xii
Getting Started xii
4GL xiii
Database xiii
DataServers xiv
SQL-92 xiv
SQL-92 Reference xiv
ODBC Reference xv

1. Introduction **1–1**
1.1 Overview of ODBC 1–2
1.1.1 The Progress SQL-92 ODBC Driver 1–2
1.1.2 ODBC Architecture 1–3

2. Configuring Progress Data Sources **2–1**
2.1 Configuring Data Sources with the ODBC Administrator Utility 2–2
2.1.1 Adding a New Data Source 2–2
2.1.2 Modifying a Data Source 2–7
2.1.3 Test Connect Failure 2–8
2.1.4 Test Connect Success 2–9
2.2 Connecting to a Data Source Using a Logon Dialog Box 2–9
2.3 Connecting to a Data Source Using a Connection String 2–10

2.4 Isolation and Lock Levels Supported 2–11
2.5 ODBC Conformance Level 2–11
2.6 Connections and Statements Supported 2–11

3. ODBC API and Scalar Functions 3–1
3.1 API Functions 3–2
3.2 Scalar Functions 3–4
3.3 String Functions 3–4
3.4 Numeric Functions 3–7
3.5 Date and Time Functions 3–9
3.6 System Functions 3–12

4. The UNIX Environment 4–1
4.1 Configuring the ODBC Driver 4–2
4.1.1 Setting Environment Variables 4–2
4.1.2 Configuring A Progress Data Source 4–3
4.1.3 The System Information File (.odbc.ini) 4–3
4.1.4 Sample Solaris System Information File 4–3
4.1.5 The ODBCINI Environment Variable. 4–4
4.2 Running an ODBC Client Application on UNIX 4–5
4.3 Using Double-Byte Character Sets 4–6
4.4 Translators 4–6

A. SQLGetInfo A–1

B. Progress and ODBC Data Types B–1

Glossary Glossary–1

Index Index–1

Figures

Figure 1–1: Components of a Progress ODBC Environment 1–4

Tables

Table 2–1:	Connection String Attributes	2–10
Table 3–1:	Scalar String Functions	3–5
Table 3–2:	Scalar Numeric Functions	3–7
Table 3–3:	Scalar Time and Date Functions	3–10
Table 3–4:	Scalar System Functions	3–12
Table 4–1:	ODBC Driver File Location by Platform	4–2
Table 4–2:	ODBC Driver Environment Variables	4–2
Table 4–3:	System Information File Entries by Platform	4–4
Table 4–4:	Runtime System Shared Library Path Settings	4–5
Table 4–5:	Translators	4–6
Table A–1:	Information the Progress SQL-92 ODBC Driver Returns to SQLGetInfo	A–2
Table B–1:	Progress and ODBC Data Types	B–2

Preface

Purpose

This manual describes how to configure the Progress SQL-92 ODBC driver to provide access to the Progress database from desktop tools and applications that support the ODBC (Open Database Connectivity) interface. This manual also provides details about the information returned to your application by the driver. The Progress SQL-92 ODBC driver is supplied by MERANT Inc. In this manual, the driver will be referred to generically as the Progress SQL-92 ODBC driver.

Audience

This book is designed as a guide for anyone who wants to use the Progress SQL-92 ODBC driver to access a Progress Version 9 database.

Organization of This Manual

Chapter 1, "Introduction"

Introduces the Progress SQL-92 ODBC driver and describes how it works.

Chapter 2, "Configuring Progress Data Sources"

Describes how to add and modify, Progress data sources that use the Progress SQL-92 ODBC driver on Windows.

Chapter 3, "ODBC API and Scalar Functions"

This chapter lists the ODBC API functions that the Progress SQL-92 ODBC driver supports. It also lists the scalar functions used in SQL-92 statements

Chapter 4, "The UNIX Environment"

This chapter describes how to configure the Progress SQL-92 ODBC Driver in the UNIX environment.

Appendix A, "SQLGetInfo"

This appendix shows information the Progress SQL-92 ODBC driver Returns to SQLGetInfo.

Appendix B, "Progress and ODBC Data Types"

This appendix shows how the Progress data types are mapped to the standard ODBC data types.

"Glossary"

Defines terms introduced in this guide.

Index

How to Use This Manual

This book assumes that you have successfully installed the Progress database as described in the *Progress Installation and Configuration Guide Version 9 for Windows*, or the *Progress Installation and Configuration Guide Version 9 for UNIX*.

Typographical Conventions

This manual uses the following typographical conventions:

- **Bold typeface** indicates:
  - Commands or characters that the user types
  - That a word carries particular weight or emphasis
- *Italic typeface* indicates:
  - Progress variable information that the user supplies
  - New terms
  - Titles of complete publications

- `Monospaced typeface` indicates:
  - Code examples
  - System output
  - Operating system filenames and pathnames

The following typographical conventions are used to represent keystrokes:

- Small capitals are used for Progress key functions and generic keyboard keys.

  END-ERROR, GET, GO
  ALT, CTRL, SPACEBAR, TAB

- When you have to press a combination of keys, they are joined by a dash. You press and hold down the first key, then press the second key.

  CTRL-X

- When you have to press and release one key, then press another key, the key names are separated with a space.

  ESCAPE H
  ESCAPE CURSOR-LEFT

Syntax Notation

The syntax for each SQL-92 component follows a set of conventions:

- Uppercase words are keywords. Although they are always shown in uppercase, you can use either uppercase or lowercase when using them in an expression or a statement.

  In this example, GRANT, RESOURCE, DBA, and TO are keywords:

  SYNTAX

  ```
  GRANT { RESOURCE, DBA } TO user_name [, user_name ] ... ;
  ```

- Italics identify options or arguments that you must supply. These options can be defined as part of the syntax or in a separate syntax identified by the name in italics. In the GRANT statement you must supply at least one *user_name*.

- Square brackets (**[ ]**) around an item indicate that the item, or a choice of one of the enclosed items, is optional.

In this syntax example the first *user_name* is required, and additional *user_name* specifications are optional:

SYNTAX

```
GRANT { RESOURCE, DBA } TO user_name [, user_name ] ... ;
```

NOTE: The ellipsis (**...**) indicates repetition, as shown in a following description.

- Braces (**{ }**) around an item indicate that the item, or a choice of one of the enclosed items, is required.

  In the GRANT example, you must specify either RESOURCE or DBA or both, and at least one *user_name*. Any *user_name* specifications after the first are optional:

  SYNTAX

  ```
  GRANT { RESOURCE, DBA } TO user_name [,user_name ] ... ;
  ```

  In some cases, braces are not a syntax notation, but part of the language.

  For example, this excerpt from an ODBC application invokes a stored procedure using the ODBC syntax { call *procedure_name* (*param*) }, where braces and parentheses are part of the language:

  SYNTAX

  ```
  proc1( param, "{ call proc2 (param) }", param);
  ```

- A vertical bar (**|**) indicates a choice. In the CREATE SYNONYM example you must specify a *table_name*, *view_name*, or *synonym* but you can only choose one. Note that in all SQL-92 syntax, if you specify the optional *owner_name* qualifier, there must not be a blank between the period separator and the *table_name* or *view_name* or *synonym*:

  SYNTAX

  ```
  CREATE [ PUBLIC ] SYNONYM synonym
  FOR [ owner_name. ] { table_name | view_name | synonym } ;
  ```

In the DELETE FROM example, you must specify one of *table_name* or *view_name*:

SYNTAX

```
DELETE FROM [ owner_name.]{table_name | view_name}
[ WHERE search_condition];
```

- Ellipses (**. . .**) indicate that you can choose one or more of the preceding items. If a group of items is enclosed in braces and followed by ellipses, you must choose one or more of those items. If a group of items is enclosed in brackets and followed by ellipses, you can optionally choose one or more of those items.

  In the ORDER BY example, you must include one expression *(expr)* or column position *(posn)*, and you can optionally specify the sort order as ascending (*ASC*) or descending (*DESC*). You can specify additional expressions or column positions for sorting within a sorted result set. The SQL-92 engine orders the rows on the basis of the first *expr* or *posn*. If the values are the same, the second *expr* or *posn* is used in the ordering:

  SYNTAX

  ```
  ORDER BY { expr | posn} [ ASC | DESC ]
    [ , [ { expr | posn} [ ASC | DESC ] ] ... ]
  ```

  In the GRANT example, you must include one *user_name*, but you can optionally include more. Note that each subsequent *user_name* must be preceded by a comma:

  SYNTAX

  ```
  GRANT { RESOURCE, DBA } TO user_name [, user_name ] ... ;
  ```

- In many examples, the syntax is too long to place in one horizontal row. In such cases, optional items appear individually bracketed in multiple rows in order, left-to-right and top-to-bottom. This order generally applies, unless otherwise specified. Required items also appear on multiple rows in the required order, left-to-right and top-to-bottom. In cases where grouping and order might otherwise be ambiguous, braced (required) or bracketed (optional) groups clarify the groupings.

In this example, CREATE VIEW is followed by several optional items:

SYNTAX

```
CREATE VIEW [ owner_name. ] view_name
 [ ( column_name [, column_name ] ... ) ]
  AS [ ( ] query_expression [ ) ] [ WITH CHECK OPTION ] ;
```

Other Useful Documentation

This section lists Progress Software Corporation documentation that you might find useful. Unless otherwise specified, these manuals support both Windows and Character platforms and are provided in electronic documentation format on CD-ROM.

Getting Started

Progress Electronic Documentation Installation and Configuration Guide (Hard copy only)

A booklet that describes how to install the Progress EDOC viewer and collection on UNIX and Windows.

Progress Installation and Configuration Guide Version 9 for UNIX

A manual that describes how to install and set up Progress Version 9.1 for the UNIX operating system.

Progress Installation and Configuration Guide Version 9 for Windows

A manual that describes how to install and set up Progress Version 9.1 for all supported Windows and Citrix MetaFrame operating systems.

Progress Version 9 Product Update Bulletin

A guide that provides a brief description of each new feature of the release. The booklet also explains where to find more detailed information in the documentation set about each new feature.

Progress Master Glossary for Windows and *Progress Master Glossary for Character* (EDOC only)

Platform-specific master glossaries for the Progress documentation set. These books are in electronic format only.

Progress Master Index and Glossary for Windows and *Progress Master Index and Glossary for Character* (Hard copy only)

Platform-specific master indexes and glossaries for the Progress hard-copy documentation set.

Progress Startup Command and Parameter Reference

A reference manual that describes the Progress startup commands and parameters in alphabetical order.

Welcome to Progress (Hard copy only)

A booklet that explains how Progress software and media are packaged. An icon-based map groups the documentation by functionality, providing an overall view of the documentation set. *Welcome to Progress* also provides descriptions of the various services Progress Software Corporation offers.

4GL

Progress Internationalization Guide

A guide to developing Progress applications for markets worldwide. The guide covers both internationalization—writing an application so that it adapts readily to different locales (languages, cultures, or regions)—and localization—adapting an application to different locales.

Database

Progress Database Design Guide

A guide that uses a sample database and the Progress Data Dictionary to illustrate the fundamental principles of relational database design. Topics include relationships, normalization, indexing, and database triggers.

Progress Database Administration Guide and Reference

This guide describes Progress database administration concepts and procedures. The procedures allow you to create and maintain your Progress databases and manage their performance.

DataServers

Progress DataServer Guides

These guides describe how to use the DataServers to access non-Progress databases. They provide instructions for building the DataServer modules, a discussion of programming considerations, and a tutorial. Each DataServer has its own guide, such as: the *Progress DataServer for Microsoft SQL Server Guide*, the *Progress DataServer for ODBC Guide*, the *Progress DataServer for ORACLE Guide*, or the *Progress/400 Product Guide*

SQL-92

Progress Embedded SQL-92 Guide and Reference

A guide to Progress Embedded SQL-92 for C, including step-by-step instructions for building ESQL-92 applications and reference information about all Embedded SQL-92 Preprocessor statements and supporting function calls. This guide also describes the relationship between ESQL-92 and the ANSI standards upon which it is based.

Progress JDBC Driver Guide

A guide to the Java Database Connectivity (JDBC) interface and the Progress SQL-92 JDBC driver. It describes how to set up and use the driver and details the driver's support for the JDBC interface.

Progress SQL-92 Guide and Reference

A user guide and reference for programmers who use Progress SQL-92. It includes information on all supported SQL-92 statements, SQL-92 Data Manipulation Language components, SQL-92 Data Definition Language components, and Progress functions. The guide describes how to use the Progress SQL-92 Java classes and how to create and use Java stored procedures and triggers.

SQL-92 Reference

(These are non-Progress resources available from your technical bookseller.)

A Guide to the SQL Standard

Date, C.J., with Hugh Darwen. 1997. Reading, MA: Addison Wesley.

Understanding the New SQL: A Complete Guide

Melton, Jim (Digital Equipment Corporation) and Alan R. Simon. 1993. San Francisco: Morgan Kaufmann Publishers.

ODBC Reference

Microsoft ODBC 3.0 Programmer's Reference

Microsoft Press
A Division of Microsoft Corporation
One Microsoft Way
Redmond, Washington, 1997

1

Introduction

The Progress SQL-92 ODBC (Open Database Connectivity) Driver is supplied by MERANT Inc., formerly known as Intersolv. In this manual, the driver will be referred to generically as the Progress SQL-92 ODBC driver. It runs on Windows, Window NT, and UNIX environments. It is installed when you install the Progress SQL-92 Client Access product.

This chapter provides a brief overview of the ODBC interface, and describes how the Progress SQL-92 ODBC driver provides access from a wealth of desktop tools to the Progress database.

1.1 Overview of ODBC

The ODBC interface from Microsoft has emerged as the standard mechanism for client applications to access data from a variety of different sources through a single interface. Users of applications that support ODBC merely select a new database from a point-and-click menu to connect, transparently, to data from that data source.

- ODBC, Microsoft's Open Data Base Connectivity has become the de facto database access standard on the desktop.
- Desktop tools and applications that support ODBC can access any data source supported by an ODBC driver. These tools include client/server development, query, and personal productivity tools.

To become accessible from ODBC client applications, database environments must provide software, called an ODBC driver, on the client system where the application resides. The driver translates the standard ODBC function calls into calls the data source can process, and returns data to the application. Each data source provides a driver on the client system for applications to use to access data from that source.

1.1.1 The Progress SQL-92 ODBC Driver

The Progress SQL-92 ODBC driver allows tools and applications that support the ODBC call library to use the Progress database as a data source. With it, applications based on tools such as Crystal Reports, PowerBuilder, and Visual Basic can include a Progress database as a data source.

The ODBC interface specifies two major components:

- A library of function calls that allow applications to connect with a database system and issue statements through an application programming interface (API)
- Syntax for Structured Query Language (SQL) statements, based on existing standards

1.1.2 ODBC Architecture

The ODBC driver fits in as a layer of "middleware" in the ODBC architecture, which includes the following components:

Application

An ODBC application is any program that calls ODBC functions and uses them to issue SQL statements. Many vendors have added ODBC support to their existing Windows-based tools so those tools can use ODBC for data access.

ODBC Driver Manager

On Windows platforms, the ODBC driver manager is a Microsoft-supplied dynamic-link library (DLL). On UNIX platforms, the ODBC driver manager is a MERANT-supplied shared library. The driver manager routes calls from an application to the ODBC driver. To an application, the ODBC driver manager and the driver are a single entity that processes requests to the database. The ODBC driver manager loads the requested driver in response to an application's call to the ODBC SQLConnect or SQLDriverConnect functions.

ODBC Driver

An ODBC driver is a dynamic link library (DLL) or a shared library that processes ODBC function calls for a specific data source. The driver connects to the data source, translates the standard SQL statements into syntax the data source can process, and returns data to the application. There are ODBC drivers for every major database system.

Data Source

A data source is a combination of a database system, the operating system it uses, and any network software required to access it. ODBC defines a database system (DBMS) as any vendor's implementation of a data access system that provides an SQL interface.

Figure 1–1 shows the components in a Progress ODBC environment.

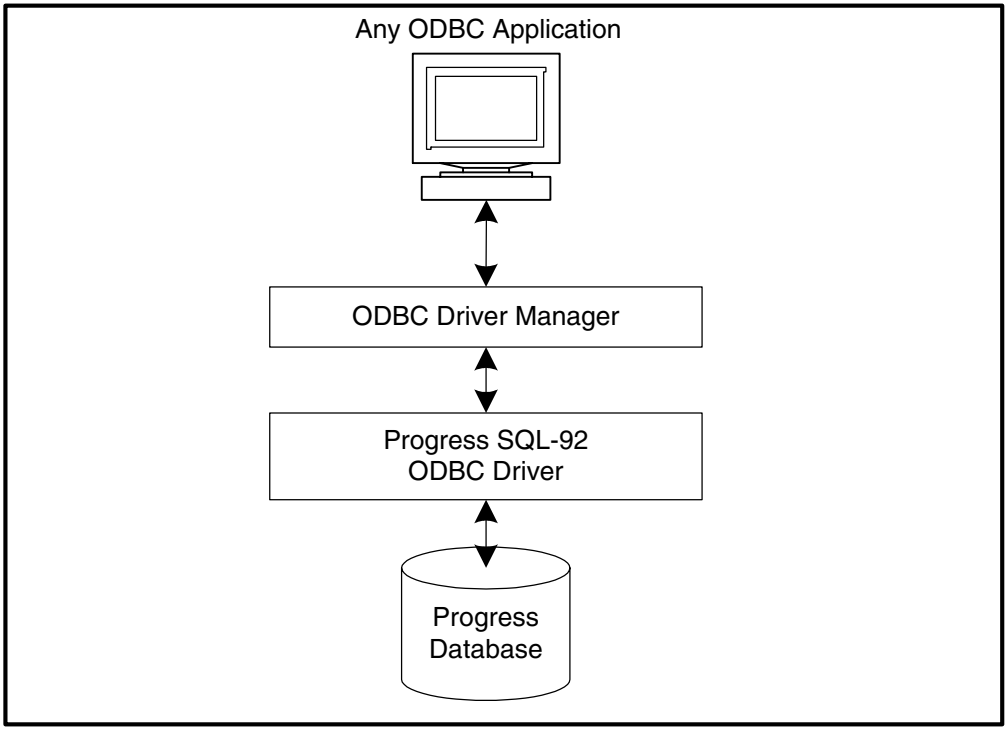


Figure 1–1: Components of a Progress ODBC Environment

2

Configuring Progress Data Sources

This chapter describes how to use the ODBC Administrator Utility to add and modify Progress SQL-92 ODBC data sources on Windows client systems. For information on configuring UNIX client systems, see Chapter 4, "The UNIX Environment."

2.1 Configuring Data Sources with the ODBC Administrator Utility

The ODBC Administrator is a Microsoft utility for configuring ODBC data sources and drivers. The Progress installation procedure installs the ODBC Driver Manager software and the associated ODBC Data Source Administrator utility if it was not already installed.

2.1.1 Adding a New Data Source

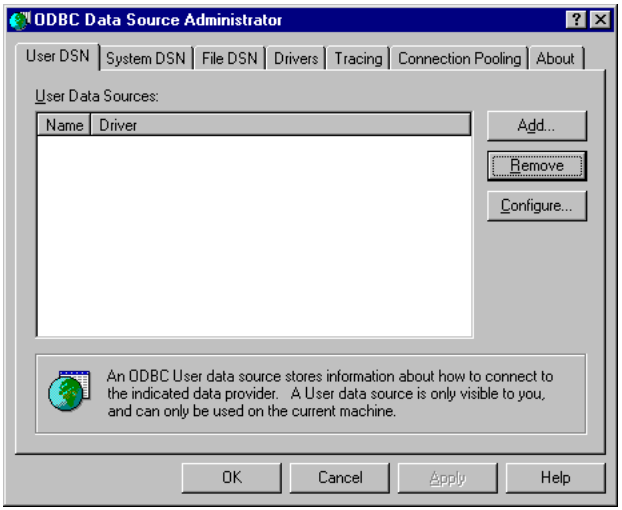
To add an ODBC data source:

1 ♦ Choose Settings→ Control Panel from the Windows Start menu. The Control Panel displays.

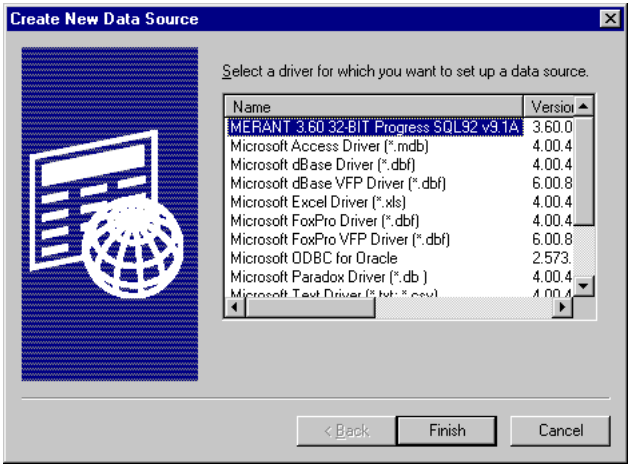
2 ♦ Double-click the ODBC icon in the Control Panel:



3 ♦ The ODBC Administrator displays the ODBC Data Source Administrator dialog box:



4 ♦ Click Add to display a list of installed drivers. The Create New Data Source dialog box appears:



5 ♦ Highlight the MERANT 3.60 32-BIT Progress SQL-92 ODBC driver, then click Finish to display the ODBC Progress SQL-92 Driver Setup dialog box.

6 ♦ Specify values for the following:

Data Source Name

Identifies this Progress data source configuration in the system information. Examples include “Accounting” or “9.1 newsports.”

Description

An optional long description of the data source name. For example, “My Accounting Database” or “Progress 9.1 local newsports.”

Host Name

The name of the system where the database or database broker is located.

Port Number

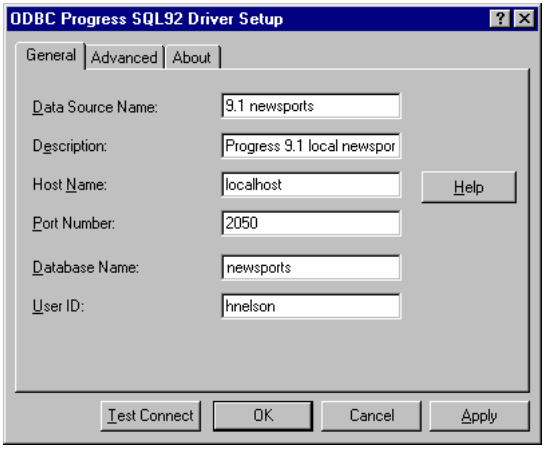
The system port number setup for the database listener process.

Database Name

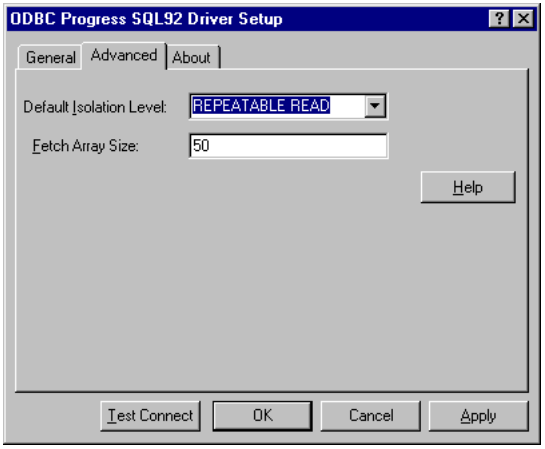
The name of the database to which you want to connect by default.

User ID

The default logon ID (user name) used to connect to your Progress database. Your ODBC application may override this value or you may override it in the Logon dialog box or connection string:



7 ♦ Click on the Advanced tab:



8 ♦ Specify values for the following:

Default Isolation Level

Specifies the default isolation level for concurrent transactions. Choose from the following values: READ COMMITTED, READ UNCOMMITTED, REPEATABLE READ, and SERIALIZABLE. REPEATABLE READ is the default. See the *Progress SQL-92 Guide and Reference* for an explanation of the SET TRANSACTION ISOLATION LEVEL command.

Fetch Array Size

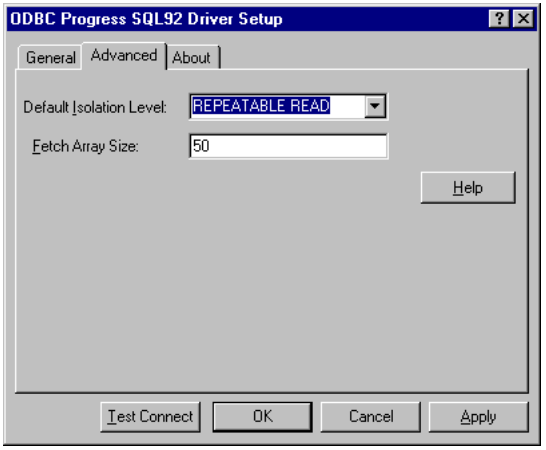
The number of rows the driver retrieves when fetching from the server. This is not the number of rows given to the user. The default is 50.

9 ♦ Click Test Connect to attempt to connect to the data source using the connection properties specified in the Driver Setup dialog box.

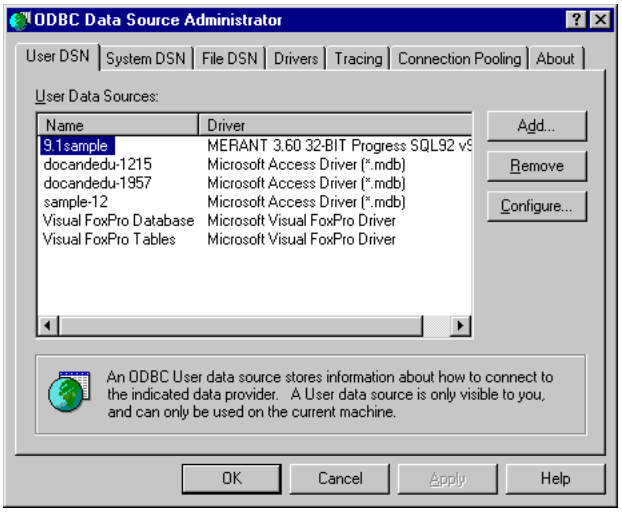
10 ♦ If the driver can connect, it releases the connection and displays a connection established message. Click OK:



11 ♦ The ODBC Progress SQL92 Driver Setup dialog box appears. Click Apply:



12 ♦ Click OK or Cancel to close the Driver Setup dialog box. If you click OK, the values you have specified become the defaults when you connect to the data source. You can change these defaults by using this procedure to reconfigure your data source. You can override these defaults by connecting to the data source using a connection string with alternative values. The data source appears in the ODBC Data Source Administrator:



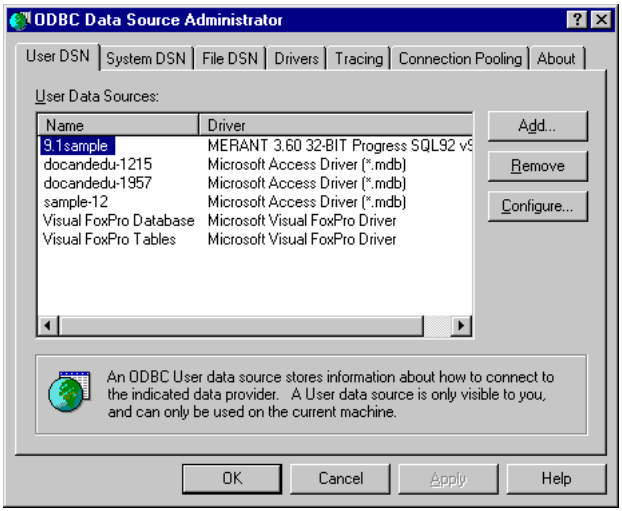
2.1.2 Modifying a Data Source

To modify an ODBC data source:

1 ♦ Choose Settings→ Control Panel from the Windows Start menu. The Control panel displays:

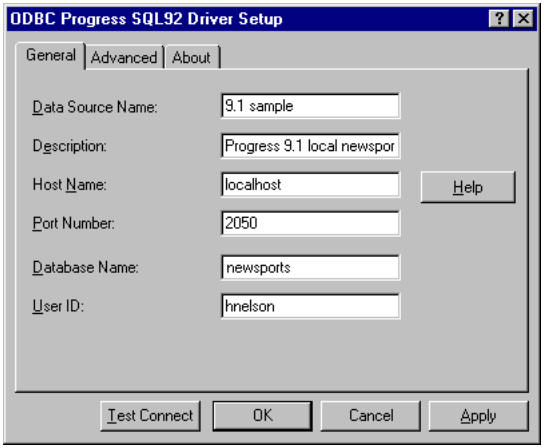


2 ♦ Double click the ODBC icon in the Control Panel. The ODBC Administrator displays the ODBC Data Source Administrator dialog box:



3 ♦ Highlight the User Data Source Name you wish to modify and click Configure. The ODBC Progress SQL-92 Driver Setup dialog box appears.

4 ♦ Modify the fields you wish to change:



5 ♦ Click Apply.

6 ♦ Click OK.

2.1.3 Test Connect Failure

If you click on Test Connect and the driver cannot connect because of an improper environment or incorrect connection value, you receive an error message.

Click OK. The following message box appears:



You must verify that the all required client software is properly installed before you can proceed. Refer to the *Progress Installation and Configuration Guide Version 9 for Windows* for more information.

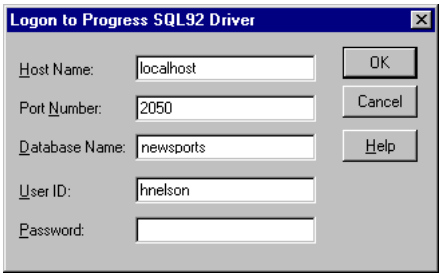
2.1.4 Test Connect Success

If you click on Test Connect and the driver does connect successfully, the following dialog box appears:



2.2 Connecting to a Data Source Using a Logon Dialog Box

Some ODBC applications display a logon dialog box when you are connecting to a data source. In these cases, the data source name has already been specified. The Progress SQL-92 ODBC driver dialog box is as follows:



In this dialog box, enter the following:

1 ♦ Type the name of the system where the database is stored.

2 ♦ Type the Port Number setup for the database listener process.

3 ♦ Type the name of the database to which you want to connect.

4 ♦ Type your user name.

5 ♦ If required, type your password. Password is an optional parameters. You can log on to the database without this parameter, but you may not be able to create, delete, or manipulate the data.

6 ♦ Click OK to complete the logon.

2.3 Connecting to a Data Source Using a Connection String

If your application requires a connection string to connect to a data source, you must specify the data source name that tells the driver which section in the system information to use for the default connection information. Optionally, you may specify attribute=value pairs in the connection string to override the default values stored in the system information. These values are not written to the system information. You can specify either long or short names in the connection string. The connection string has the form:

```
DSN=data_source_name[;attribute=value[;attribute=value]...]
```

An example of a connection string is:

```
DSN=PROGRESS;DB=PAYROLL;UID=JOHN;PWD=XYZZY;HOST=LOCALHOST;
PORT=2055
```

Table 2–1 gives the long and short names for each attribute, as well as a description.

The defaults listed in the table are initial defaults that apply when no value is specified in either the connection string or in the data source definition in the system information. If you specified a value for the attribute when configuring the data source, that value is the default.

Table 2–1: Connection String Attributes *(1 of 2)*

Attribute	Description
Database (DB)	The name of database to which you want to connect.
DataSourceName (DSN)	A string that identifies a Progress data source configuration in the system information
HostName (HOST)	The name of the system where the database is stored.
LogonID (UID)	The default logon ID (user name) used to connect to your Progress database. A logon ID is required only if security is enabled on your database. If so, contact your system administrator to get your logon ID. This ID is case-sensitive.

Table 2–1: Connection String Attributes *(2 of 2)*

Attribute	Description
Password (PWD)	A case-sensitive password.
PortNumber (PORT)	The system port number setup for the database listener process.

2.4 Isolation and Lock Levels Supported

Progress supports isolation level 0 (read uncommitted), isolation level 1 (read committed), isolation level 2 (repeatable read), and isolation level 3 (serializable). For more information on isolation levels see the *Progress SQL-92 Guide and Reference*.

2.5 ODBC Conformance Level

See Chapter 3, "ODBC API and Scalar Functions," for a list of the API functions supported by the Progress SQL-92 ODBC driver. The driver supports the core SQL grammar.

2.6 Connections and Statements Supported

Progress supports multiple connections and multiple statements per connection.

3

ODBC API and Scalar Functions

This chapter lists the ODBC API functions that the Progress SQL-92 ODBC driver supports and the scalar functions, which you use in SQL-92 statements.

3.1 API Functions

The Progress SQL-92 ODBC driver is ODBC Level 1-compliant—it supports all ODBC Core and Level 1 functions. The driver supports the functions listed below:

SQLAllocHandle

SQLBindCol

SQLBindParameter

SQLBulkOperations

SQLCancel

SQLCloseCursor

SQLColAttribute

SQLColumns

SQLConnect

SQLCopyDesc

SQLDataSources

SQLDescribeCol

SQLDisconnect

SQLDriverConnect

SQLDrivers

SQLEndTran

SQLExecDirect

SQLExecute

SQLExtendedFetch (backward and forward scrolling)

SQLFetch

SQLFetchScroll (backward and forward scrolling)

SQLFreeHandle

SQLFreeStmt

SQLGetConnectAttr

SQLGetCursorName

SQLGetData

SQLGetDescField

SQLGetDescRec

SQLGetDiagField

SQLGetDiagRec

SQLGetEnvAttr

SQLGetFunctions

SQLGetInfo

SQLGetStmtAttr

SQLGetTypeInfo

SQLMoreResults

SQLNativeSql

SQLNumParams

SQLNumResultCols

SQLParamData

SQLPrepare

SQLPutData

NOTE: The MERANT Progress SQL-92 ODBC Driver has an internal limitation of 2 Mg when a user specifies the placement of data through the SQLPutData call.

SQLRowCount

SQLSetConnectAttr

SQLSetCursorname

SQLSetDescField

SQLSetDescRec

SQLSetEnvAttr

SQLSetStmtAttr

SQLSetPos

SQLSpecialColumns

SQLStatistics

SQLTables

SQLTransact

3.2 Scalar Functions

The following tables list the scalar functions that ODBC supports. You can use these functions in SQL statements using the following syntax:

SYNTAX

```
{fn scalar-function}
```

where *scalar-function* is one of the functions listed in the following tables. For example:

```
SELECT {fn UCASE(NAME)} FROM EMP
```

3.3 String Functions

Table 3–1 lists the string functions that ODBC supports.

The string functions listed can take the following arguments:

- *string_exp* can be the name of a column, a string literal, or the result of another scalar function, where the underlying data type is SQL_CHAR or SQL_VARCHAR.
- *start*, *length*, and *count* can be the result of another scalar function or a literal numeric value, where the underlying data type is SQL_TINYINT, SQL_SMALLINT, or SQL_INTEGER.

The string functions are one-based; that is, the first character in the string is character 1.

Character string literals must be surrounded in single quotation marks.

Table 3–1: Scalar String Functions *(1 of 3)*

Function	Returns
ASCII(*string_exp*)	ASCII code value of the leftmost character of *string_exp* as an integer.
BIT_LENGTH(*string_exp*) ODBC 3.0	The length in bits of the string expression.
CHAR(*code*)	The character with the ASCII code value specified by *code*. *code* should be between 0 and 255; otherwise, the return value is data-source dependent.
CHAR_LENGTH(*string_exp*) ODBC 3.0	The length in characters of the string expression, if the string expression is of a character data type; otherwise, the length in bytes of the string expression (the smallest integer not less than the number of bits divided by 8). (This function is the same as the CHARACTER_LENGTH function.)
CHARACTER_LENGTH(*string_exp*) ODBC 3.0	The length in characters of the string expression, if the string expression is of a character data type; otherwise, the length in bytes of the string expression (the smallest integer not less than the number of bits divided by 8). (This function is the same as the CHAR_LENGTH function.)
CONCAT(*string_exp1*,*string_exp*)	The string resulting from concatenating *string_exp2* and *string_exp1*. The string is system dependent.
DIFFERENCE(*string_exp2 and string_exp1*)	An integer value that indicates the difference between the values returned by the SOUNDEX function for *string_exp2* and *string_exp1*.
INSERT(*string_exp1 , start, length, string_exp2*)	A string where *length* characters have been deleted from *string_exp1* beginning at *start* and where *string_exp2* has been inserted into *string_exp*, beginning at *start*.
LCASE(*string_exp*)	Uppercase characters in *string_exp* converted to lowercase.
LEFT(*string_exp, count*)	The *count* of characters of *string_exp*.

Table 3–1: Scalar String Functions *(2 of 3)*

Function	Returns
LENGTH(*string_exp*)	The number of characters in *string_exp*, excluding trailing blanks and the string termination character.
LOCATE(*string_exp1, string_exp2* [,*start*,])	The starting position of the first occurrence of *string_exp1* within *string_exp2*. If *start* is not specified the search begins with the first character position in *string_exp2*. If start is specified, the search begins with the character position indicated by the value of *start*. The first character position in *string_exp2* is indicated by the value 1. If *string_exp1* is not found, 0 is returned.
LTRIM(*string_exp*)	The characters of *string_exp*, with leading blanks removed.
OCTET_LENGTH(*string_exp*) ODBC 3.0	The length in bytes of the string expression. The result is the smallest integer not less than the number of bits divided by 8.
POSITION(*character_exp* IN *character_exp*)	The position of the first character expression in the second character expression. The result is an exact numeric with an implementation-defined precision and a scale of 0.
REPEAT(*string_exp, count*)	A string composed of *string_exp* repeated *count* times.
REPLACE(*string_exp1*, *string_exp2*, *string_exp3*)	Replaces all occurrences of *string_exp2* in *string_exp1* with *string_exp3*.
RIGHT(*string_exp, count*)	The rightmost *count* of characters in *string_exp*.
RTRIM(*string_exp*)	The characters of *string_exp* with trailing blanks removed.
SPACE(*count*)	A string consisting of *count* spaces.

Table 3–1: Scalar String Functions *(3 of 3)*

Function	Returns
SUBSTRING(*string_exp , start, length*)	A string derived from *string_exp* beginning at the character position *start* for *length* characters.
UCASE(*string_exp*)	Lowercase characters in *string_exp* converted to uppercase.

3.4 Numeric Functions

Table 3–2 lists the numeric functions that ODBC supports.

The numeric functions listed can take the following arguments:

- *numeric_exp* can be a column name, a numeric literal, or the result of another scalar function, where the underlying data type is SQL_NUMERIC, SQL_DECIMAL, SQL_TINYINT, SQL_SMALLINT, SQL_INTEGER, SQL_FLOAT, SQL_REAL, or SQL_DOUBLE.
- *float_exp* can be a column name, a numeric literal, or the result of another scalar function, where the underlying data type is SQL_FLOAT.
- *integer_exp* can be a column name, a numeric literal, or the result of another scalar function, where the underlying data type is SQL_TINYINT, SQL_SMALLINT, or SQL_INTEGER.

Table 3–2: Scalar Numeric Functions *(1 of 3)*

Function	Returns
ABS(*numeric_exp*)	Absolute value of *numeric_exp*.
ACOS(*float_exp*)	Arccosine of *float_exp* as an angle in radians.
ASIN(*float_exp*)	Arcsine of *float_exp* as an angle in radians.
ATAN(*float_exp*)	Arctangent of *float_exp* as an angle in radians.

Table 3–2: Scalar Numeric Functions *(2 of 3)*

Function	Returns
ATAN2(*float_exp1*, *float_exp2*)	Arctangent of the x and y coordinates, specified by *float_exp1* and *float_exp2* as an angle in radians.
CEILING(*numeric_exp*)	Smallest integer greater than or equal to *numeric_exp* .
COS(*float_exp*)	Cosine of *float_exp* as an angle in radians.
COT(*float_exp*)	Cotangent of *float_exp* as an angle in radians.
DEGREES(*numeric_exp*)	Number of degrees converted from *numeric_exp* radians.
EXP(*float_exp*)	Exponential value of *float_exp*.
FLOOR(*numeric_exp*)	Largest integer less than or equal to *numeric_exp*.
LOG(*float_exp*)	Natural log of *float_exp*
LOG10(*float_exp*)	Base 10 log of *float_exp*
MOD(*integer_exp1*, *integer_exp2*)	Remainder of *integer_exp1* divided by *integer_exp2*
PI()	Constant value of pi as a floating-point number.
POWER(*numeric_exp*, *integer_exp)*	Value of *numeric_exp* to the power of *integer_exp*.
RADIANS(*numeric_exp*)	Number of radians converted from *numeric_exp* degrees.
RAND([*integer_exp*])	Random floating-point value using *integer_exp* as the optional seed value.
ROUND(*numeric_exp*, *integer_exp*)	*numeric_exp* rounded to *integer_exp* places right of the decimal (left of the decimal if *integer_exp* is negative).

Table 3–2: Scalar Numeric Functions *(3 of 3)*

Function	Returns
SIGN(*numeric_exp*)	Indicator of the sign of *numeric_exp*. If *numeric_exp* <0, -1 is returned. If *numeric_exp* = 0, 0 is returned. If *numeric_exp* > 0, 1 is returned.
SIN(*float_exp*)	Sine of *float_exp*, where *float_exp* is an angle in radians.
SQRT(*float_exp*)	Square root of *float_exp*.
TAN(*float_exp*)	Tangent of *float_exp*, where *float_exp* is an angle in radians.
TRUNCATE(*numeric_exp*, *integer_exp*)	*numeric_exp* truncated to *integer_exp* places right of the decimal. (If *integer_exp* is negative, truncation is to the left of the decimal.)

3.5 Date and Time Functions

Table 3–3 lists the date and time functions that ODBC supports.

The date and time functions listed can take the following arguments:

- *date-exp* can be a column name, a date or timestamp literal, or the result of another scalar function, where the underlying data type can be represented as SQL_CHAR, SQL_VARCHAR, SQL_DATE, or SQL_TIMESTAMP.
- *time_exp* can be a column name, a timestamp or timestamp literal, or the result of another scalar function, where the underlying data type can be represented as SQL_CHAR, SQL_VARCHAR, SQL_TIME, or SQL_TIMESTAMP.
- *timestamp_exp* can be a column name; a time, date, or timestamp literal; or the result of another scalar function, where the underlying data type can be represented as SQL_CHAR, SQL_VARCHAR, SQL_TIME, SQL_DATE, or SQL_TIMESTAMP.

Table 3–3: Scalar Time and Date Functions *(1 of 2)*

Function	Returns
CURRENT DATE() (ODBC 3.6)	Current date.
CURRENT_TIME[(*time-precision*)] (ODBC 3.6)	Current local time. The *time-precision* argument determines the seconds precision of the returned value.
CURRENT_TIMESTAMP [(*timestamp-precision*)] (ODBC 3.6)	Current local date and local time as a timestamp value. The *timestamp-precision* argument determines the seconds precision of the returned timestamp.
CURDATE()	Current date as a date value.
CURTIME()	Current local time as a time value.
DAYNAME(*date_exp*)	Character string containing a date source-specific name of the day for the day portion of *date_exp*.
DAYOFMONTH(*date_exp*)	Day of the month in *date_exp* as an integer value (1-31).
DAYOFWEEK(*date_exp*)	Day of the week in *date_exp* as an integer value (1-7).
DAYOFYEAR(*date_exp*)	Day of the year in *date_exp* as an integer value (1-366).
HOUR(*time_exp*)	Hour in *time_exp* as an integer value (0-23).
MINUTE(*time_exp*)	Minute in *time_exp* as an integer value (0-59).
MONTH(*date_exp*)	Month in *date_exp* as an integer value (1-366).
MONTHNAME(*date_exp*)	Character string containing the data source-specific name of the month.
NOW()	Current date and time as a timestamp value.

Table 3–3: Scalar Time and Date Functions *(2 of 2)*

Function	Returns
QUARTER(*date_exp*)	Quarter in *date_exp* as an integer value (1-4).
SECOND(*time_exp*)	Second in *date_exp* as an integer value (0-59).
TIMESTAMPADD(*interval*, *integer_exp*, *time_exp*)	Timestamp calculated by adding *integer_exp* intervals of type *interval* to *time_exp*. *interval* can be: SQL_TSI_FRAC_SECOND SQL_TSI_SECOND SQL_TSI_MINUTE SQL_TSI_HOUR SQL_TSI_DAY SQL_TSI_WEEK SQL_TSI_MONTH SQL_TSI_QUARTER SQL_TSI_YEAR Fractional seconds are expressed in billionths of a second.
TIMESTAMPDIFF(*interval*, *time_exp1*, *time_exp2*)	Integer number of intervals of type *interval* by which *time_exp2* is greater than *time_exp1*. *interval* has the same values as TIMESTAMPADD. Fractional seconds are expressed in billionths of a second.
WEEK(*date_exp*)	Week of the year in *date_exp* as an integer value (1-53).
YEAR(*date_exp*)	Year in *date_exp*. The range is data source dependent.

3.6 System Functions

Table 3–4 lists the system functions that ODBC supports.

Table 3–4: Scalar System Functions

Function	Returns
DATABASE()	Name of the database, corresponding to the connection handle (*hdbc*).
IFNULL(*exp*, *value*)	*value*, if *exp* is null.
ROWID(*extension*)	The row identifier of the current row in a table.
USER()	Authorization name of the user.

4

The UNIX Environment

This chapter describes how to configure the Progress SQL-92 ODBC Driver in the UNIX environment.

4.1 Configuring the ODBC Driver

On UNIX, the Progress SQL-92 ODBC Driver is a series of MERANT-supplied shared libraries. In order to configure the ODBC Driver on a UNIX system, the Progress SQL-92 ODBC Driver libraries and executable files must have been installed, as described in the *Progress Installation and Configuration Guide Version 9 for UNIX*. Table 4–1 lists the location of the driver libraries and executable files and the name of the Progress SQL-92 ODBC Driver Manager library by platform.

Table 4–1: ODBC Driver File Location by Platform

Platform	Location of Files	Driver Manager Library
Solaris	`$DLC/odbc/lib`	`libodbc.so`
IBM AIX	`$DLC/odbc/lib`	`libodbc.a`
Hewlett Packard	`$DLC/odbc/lib`	`libodbc.sl`

NOTE: $DLC is an environment variable that points to the directory in which Progress is installed. Using $DLC allows you to use the same script to set environment variables relative to where Progress is installed, regardless of where it is installed on a particular system.

To configure the driver you must set the appropriate environment variables and configure your data source. The following sections contain instructions for completing these tasks.

4.1.1 Setting Environment Variables

Table 4–2 lists the environment variables you must set to configure the ODBC driver and the location to which you must set them (the directory containing the ODBC Driver libraries), by platform.

Table 4–2: ODBC Driver Environment Variables

Platform	Environment Variable	Location
Solaris	LD_LIBRARY_PATH	`$DLC/odbc/lib`
IBM AIX	LIBPATH	`$DLC/odbc/lib`
Hewlett Packard	SHLIB_PATH	`$DLC/odbc/lib`

4.1.2 Configuring A Progress Data Source

In the UNIX environment, there is no ODBC Administrator. To configure a data source, you must edit the system information file, a plain text file that is normally located in the $HOME directory and is usually called `.odbc.ini`.

4.1.3 The System Information File (.odbc.ini)

The system information file is maintained using any text editor to define data source entries as described in Chapter 2, "Configuring Progress Data Sources." A sample `odbc.ini` file is located in $DLC/odbc.

4.1.4 Sample Solaris System Information File

In the following sample, *xx* represents the driver number:

```
[ODBC Data Sources]
results=Progress_SQL92_Driver

[results]
Driver=/dlc/odbc/lib/pgproxx.so
DatabaseName=testdb
PortNumber=sqlservicename
HostName=myserverhost
LogonID=mylogin
Password=xxx
APILevel=1
ConnectFunctions=YYN
CPTimeout=60
DriverODBCVer=03.60
FileUsage=0
SQLLevel=0
UsageCount=1
ArraySize=50
DefaultIsolationLevel=REPEATABLE READ
LogonID=ccall
StaticCursorLongColBuffLen=4096

[ODBC]
InstallDir=/dlc/odbc
Trace=0
TraceFile=odbctrace.out
TraceDll=/dlc/odbc/lib/odbctrac.so
UseCursorLib=0
```

The library file names in the system information file vary by platform. Table 4–3 lists the Driver and Tracing library names by platform.

Table 4–3: System Information File Entries by Platform

Platform	Driver Library Name	Tracing Library Name
Solaris	pgpro915.so	odbctrac.so
IBM AIX	pgpro915.a	odbctrac.a
Hewlett Packard	pgpro915.sl	odbctrac.sl

4.1.5 The ODBCINI Environment Variable

UNIX support of the ODBC driver permits the use of a centralized system information file that a system administrator can control. This is accomplished by setting the environment variable ODBCINI to point to the fully qualified pathname of the centralized file. For example, in the C shell you could set this variable as follows:

```
setenv ODBCINI /dlc/odbc/odbc.ini
```

In the Bourne or Korn shell, you would set it as:

```
ODBCINI=/dlc/odbc/system_odbc.ini;export ODBCINI
```

ODBCINI is an environment variable that the Progress SQL-92 ODBC driver will recognize. You can set ODBCINI to use a system information file other than the default file. You can have multiple system information files and change the ODBCINI environment variable to point to the one you want to use.

The search order for the location of the system information file is as follows:

1. Check ODBCINI
2. Check $HOME for .odbc.ini

There must be an [ODBC] section in the system information file that includes the InstallDir keyword. The value of this keyword must be the path to the directory under which the `/lib` and `/messages` directories are contained. For example, if you choose the default install directory, then the following line must be in the [ODBC] section:

```
InstallDir=/dlc/odbc
```

4.2 Running an ODBC Client Application on UNIX

In order to run an ODBC client application on UNIX you must set the system shared library path to include the directory containing the ODBC driver `DLC/odbc/lib` as follows in Table 4–4:

Table 4–4: Runtime System Shared Library Path Settings

Platform	System Shared Library Path Setting
Solaris	`LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$DLC/odbc/lib`
IBM AIX	`LIBPATH=$LIBPATH:$DLC/odbc/lib`
Hewlett Packard	`SHLIB_PATH=$SHLIB_PATH:DLC/odbc/lib`

Setting the appropriate library search path environment variable (LD_LIBRARY_PATH on Solaris, SHLIB_PATH on HP/UX, or LIBPATH on AIX) enables the ODBC application to locate the ODBC core components and drivers at the time of execution.

Alternatively, you can set the ODBC environment variable and then run the script `$DLC/bin/slibod_env`:

```
$ DLC/bin/proenv
. $DLC/bin/slibod_env
```

4.3 Using Double-Byte Character Sets

The Progress SQL-92 ODBC driver is capable of using double-byte character sets. The driver normally uses the character set defined by the default locale "C" unless explicitly pointed to another character set. The default locale "C" corresponds to the 7-bit ASCII character set in which only characters from ISO 8859-1 are valid. Use the following procedure to set the locale to a different character set:

1 ♦ Add the following line at the very beginning of applications that use double-byte character sets:

```
setlocale (LC_ALL, "");
```

This is a standard UNIX function. It selects the character set indicated by the environment variable LANG as the one to be used by X/Open compliant character handling functions. If this line is not present, or if LANG is either not set or is set to NULL, the default locale "C" is used.

2 ♦ Set the LANG environment variable to the appropriate character set. The UNIX command `locale -a` can be used to display all supported character sets on your system.

For more information, see the man pages for “locale” and “setlocale.”

4.4 Translators

Progress provides a sample translator named “OEM to ANSI” that provides a framework for coding a translation library.

Table 4–5 lists the Translation keyword options. You must add the TranslationSharedLibrary keyword to the data source section of the system information file to perform a translation. Adding the TranslationOption keyword is optional.

Table 4–5: Translators

Keyword	Definition
TranslationSharedLibrary	Full path of translation library
TranslationOption	ASCII representation of the 32-bit integer translation option

A

SQLGetInfo

This appendix shows information the Progress SQL-92 ODBC Driver returns to SQLGetInfo.

Table A–1 describes return values to SQLGetInfo from the Progress SQL-92 ODBC driver.

Table A–1: Information the Progress SQL-92 ODBC Driver Returns to SQLGetInfo *(1 of 27)*

Description	*fInfoType* Argument	Returns
Guaranteed execute privileges on all procedures returned by SQLProcedures	SQL_ACCESSIBLE _PROCEDURES	N
Guaranteed read access to all table names returned by SQLTables	SQL_ACCESSIBLE_TABLES	N
Maximum number of active connections	SQL_ACTIVE_ CONNECTIONS	0
Maximum number of active statements supported for an active connection	SQL_ACTIVE_STATEMENTS	0
Maximum number of active environments	SQL_ACTIVE_ ENVIRONMENTS	0
Support for ALTER DOMAIN statement	SQL_ALTER_DOMAIN	0x00000000
Support for ALTER TABLE clauses	SQL_ALTER_TABLE	0x00000000
SQL Conformance	SQL_SQL_CONFORMANCE	SQL_SC_SQL92_ENTRY
Support for datetime literals	SQL_DATETIME_LITERALS	0x00000000
Level of asynchronous mode support	SQL_ASYNC_MODE	SQL_AM_NONE

Table A–1: Information the Progress SQL-92 ODBC Driver Returns to SQLGetInfo *(2 of 27)*

Description	*fInfoType* Argument	Returns
Behavior with respect to the availability of row counts in batches	SQL_BATCH_ROW_COUNT	0x00000000
Support for batches	SQL_BATCH_SUPPORT	0x00000000
Support for bookmarks	SQL_BOOKMARK_ PERSISTENCE	SQL_BP_UPDATE SQL_BP_SCROLL
Position of qualifier in a qualified table name	SQL_CATALOG_LOCATION	0 (does not support)
Support for catalog names	SQL_CATALOG_NAME	N
Character used to separate table, column qualifiers	SQL_CATALOG_NAME_ SEPARATOR	“ ”
Term for object that qualifies table names	SQL_CATALOG_TERM	“ ”
Statements that support qualifiers	SQL_CATALOG_USAGE	0x00000000
Default collation sequence name for the default character set	SQL_COLLATION_SEQ	“ ”
Support for column aliases	SQL_COLUMN_ALIAS	Y
Result of concatenation of NULL character column with non-NULL column	SQL_CONCAT_NULL_ BEHAVIOR	SQL_CB_NULL = 0

Table A–1: Information the Progress SQL-92 ODBC Driver Returns to SQLGetInfo *(3 of 27)*

Description	*fInfoType* Argument	Returns
Conversion from BIGINT	SQL_CONVERT_BIGINT	SQL_CVT_CHAR SQL_CVT_BIGINT SQL_CVT_TINYINT SQL_CVT_SMALLINT SQL_CVT_INTEGER SQL_CVT_FLOAT SQL_CVT_DOUBLE
Conversion from BINARY	SQL_CONVERT_BINARY	0x00000000
Conversion from BIT	SQL_CONVERT_BIT	0x00000000
Conversion from CHAR	SQL_CONVERT_CHAR	SQL_CVT_CHAR SQL_CVT_NUMERIC SQL_CVT_DECIMAL SQL_CVT_INTEGER SQL_CVT_SMALLINT SQL_CVT_FLOAT SQL_CVT_REAL SQL_CVT_DOUBLE SQL_CVT_VARCHAR SQL_CVT_TINYINT SQL_CVT_BIGINT SQL_CVT_DATE SQL_CVT_TIME SQL_CVT_TIMESTAMP
Conversion from DATE	SQL_CONVERT_DATE	SQL_CVT_CHAR SQL_CVT_VARCHAR SQL_CVT_DATE SQL_CVT_TIMESTAMP

Table A–1: Information the Progress SQL-92 ODBC Driver Returns to SQLGetInfo *(4 of 27)*

Description	*fInfoType* Argument	Returns
Conversion from DECIMAL	SQL_CONVERT_DECIMAL	SQL_CVT_CHAR SQL_CVT_NUMERIC SQL_CVT_DECIMAL SQL_CVT_INTEGER SQL_CVT_SMALLINT SQL_CVT_FLOAT SQL_CVT_REAL SQL_CVT_DOUBLE SQL_CVT_VARCHAR SQL_CVT_TINYINT SQL_CVT_BIGINT
Conversion from DOUBLE	SQL_CONVERT_DOUBLE	SQL_CVT_CHAR SQL_CVT_NUMERIC SQL_CVT_DECIMAL SQL_CVT_INTEGER SQL_CVT_SMALLINT SQL_CVT_FLOAT SQL_CVT_REAL SQL_CVT_DOUBLE SQL_CVT_VARCHAR SQL_CVT_TINYINT SQL_CVT_BIGINT
Conversion from FLOAT	SQL_CONVERT_FLOAT	SQL_CVT_CHAR SQL_CVT_NUMERIC SQL_CVT_DECIMAL SQL_CVT_INTEGER SQL_CVT_SMALLINT SQL_CVT_FLOAT SQL_CVT_REAL SQL_CVT_DOUBLE SQL_CVT_VARCHAR SQL_CVT_TINYINT SQL_CVT_BIGINT
Support for conversion functions	SQL_CONVERT_FUNCTIONS	SQL_FN_CVT_CONVERT

Table A–1: Information the Progress SQL-92 ODBC Driver Returns to SQLGetInfo *(5 of 27)*

Description	*fInfoType* Argument	Returns
Conversion from INTEGER	SQL_CONVERT_INTEGER	SQL_CVT_CHAR SQL_CVT_NUMERIC SQL_CVT_DECIMAL SQL_CVT_INTEGER SQL_CVT_SMALLINT SQL_CVT_FLOAT SQL_CVT_REAL SQL_CVT_DOUBLE SQL_CVT_VARCHAR SQL_CVT_TINYINT SQL_CVT_BIGINT
Conversion from INTERVAL_DAY_TIME	SQL_CONVERT_INTERVAL_DAY_TIME	0x00000000
Conversion from INTERVAL_YEAR_MONTH	SQL_CONVERT_INTERVAL_YEAR_MONTH	0x00000000
Conversion from LONGVARBINARY	SQL_CONVERT_LONGVARBINARY	0x00000000
Conversion from LONGVARCHAR	SQL_CONVERT_LONGVARCHAR	0x00000000
Conversion from NUMERIC	SQL_CONVERT_NUMERIC	SQL_CVT_CHAR SQL_CVT_NUMERIC SQL_CVT_DECIMAL SQL_CVT_INTEGER SQL_CVT_SMALLINT SQL_CVT_FLOAT SQL_CVT_REAL SQL_CVT_DOUBLE SQL_CVT_VARCHAR SQL_CVT_TINYINT SQL_CVT_BIGINT

Table A–1: Information the Progress SQL-92 ODBC Driver Returns to SQLGetInfo *(6 of 27)*

Description	*fInfoType* Argument	Returns
Conversion from REAL	SQL_CONVERT_REAL	SQL_CVT_CHAR SQL_CVT_NUMERIC SQL_CVT_DECIMAL SQL_CVT_INTEGER SQL_CVT_SMALLINT SQL_CVT_FLOAT SQL_CVT_REAL SQL_CVT_DOUBLE SQL_CVT_VARCHAR SQL_CVT_TINYINT SQL_CVT_BIGINT
Conversion from SMALLINT	SQL_CONVERT_SMALLINT	SQL_CVT_CHAR SQL_CVT_NUMERIC SQL_CVT_DECIMAL SQL_CVT_INTEGER SQL_CVT_SMALLINT SQL_CVT_FLOAT SQL_CVT_REAL SQL_CVT_DOUBLE SQL_CVT_VARCHAR SQL_CVT_TINYINT SQL_CVT_BIGINT
Conversion from TIME	SQL_CONVERT_TIME	SQL_CVT_CHAR SQL_CVT_TIME SQL_CVT_TIMESTAMP
Conversion from TIMESTAMP	SQL_CONVERT_TIMESTAMP	SQL_CVT_CHAR SQL_CVT_VARCHAR SQL_CVT_DATE SQL_CVT_TIME SQL_CVT_TIMESTAMP

Table A–1: Information the Progress SQL-92 ODBC Driver Returns to SQLGetInfo *(7 of 27)*

Description	*fInfoType* Argument	Returns
Conversion from TINYINT	SQL_CONVERT_TINYINT	SQL_CVT_CHAR SQL_CVT_NUMERIC SQL_CVT_DECIMAL SQL_CVT_INTEGER SQL_CVT_SMALLINT SQL_CVT_FLOAT SQL_CVT_REAL SQL_CVT_DOUBLE SQL_CVT_VARCHAR SQL_CVT_TINYINT SQL_CVT_BIGINT
Conversion from VARBINARY	SQL_CONVERT_ VARBINARY	0x00000000
Conversion from VARCHAR	SQL_CONVERT_VARCHAR	SQL_CVT_CHAR SQL_CVT_NUMERIC SQL_CVT_DECIMAL SQL_CVT_INTEGER SQL_CVT_SMALLINT SQL_CVT_FLOAT SQL_CVT_REAL SQL_CVT_DOUBLE SQL_CVT_VARCHAR SQL_CVT_TINYINT SQL_CVT_BIGINT SQL_CVT_DATE SQL_CVT_TIME SQL_CVT_TIMESTAMP
Conversion from WCHAR	SQL_CONVERT_WCHAR	0x00000000
Conversion from WLONGVAR CHAR	SQL_CONVERT_ WLONGVARCHAR	0x00000000
Conversion from WVARCHAR	SQL_CONVERT_WVARCHAR	0x00000000
Support for table correlation names	SQL_CORRELATION_NAME	SQL_CN_DIFFERENT

Table A–1: Information the Progress SQL-92 ODBC Driver Returns to SQLGetInfo *(8 of 27)*

Description	*fInfoType* Argument	Returns
Support for CREATE ASSERTION statement	SQL_CREATE_ASSERTION	0x00000000
Support for CREATE CHARACTER SET statement	SQL_CREATE_ CHARACTER SET	0x00000000
Support for CREATE COLLATION statement	SQL_CREATE_COLLATION	0x00000000
Support for CREATE DOMAIN statement	SQL_CREATE_DOMAIN	0x00000000
Support for CREATE SCHEMA statement	SQL_CREATE_SCHEMA	0x00000000
Support for CREATE TABLE statement	SQL_CREATE_TABLE	SQL_CT_CREATE_ TABLE SQL_CT_COLUMN _CONSTRAINT SQL_CT_TABLE _CONSTRAINT
Support for CREATE TRANSLATION statement	SQL_CREATE_ TRANSLATION	0x00000000
Support for CREATE VIEW statement	SQL_CREATE_VIEW	SQL_CV_CREATE_VIEW SQL_CV_CHECK_OPTION

Table A–1: Information the Progress SQL-92 ODBC Driver Returns to SQLGetInfo *(9 of 27)*

Description	*fInfoType* Argument	Returns
Effect of COMMIT operation on cursors and prepared statements	SQL_CURSOR_COMMIT_ BEHAVIOR	SQL_CB_PRESERVE
Effect of ROLLBACK operation on cursors and prepared statements	SQL_CURSOR_ROLLBACK _BEHAVIOR	0x00000000
Support for cursor sensitivity	SQL_CURSOR_SENSITIVITY	SQL_INSENSITIVE
Name of the data source as specified to the ODBC Administrator	SQL_DATA_SOURCE_NAME	(string containing the name)
Access limited to read-only	SQL_DATA_SOURCE_READ_ ONLY	N (read-write access)
Name of the Progress SQL-92 ODBC data source on the server system	SQL_DATABASE_NAME	(string containing the name)
Name of the database product supporting the data source	SQL_DBMS_NAME	PROGRESS
Version of the database product	SQL_DBMS_VER	09.1A
Default transaction isolation level	SQL_DEFAULT_TXN_ ISOLATION	SQL_TXN_REPEATABLE_ READ

Table A–1: Information the Progress SQL-92 ODBC Driver Returns to SQLGetInfo *(10 of 27)*

Description	*fInfoType* Argument	Returns
Support for describing parameters via DESCRIBE INPUT statement	SQL_DESCRIBE_ PARAMETER	Y (supports)
Version of the driver manager	SQL_DM_VER	03.51.37110.0000
Connection handle determined by the argument InfoType	SQL_DRIVER_HDBC	0x0013EBE0
Driver's descriptor handle determined by the Driver Manager's descriptor handle	SQL_DRIVER_HDESC	0x0013F820
Environment handle determined by the argument InfoType	SQL_DRIVER_HENV	0x0013C8A0
Handle library from the load library returned to the Driver Manager when it loaded the driver DLL	SQL_DRIVER_HLIB	0x28660000
Driver's statement handle determined by the Driver Manager's statement handle	SQL_DRIVER_HSTMT	0x001427E0
Name of the dynamic link library file for the ODBC Driver	SQL_DRIVER_NAME	PGPRO915.DLL

Table A–1: Information the Progress SQL-92 ODBC Driver Returns to SQLGetInfo *(11 of 27)*

Description	*fInfoType* Argument	Returns
Supported ODBC version	SQL_DRIVER_ODBC_VER	03.51
Current version of the ODBC Driver	SQL_DRIVER_VER	03.60.0000
Support for DROP ASSERTION statement	SQL_DROP_ASSERTION	0x00000000
Support for DROP CHARACTER SET statement	SQL_DROP_CHARACTER_ SET	0x00000000
Support for DROP COLLATION statement	SQL_DROP_COLLATION	0x00000000
Support for DROP DOMAIN statement	SQL_DROP_DOMAIN	0x00000000
Support for DROP SCHEMA statement	SQL_DROP_SCHEMA	0x00000000
Support for DROP TABLE statement	SQL_DROP_TABLE	SQL_DT_DROP_TABLE
Support for DROP TRANSLATION statement	SQL_DROP_TRANSLATION	0x00000000
Support for DROP VIEW statement	SQL_DROP_VIEW	SQL_DT_DROP_VIEW
Supported attributes of a dynamic cursor: subset 1	SQL_DYNAMIC_CURSOR_ ATTRIBUTES1	0x00000000

Table A–1: Information the Progress SQL-92 ODBC Driver Returns to SQLGetInfo *(12 of 27)*

Description	*fInfoType* Argument	Returns
Supported attributes of a dynamic cursor: subset 2	SQL_DYNAMIC_CURSOR_ ATTRIBUTES2	0x00000000
Support for expressions in ORDER BY clause	SQL_EXPRESSIONS_IN_ ORDERBY	Y
Supported fetch direction options	SQL_FETCH_DIRECTION	SQL_FD_FETCH_NEXT SQL_FD_FETCH_FIRST SQL_FD_FETCH_LAST SQL_FD_FETCH_PRIOR SQL_FD_FETCH_ ABSOLUTE SQL_FD_FETCH_ RELATIVE SQL_FD_FETCH_ BOOKMARK
Single-tier driver behavior	SQL_FILE_USAGE	SQL_FILE_NOT _SUPPORTED
Supported attributes of a forward-only cursor: subset 1	SQL_FORWARD_ONLY_ CURSOR_ATTRIBUTES1	SQL_CA1_NEXT SQL_CA1_BULK_ADD
Supported attributes of a forward-only cursor: subset 2	SQL_FORWARD_ONLY_ CURSOR_ATTRIBUTES2	SQL_CA2_MAX_ROWS_ SELECT SQL_CA2_MAX_ROWS_ CATALOG
Supported extensions to SQLGetData	SQL_GETDATA_ EXTENSIONS	SQL_GD_ANY_COLUMN SQL_GD_ANY_ORDER SQL_GD_BOUND
Relationship between GROUP BY clause and columns in the select list	SQL_GROUP_BY	SQL_GB_GROUP_BY _CONTAINS_SELECT

Table A–1: Information the Progress SQL-92 ODBC Driver Returns to SQLGetInfo *(13 of 27)*

Description	*fInfoType* Argument	Returns
Case sensitivity of user-supplied names	SQL_IDENTIFIER_CASE	SQL_IC_UPPER
Character used to enclose delimited identifiers	SQL_IDENTIFIER_QUOTE_ CHAR	" (double quotation mark)
Keywords supported in the CREATE INDEX statement	SQL_INDEX_KEYWORDS	SQL_IK_ASC | SQL_IK_DESC
Supported views in INFORMATION_ SCHEMA	SQL_INFO_SCHEMA_VIEWS	0x00000000
Support for Integrity Enhancement Facility	SQL_INTEGRITY	Y
Supported attributes of a keyset cursor: subset 1	SQL_KEYSET_CURSOR_ ATTRIBUTES1	SQL_CA1_NEXT SQL_CA1_ABSOLUTE SQL_CA1_RELATIVE SQL_CA1_BOOKMAR K SQL_CA1_LOCK_NO_ CHANGE SQL_CA1_POS_POSITION SQL_CA1_POS_UPDATE SQL_CA1_POS_DELETE SQL_CA1_POS_REFRESH SQL_CA1_POSITIONED _UPDATE SQL_CA1_POSITIONED _DELETE SQL_CA1_SELECT_FOR _UPDATE SQL_CA1_BULK_ADD

Table A–1: Information the Progress SQL-92 ODBC Driver Returns to SQLGetInfo *(14 of 27)*

Description	*fInfoType* Argument	Returns
Supported attributes of a keyset cursor: subset 2	SQL_KEYSET_CURSOR_ ATTRIBUTES2	SQL_CA2_READ_ONLY_ CONCURRENCY SQL_CA2_OPT_VALUES _CONCURRENCY SQL_CA2_SENSITIVITY _DELETIONS SQL_CA2_SENSITIVITY _UPDATES SQL_CA2_MAX_ROWS _SELECT SQL_CA2_CRC_EXACT SQL_CA2_SIMULATE _TRY_UNIQUE
Data source specific keywords	SQL_KEYWORDS	See the *Progress SQL-92 Guide and Reference* for a list of SQL Keywords.
Support for escape clause in LIKE predicates	SQL_LIKE_ESCAPE_CLAUSE	Y
Support for lock types	SQL_LOCK_TYPES	SQL_LCK_NO_CHANGE
Maximum number of active concurrent statements in asynchronous mode	SQL_MAX_ASYNC _CONCURRENT_ STATEMENTS	0
Maximum length in hexadecimal characters of binary literals	SQL_MAX_BINARY_ LITERAL_LEN	31995
Maximum length of a table or column qualifier	SQL_MAX_CATALOG_NAME _LEN	0

Table A–1: Information the Progress SQL-92 ODBC Driver Returns to SQLGetInfo *(15 of 27)*

Description	*fInfoType* Argument	Returns
Maximum length in characters of character string literals	SQL_MAX_CHAR_LITERAL_LEN	31995
Maximum length of a column name	SQL_MAX_COLUMN_NAME_LEN	32
Maximum number of columns allowed in GROUP BY clause	SQL_MAX_COLUMNS_IN_GROUP_BY	499
Maximum number of columns allowed in an index	SQL_MAX_COLUMNS_IN_INDEX	16
Maximum number of columns allowed in ORDER BY clause	SQL_MAX_COLUMNS_IN_ORDER_BY	0
Maximum number of columns allowed in a select list	SQL_MAX_COLUMNS_IN_SELECT	500
Maximum number of columns allowed in a table	SQL_MAX_COLUMNS_IN_TABLE	500
Maximum number of active SQL statements	SQL_MAX_CONCURRENT_ACTIVITIES	0 (no maximum)
Maximum length of a cursor name	SQL_MAX_CURSOR_NAME_LEN	18
Maximum number of active connections	SQL_MAX_DRIVER_CONNECTIONS	0

Table A–1: Information the Progress SQL-92 ODBC Driver Returns to SQLGetInfo *(16 of 27)*

Description	*fInfoType* Argument	Returns
Maximum length of user-defined names	SQL_MAX_IDENTIFIER_LEN	32
Maximum number of bytes allowed in the combined fields of an index	SQL_MAX_INDEX_SIZE	115
Maximum length of a procedure name	SQL_MAX_PROCEDURE_ NAME_LEN	32
Maximum length in bytes of a table row	SQL_MAX_ROW_SIZE	0
Whether maximum row size includes LONGVARCHAR and LONGVARBINARY	SQL_MAX_ROW_SIZE _INCLUDES_LONG	N
Maximum length of an owner name	SQL_MAX_SCHEMA_NAME _LEN	32
Maximum number of characters in an SQL statement	SQL_MAX_STATEMENT_ LEN	131000
Maximum length of a table name	SQL_MAX_TABLE_NAME_ LEN	32
Maximum number of tables allowed in FROM clause	SQL_MAX_TABLES_IN_ SELECT	250
Maximum length of a user name	SQL_MAX_USER_NAME_ LEN	32
Maximum length of owner name	SQL_MAX_OWNER_NAME_ LEN	32

Table A–1: Information the Progress SQL-92 ODBC Driver Returns to SQLGetInfo *(17 of 27)*

Description	*fInfoType* Argument	Returns
Maximum length of a qualifier name	SQL_MAX_QUALIFIER_ NAME_LEN	0
Support for multiple result sets	SQL_MULT_RESULT_SETS	N
Support for active transactions on multiple connections	SQL_MULTIPLE_ACTIVE _TXN	Y
Whether data source requires length of LONGVARCHAR and LONGVARBINA RY data	SQL_NEED_LONG_DATA _LEN	Y
Support for NOT NULL clause in CREATE TABLE statement	SQL_NON_NULLABLE _COLUMNS	SQL_NNC_NON_NULL
Where null values are sorted in a list	SQL_NULL_COLLATION	SQL_NC_LOW

Table A–1: Information the Progress SQL-92 ODBC Driver Returns to SQLGetInfo *(18 of 27)*

Description	*fInfoType* Argument	Returns
Numeric functions supported	SQL_NUMERIC_FUNCTIONS	SQL_FN_NUM_ABS SQL_FN_NUM_ACOS SQL_FN_NUM_ASIN SQL_FN_NUM_ATAN SQL_FN_NUM_ATAN2 SQL_FN_NUM_CEILING SQL_FN_NUM_COS SQL_FN_NUM_EXP SQL_FN_NUM_FLOOR SQL_FN_NUM_MOD SQL_FN_NUM_PI SQL_FN_NUM_POWER SQL_FN_NUM_RADIANS SQL_FN_NUM_RAND SQL_FN_NUM_ROUND SQL_FN_NUM_SIGN SQL_FN_NUM_SIN SQL_FN_NUM_SQRT SQL_FN_NUM_TAN SQL_FN_NUM_DEGREES SQL_FN_NUM_LOG10
Level of ODBC conformance	SQL_ODBC_API _CONFORMANCE	SQL_OAC_LEVEL1
Level of ODBC 3.x interface conformance	SQL_ODBC_INTERFACE _CONFORMANCE	SQL_OIC_CORE
SQL Access Group (SAG) conformance	SQL_ODBC_SAG_CLI _CONFORMANCE	SQL_OSCC_COMPLIANT
Level of SQL conformance	SQL_ODBC_SQL _CONFORMANCE	SQL_OSC_EXTENDED
Referential integrity syntax support	SQL_ODBC_SQL_OPT_IEF	Y

Table A–1: Information the Progress SQL-92 ODBC Driver Returns to SQLGetInfo *(19 of 27)*

Description	*fInfoType* Argument	Returns
ODBC version supported by driver manager	SQL_ODBC_VER	03.51.0000
Types of outer joins supported	SQL_OJ_CAPABILITIES	SQL_OJ_LEFT SQL_OJ_RIGHT SQL_OJ_NOT_ORDERED SQL_OJ_INNER SQL_OJ_ALL _COMPARISON_OPS
Whether columns in ORDER BY clause must also be in select list	SQL_ORDER_BY_COLUMNS _IN_SELECT	N
Support for outer joins	SQL_OUTER_JOINS	Y
Name for an owner	SQL_OWNER_TERM	owner
Statements in which owner may be used	SQL_OWNER_USAGE	SQL_OU_DML _STATEMENTS SQL_OU_PROCEDURE _INVOCATION SQL_OU_TABLE _DEFINITION SQL_OU_INDEX _DEFINITION SQL_OU_PRIVILEGE _DEFINITION
Characteristics of row counts available in a parameterized execution	SQL_PARAM_ARRAY_ROW _COUNTS	SQL_PARC_NO_BATCH
Characteristics of result sets available in a parameterized execution	SQL_PARAM_ARRAY_ SELECTS	SQL_PAS_NO_SELECT

Table A–1: Information the Progress SQL-92 ODBC Driver Returns to SQLGetInfo *(20 of 27)*

Description	*fInfoType* Argument	Returns
Supported operations in SQLSetPos	SQL_POS_OPERATIONS	SQL_POS_POSITION SQL_POS_REFRESH SQL_POS_UPDATE SQL_POS_DELETE SQL_POS_ADD
Supported positioned SQL statements	SQL_POSITIONED_ STATEMENTS	SQL_PS_POSITIONED _DELETE SQL_PS_POSITIONED _UPDATE SQL_PS_SELECT_FOR _UPDATE
Term for procedures	SQL_PROCEDURE_TERM	procedure
SQL procedures support	SQL_PROCEDURES	Y
Support for qualifiers	SQL_QUALIFIER_USAGE	0x00000000
Case sensitivity of quoted user-supplied names	SQL_QUOTED_IDENTIFIER _CASE	SQL_IC_MIXED
Separator character used between qualifier name and element	SQL_QUALIFIER_NAME _SEPARATOR	“ ”
Term used for a qualifier	SQL_QUALIFIER_TERM	“ ”
Position of the qualifier in a qualified table name	SQL_QUALIFIER_LOCATION	0 (does not support)

Table A–1: Information the Progress SQL-92 ODBC Driver Returns to SQLGetInfo *(21 of 27)*

Description	*fInfoType* Argument	Returns
Detect changes to any row in mixed-cursor operations	SQL_ROW_UPDATES	Y
Term for entity that has owner privileges on objects	SQL_SCHEMA_TERM	owner
Statements that support use of owner qualifiers	SQL_SCHEMA_USAGE	SQL_OU_DML_ STATEMENTS SQL_OU_PROCEDURE _INVOCATION SQL_OU_TABLE _DEFINITION SQL_OU_INDEX _DEFINITION SQL_OU_PRIVILEGE _DEFINITION
Options supported for scrollable cursors	SQL_SCROLL_OPTIONS	SQL_SO_FORWARD _ONLY SQL_SO_STATIC SQL_SO_KEYSET_DRIVEN
Support for scrollable cursors	SQL_SCROLL_ CONCURRENCY	SQL_SCCO_READ_ONLY SQL_SCCO_OPT_VALUES
Character to permit wildcard characters in search strings	SQL_SEARCH_PATTERN _ESCAPE	\ (backslash)
Name of the system where the Progress SQL-92 ODBC data source resides	SQL_SERVER_NAME	(string containing the name)
Special characters allowed in user-supplied names	SQL_SPECIAL_ CHARACTERS	"_", "%"

Table A–1: Information the Progress SQL-92 ODBC Driver Returns to SQLGetInfo *(22 of 27)*

Description	*fInfoType* Argument	Returns
Datetime scalar functions supported	SQL_SQL92_DATETIME _FUNCTIONS	0x00000000
Behavior of DELETE statement that refers to a foreign key	SQL_SQL92_FOREIGN_KEY _DELETE_RULE	0x00000000
Behavior of UPDATE statement that refers to a foreign key	SQL_SQL92_FOREIGN_KEY _UPDATE_RULE	0x00000000
GRANT statement clauses supported	SQL_SQL92_GRANT	SQL_SG_DELETE_TABLE SQL_SG_INSERT_TABLE SQL_SG_INSERT _COLUMN SQL_SG_REFERENCES _TABLE SQL_SG_REFERENCES _COLUMN SQL_SG_SELECT_TABLE SQL_SG_UPDATE _TABLE SQL_SG_UPDATE _COLUMN
Numeric scalar functions supported	SQL_SQL92_NUMERIC_ VALUE_FUNCTIONS	SQL_SNVF_CHAR _LENGTH SQL_SNVF_CHARACTER _LENGTH
Predicates supported	SQL_SQL92_PREDICATES	SP_EXISTS SQL_SP_ISNOTNULL SQL_SP_ISNULL SQL_SP_UNIQUE SQL_SP_LIKE SQL_SP_IN SQL_SP_BETWEEN
Relational join operators supported	SQL_SQL92_RELATIONAL _JOIN_OPERATORS	0x00000000

Table A–1: Information the Progress SQL-92 ODBC Driver Returns to SQLGetInfo *(23 of 27)*

Description	*fInfoType* Argument	Returns
REVOKE statement clauses supported	SQL_SQL92_REVOKE	SQL_SR_GRANT_ OPTION_FOR SQL_SR_CASCADE SQL_SR_DELETE_TABLE SQL_SR_INSERT_TABLE SQL_SR_INSERT _COLUMN SQL_SR_REFERENCES _TABLE SQL_SR_REFERENCES _COLUMN SQL_SR_SELECT_TABLE SQL_SR_UPDATE _TABLE SQL_SR_UPDATE _COLUMN
Row value constructor expressions supported	SQL_SQL92_ROW_VALUE _CONSTRUCTOR	0x00000000
String scalar functions supported	SQL_SQL92_STRING _FUNCTIONS	SQL_SSF_CONVERT SQL_SSF_LOWER SQL_SSF_UPPER SQL_SSF_SUBSTRING SQL_SSF_TRANSLATE SQL_SSF_TRIM _LEADING SQL_SSF_TRIM_TRAILING
Value expressions supported	SQL_SQL92_VALUE _EXPRESSIONS	SQL_SVE_COALESCE SQL_SVE_NULLIF
CLI standards to which the driver conforms	SQL_STANDARD_CLI _CONFORMANCE	SQL_SCC_XOPEN_CLI _VERSION1

Table A–1: Information the Progress SQL-92 ODBC Driver Returns to SQLGetInfo *(24 of 27)*

Description	*fInfoType* Argument	Returns
Supported attributes of a static cursor: subset 1	SQL_STATIC_CURSOR _ATTRIBUTES1	SQL_CA1_NEXT SQL_CA1_ABSOLUTE SQL_CA1_RELATIVE SQL_CA1_BOOKMARK SQL_CA1_LOCK_NO _CHANGE SQL_CA1_POS_POSITION SQL_CA1_POS_UPDATE SQL_CA1_POS_DELETE SQL_CA1_POS_REFRESH SQL_CA1_POSITIONED _UPDATE SQL_CA1_POSITIONED _DELETE SQL_CA1_SELECT_FOR _UPDATE SQL_CA1_BULK_ADD
Supported attributes of a static cursor: subset 2	SQL_STATIC_CURSOR _ATTRIBUTES2	SQL_CA2_READ_ONLY _CONCURRENCY SQL_CA2_OPT_VALUES _CONCURRENCY SQL_CA2_CRC_EXACT SQL_CA2_SIMULATE _TRY_UNIQUE
Support for detection of changes made to a static or key-set driven cursor through SQLSetPos	SQL_STATIC_SENSITIVITY	0x00000000

Table A–1: Information the Progress SQL-92 ODBC Driver Returns to SQLGetInfo *(25 of 27)*

Description	*fInfoType* Argument	Returns
String functions supported	SQL_STRING_FUNCTIONS	SQL_FN_STR_CONCAT SQL_FN_STR_INSERT SQL_FN_STR_LEFT SQL_FN_STR_LTRIM SQL_FN_STR_LENGTH SQL_FN_STR_LOCATE SQL_FN_STR_LCASE SQL_FN_STR_REPEAT SQL_FN_STR_REPLACE SQL_FN_STR_RIGHT SQL_FN_STR_RTRIM SQL_FN_STR _SUBSTRING SQL_FN_STR_UCASE SQL_FN_STR_ASCII SQL_FN_STR_CHAR SQL_FN_STR _DIFFERENCE SQL_FN_STR_LOCATE_2 SQL_FN_STR_SPACE SQL_FN_STR_CHAR _LENGTH SQL_FN_STR _CHARACTER_LENGTH
Predicates that support subqueries	SQL_SUBQUERIES	SQL_SQ_COMPARISON SQL_SQ_EXISTS SQL_SQ_IN SQL_SQ_QUANTIFIED
System functions supported	SQL_SYSTEM_FUNCTIONS	SQL_FN_SYS _USERNAME
Term for tables	SQL_TABLE_TERM	table
Timestamp intervals supported for TIMESTAMPADD function	SQL_TIMEDATE_ADD _INTERVALS	0x00000000

Table A–1: Information the Progress SQL-92 ODBC Driver Returns to SQLGetInfo *(26 of 27)*

Description	*fInfoType* Argument	Returns
Timestamp intervals supported for TIMESTAMPDIFF function	SQL_TIMEDATE_DIFF_INTERVALS	0x00000000
Date-time functions supported	SQL_TIMEDATE_FUNCTIONS	SQL_FN_TD_NOW SQL_FN_CURDATE SQL_FN_TD_DAYOFMONTH SQL_FN_TD_DAYOFWEEK SQL_FN_TD_DAYOFYEAR SQL_FN_TD_MONTH SQL_FN_TD_QUARTER SQL_FN_TD_WEEK SQL_FN_TD_YEAR SQL_FN_CURTIME SQL_FN_TD_HOUR SQL_FN_TD_MINUTE SQL_FN_TD_SECOND SQL_FN_TD_TIMESTAMP_ADD SQL_FN_TD_TIMESTAMPDIFF SQL_FN_TD_DAYNAME SQL_FN_TD_MONTHNAME
Support for DML, DDL within transactions	SQL_TXN_CAPABLE	SQL_TC_ALL
Options for setting transaction isolation levels	SQL_TXN_ISOLATION_OPTION	SQL_TXN_READ_UNCOMMITTED SQL_TXN_SERIALIZABLE SQL_TXN_READ_COMMITTED SQL_TXN_REPEATABLE_READ

Table A–1: Information the Progress SQL-92 ODBC Driver Returns to SQLGetInfo *(27 of 27)*

Description	*fInfoType* Argument	Returns
UNION support	SQL_UNION	SQL_U_UNION SQL_U_UNION_ALL
Name of user connected to the data source	SQL_USER_NAME	(string containing the name)
Year of publication of the X/Open specification with which the driver complies	SQL_XOPEN_CLI_YEAR	1995

B

Progress and ODBC Data Types

This appendix shows how the Progress data types are mapped to the standard ODBC data types.

Table B–1 shows how the Progress data types are mapped to the standard ODBC data types.

Table B–1: Progress and ODBC Data Types

Progress Data Type	ODBC Data Type
Binary	SQL_BINARY
Bit	SQL_BIT
Char	SQL_CHAR
Date	SQL_TYPE_DATE
Decimal	SQL_DECIMAL
Double Precision	SQL_DOUBLE
Float	SQL_FLOAT
Integer	SQL_INTEGER
Real	SQL_FLOAT
Smallint	SQL_SMALLINT
Time	SQL_TYPE_TIME
Timestamp	SQL_TYPE_TIMESTAMP
Tinyint	SQL_TINYINT
Varbinary	SQL_VARBINARY
Lvarbinary	SQL_LONGVARBINARY
Varchar	SQL_VARCHAR

Glossary

Add [an ODBC data source]

Make a data source available to ODBC through the Add operation of the ODBC Administrator. Adding a data source tells ODBC where a specific database resides and which ODBC driver to use to access it. Adding a data source also invokes a setup dialog box for the particular driver so you can provide other details the driver needs to connect to the database.

Client

In client/server systems, the part of the system that sends requests to servers and processes the results of those requests.

Data Source

See **ODBC Data Source.**

Delete [an ODBC data source]

Removes information about an ODBC data source through the Delete operation of the ODBC Administrator utility. Deleting a data source does not delete the database it corresponds to, but removes information about the database's location from the \%WINDIR%\ODBC.INI file.

Delimited Identifiers

Names in SQL statements that are enclosed in double quotation marks ("). Enclosing a name in double quotation marks preserves the case of the name and allows it to include reserved words and special characters. Subsequent references to a delimited identifier must also use enclosing double quotation marks.

ODBC Administrator

Microsoft-supplied utility to add and delete ODBC data sources and drivers. The installation procedure installs the Administrator if it is not already present on a system. To invoke it, click on the ODBC icon in the Control Panel applications of the Windows Program Manager.

ODBC Application

Any program that calls ODBC functions and uses them to issue SQL statements. Many vendors have added ODBC capabilities to their existing Windows-based tools.

ODBC Data Source

In ODBC terminology, a specific combination of a database system, the operating system it uses, and any network software required to access it. Before applications can access a database through ODBC, you use the ODBC Administrator to add a data source—register information about the database and an ODBC driver that can connect to it—for that database. More than one data source name can refer to the same database, and deleting a data source does not delete the associated database.

ODBC Driver

Software that processes ODBC function calls for a specific data source. The driver connects to the data source, translates the standard SQL statements into syntax the data source can process, and returns data to the application.

ODBC Driver Manager

A Microsoft-supplied program that routes calls from an application to the appropriate ODBC driver for a data source.

SQL Engine

The core internal component of Progress SQL-92. The SQL engine receives requests from the Progress SQL-92 ODBC driver, processes them, and returns results.

Index

A

Audience vii

B

Bold typeface
  as typographical convention viii

C

Connecting
  connection string attributes 2–10
  using a connection string 2–10
  using a logon dialog box 2–9

Connection String Attibutes 2–10

D

Data Source
  adding 2–2
  configuring 2–2
  definition 1–3
  modifying 2–7

Data Source Name 2–3

Data Types
  how progress maps B–2
  ODBC B–2
  standard B–2

DLC/lib 4–5

Driver
  ODBC 1–3

Driver Manager
  ODBC 1–3, 2–2

E

environment variables
  LD_LIBRARY_PATH 4–2
  LIBPATH 4–2
  PATH 4–2
  SHLIB_PATH 4–2

F

Failure
  test connect 2–8

Functions
  API 3–2
  date 3–9
  numeric 3–7
  ODBC API 3–1

scalar 3–1, 3–4
date 3–10
numeric 3–7
string 3–5
system 3–12
time 3–10
string 3–4
system 3–12
time 3–9

I

Italic typeface
as typographical convention viii

K

Keystrokes ix

L

LD_LIBRARY_PATH 4–2

Levels
isolation 2–11
lock 2–11
ODBC Conformance 2–11

libodbc.a 4–2, 4–4

libodbc.sl 4–2, 4–4

libodbc.so 4–2, 4–4

LIBPATH 4–2

Logon
using a connection string 2–10
using logon dialog box 2–9

M

Manual
organization of vii
syntax notation ix

Monospaced typeface
as typographical convention ix

O

ODBC
API functions 3–1
application 1–3
architecture 1–3
components 1–4
data source 1–3
driver 1–3
driver manager 1–3, 2–2
environment 1–4
interface 1–2
Open Database Connectivity 1–1
overview 1–2

ODBC Administrator 2–2, 2–7
Create New Data Source dialog box 2–3
Data Source dialog box 2–2, 2–7
Driver Setup dialog box 2–3, 2–7

ODBC Driver 4–2
libodbc.a 4–2, 4–4
libodbc.sl 4–2, 4–4
libodbc.so 4–2, 4–4
odbctrac.a 4–4
odbctrac.sl 4–4
odbctrac.so 4–4

ODBC Driver Directory 4–2

odbctrac.a 4–4

odbctrac.sl 4–4

odbctrac.so 4–4

P

Problems
  test connect failure 2–8

S

SHLIB_PATH 4–2

SQLGetInfo A–1
  information returned A–2 to A–28

Support for
  connections 2–11
  isolation levels 2–11
  lock levels 2–11
  statements 2–11

Syntax notation ix

T

Test Connect
  failure 2–8

Typographical conventions viii