

iAnywhere Solutions[™] ODBC Drivers

Part number: 30129-01-0900-01

Last modified: June 2003

Copyright © 1989-2003 Sybase, Inc. Portions copyright © 2001-2003 iAnywhere Solutions, Inc. All rights reserved.

No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Sybase, Inc.

Sybase, SYBASE (logo), AccelaTrade, ADA Workbench, Adaptable Windowing Environment, Adaptive Component Architecture, Adaptive Server, Adaptive Server Anywhere, Adaptive Server Enterprise, Adaptive Server Enterprise Monitor, Adaptive Server Enterprise Replication, Adaptive Server Everywhere, Adaptive Server IQ, Adaptive Warehouse, AnswerBase, Anywhere Studio, Application Manager, AppModeler, APT Workbench, APT-Build, APT-Edit, APT-Execute, APT-Library, APT-Translator, ASEP, AvantGo, AvantGo Application Alerts, AvantGo Mobile Delivery, AvantGo Mobile Document Viewer, AvantGo Mobile Inspection, AvantGo Mobile Marketing Channel, AvantGo Mobile Pharma, AvantGo Mobile Sales, AvantGo Pylon, AvantGo Pylon Application Server, AvantGo Pylon Conduit, AvantGo Pylon PIM Server, AvantGo Pylon Pro, Backup Server, BayCam, Bit-Wise, BizTracker, Certified PowerBuilder Developer, Certified SYBASE Professional, Certified SYBASE Professional (logo), ClearConnect, Client Services, Client-Library, CodeBank, Column Design, ComponentPack, Connection Manager, Convoy/DM, Copernicus, CSP, Data Pipeline, Data Workbench, DataArchitect, Database Analyzer, DataExpress, DataServer, DataWindow, DB-Library, dbQueue, Developers Workbench, Direct Connect Anywhere, DirectConnect, Distribution Director, Dynamic Mobility Model, Dynamo, e-ADK, E-Anywhere, e-Biz Integrator, E-Whatever, EC Gateway, ECMAP, ECRTP, eFulfillment Accelerator, Electronic Case Management, Embedded SQL, EMS, Enterprise Application Studio, Enterprise Client/Server, Enterprise Connect, Enterprise Data Studio, Enterprise Manager, Enterprise Portal (logo), Enterprise SQL Server Manager, Enterprise Work Architecture, Enterprise Work Designer, Enterprise Work Modeler, eProcurement Accelerator, eremote, Everything Works Better When Everything Works Together, EWA, Financial Fusion, Financial Fusion (and design), Financial Fusion Server, Formula One, Fusion Powered e-Finance, Fusion Powered Financial Destinations, Fusion Powered STP, Gateway Manager, GeoPoint, GlobalFIX, iAnywhere, iAnywhere Solutions, ImpactNow, Industry Warehouse Studio, InfoMaker, Information Anywhere, Information Everywhere, InformationConnect, InstaHelp, InternetBuilder, iremote, iScript, Jaguar CTS, jConnect for JDBC, KnowledgeBase, Logical Memory Manager, M-Business Channel, M-Business Network, M-Business Server, Mail Anywhere Studio, MainframeConnect, Maintenance Express, Manage Anywhere Studio, MAP, MDI Access Server, MDI Database Gateway, media.splash, Message Anywhere Server, MetaWorks, MethodSet, ML Ouery, MobiCATS, My AvantGo, My AvantGo Media Channel, My AvantGo Mobile Marketing, MySupport, Net-Gateway, Net-Library, New Era of Networks, Next Generation Learning, Next Generation Learning Studio, O DEVICE, OASiS, OASiS (logo), ObjectConnect, ObjectCycle, OmniConnect, OmniSQL Access Module, OmniSQL Toolkit, Open Biz, Open Business Interchange, Open Client, Open Client/Server, Open Client/Server Interfaces, Open ClientConnect, Open Gateway, Open Server, Open ServerConnect, Open Solutions, Optima++, Partnerships that Work, PB-Gen, PC APT Execute, PC DB-Net, PC Net Library, PhysicalArchitect, Pocket PowerBuilder, PocketBuilder, Power Through Knowledge, Power++, power.stop, PowerAMC, PowerBuilder, PowerBuilder Foundation Class Library, PowerDesigner, PowerDimensions, PowerDynamo, Powering the New Economy, PowerS, PowerScript, PowerSite, PowerSocket, Powersoft, Powersoft Portfolio, Powersoft Professional, PowerStage, PowerStudio, PowerTips, PowerWare Desktop, PowerWare Enterprise, ProcessAnalyst, QAnywhere, Rapport, Relational Beans, RepConnector, Replication Agent, Replication Driver, Replication Server, Replication Server Manager, Replication Toolkit, Report Workbench, Report-Execute, Resource Manager, RW-DisplayLib, RW-Library, S.W.I.F.T. Message Format Libraries, SAFE, SAFE/PRO, SDF, Secure SQL Server, Secure SQL Toolset, Security Guardian, SKILS, smart.partners, smart.parts, smart.script, SQL Advantage, SQL Anywhere, SQL Anywhere Studio, SQL Code Checker, SQL Debug, SQL Edit, SQL Edit/TPU, SQL Everywhere, SQL Modeler, SQL Remote, SQL Server, SQL Server Manager, SQL Server SNMP SubAgent, SQL Server/CFT, SQL Server/DBM, SQL SMART, SQL Station, SQL Toolset, SQLJ, Stage III Engineering, Startup.Com, STEP, SupportNow, Sybase Central, Sybase Client/Server Interfaces, Sybase Development Framework, Sybase Financial Server, Sybase Gateways, Sybase Learning Connection, Sybase MPP, Sybase SQL Desktop, Sybase SQL Lifecycle, Sybase SQL Workgroup, Sybase Synergy Program, Sybase User Workbench, Sybase Virtual Server Architecture, SybaseWare, Syber Financial, SyberAssist, SybMD, SyBooks, System 10, System 11, System XI (logo), System Tools, Tabular Data Stream, The Enterprise Client/Server Company, The Extensible Software Platform, The Future Is Wide Open, The Learning Connection, The Model For Client/Server Solutions, The Online Information Center, The Power of One, TradeForce, Transact-SQL, Translation Toolkit, Turning Imagination Into Reality, UltraLite, UltraLite.NET, UNIBOM, Unilib, Uninull, Unisep, Unistring, URK Runtime Kit for UniCode, Versacore, Viewer, VisualWriter, VQL, Warehouse Control Center, Warehouse Studio, Warehouse WORKS, WarehouseArchitect, Watcom, Watcom SQL, Watcom SQL Server, Web Deployment Kit, Web.PB, Web.SQL, WebSights, WebViewer, WorkGroup SQL Server, XA-Library, XA-Server, and XP Server are trademarks of Sybase, Inc. or its subsidiaries.

All other trademarks are property of their respective owners.

Contents

About	t This Manual	v
	SQL Anywhere Studio documentation	vi
	Finding out more and providing feedback	ix
1	Introduction to ODBC Drivers	1
	About iAnywhere Solutions ODBC drivers	2
	Quick start for Windows	3
	Quick start for UNIX	4
	Key driver settings for MobiLink	9
	Error messages	12
2	iAnywhere Solutions ODBC Driver for Sybase Adaptive Server Enter-	
	prise	15
		16
	Configuring data sources	17
	Connecting to a data source using a logon dialog box	19
	Connecting to a data source using a connection string	20
		29
	Supported features	30
3	iAnywhere Solutions ODBC Driver for Oracle Wire Protocol	31
	Driver requirements	32
	Configuring data sources	33
	Connecting to a Data Source Using a Logon Dialog Box	36
	Connecting to a data source using a connection string	37
	Data types	42
	Unicode support	43
	Default Unicode Manning	44
	Stored procedure results	15
	Supported features	46
4	iAnywhere Solutions ODBC driver for DB2	47
•	Introduction	48
	Configuring data sources	10
	Connecting to a data source using a logon dialog box	-72 -50
	Connecting to a data source using a rogon dialog box	51
		59
	Data types	50
		39

Α	ODBC Functions	61 62
В	Values for AppCodePage Connection String Attribute Table of AppCodePage values	69 70
	Index	75

About This Manual

Subject	This manual describes how to use the iAnywhere Solutions ODBC drivers for Oracle, DB2, and Sybase Adaptive Server Enterprise. These ODBC drivers are included with SQL Anywhere Studio for use with MobiLink synchronization and Adaptive Server Anywhere remote data access.
	The Adaptive Server Anywhere ODBC driver is described in the main Adaptive Server Anywhere documentation, and is not described in this book.
Audience	This manual is for MobiLink synchronization users working with Oracle, DB2, or Sybase Adaptive Server Enterprise consolidated databases.

SQL Anywhere Studio documentation

This book is part of the SQL Anywhere documentation set. This section describes the books in the documentation set and how you can use them.

The SQL Anywhere Studio documentation

The SQL Anywhere Studio documentation is available in a variety of forms: in an online form that combines all books in one large help file; as separate PDF files for each book; and as printed books that you can purchase. The documentation consists of the following books:

- Introducing SQL Anywhere Studio This book provides an overview of the SQL Anywhere Studio database management and synchronization technologies. It includes tutorials to introduce you to each of the pieces that make up SQL Anywhere Studio.
- What's New in SQL Anywhere Studio This book is for users of previous versions of the software. It lists new features in this and previous releases of the product and describes upgrade procedures.
- ◆ Adaptive Server Anywhere Getting Started This book is for people new to relational databases or new to Adaptive Server Anywhere. It provides a quick start to using the Adaptive Server Anywhere database-management system and introductory material on designing, building, and working with databases.
- ♦ Adaptive Server Anywhere Database Administration Guide This book covers material related to running, managing, and configuring databases and database servers.
- ◆ Adaptive Server Anywhere SQL User's Guide This book describes how to design and create databases; how to import, export, and modify data; how to retrieve data; and how to build stored procedures and triggers.
- Adaptive Server Anywhere SQL Reference Manual This book provides a complete reference for the SQL language used by Adaptive Server Anywhere. It also describes the Adaptive Server Anywhere system tables and procedures.
- ◆ Adaptive Server Anywhere Programming Guide This book describes how to build and deploy database applications using the C, C++, and Java programming languages. Users of tools such as Visual Basic and PowerBuilder can use the programming interfaces provided by those tools. It also describes the Adaptive Server Anywhere ADO.NET data provider.

- ♦ Adaptive Server Anywhere Error Messages This book provides a complete listing of Adaptive Server Anywhere error messages together with diagnostic information.
- ◆ SQL Anywhere Studio Security Guide This book provides information about security features in Adaptive Server Anywhere databases. Adaptive Server Anywhere 7.0 was awarded a TCSEC (Trusted Computer System Evaluation Criteria) C2 security rating from the U.S. Government. This book may be of interest to those who wish to run the current version of Adaptive Server Anywhere in a manner equivalent to the C2-certified environment.
- MobiLink Synchronization User's Guide This book describes how to use the MobiLink data synchronization system for mobile computing, which enables sharing of data between a single Oracle, Sybase, Microsoft or IBM database and many Adaptive Server Anywhere or UltraLite databases.
- ♦ MobiLink Synchronization Reference This book is a reference guide to MobiLink command line options, synchronization scripts, SQL statements, stored procedures, utilities, system tables, and error messages.
- ◆ iAnywhere Solutions ODBC Drivers This book describes how to set up ODBC drivers to access consolidated databases other than Adaptive Server Anywhere from the MobiLink synchronization server and from Adaptive Server Anywhere remote data access.
- ◆ SQL Remote User's Guide This book describes all aspects of the SQL Remote data replication system for mobile computing, which enables sharing of data between a single Adaptive Server Anywhere or Adaptive Server Enterprise database and many Adaptive Server Anywhere databases using an indirect link such as e-mail or file transfer.
- SQL Anywhere Studio Help This book includes the context-sensitive help for Sybase Central, Interactive SQL, and other graphical tools. It is not included in the printed documentation set.
- ♦ UltraLite Database User's Guide This book is intended for all UltraLite developers. It introduces the UltraLite database system and provides information common to all UltraLite programming interfaces.
- ◆ UltraLite Interface Guides A separate book is provided for each UltraLite programming interface. Some of these interfaces are provided as UltraLite components for rapid application development, and others are provided as static interfaces for C, C++, and Java development.

In addition to this documentation set, PowerDesigner and InfoMaker include their own online documentation.

Documentation formats SQL Anywhere Studio provides documentation in the following formats:

◆ Online documentation The online documentation contains the complete SQL Anywhere Studio documentation, including both the books and the context-sensitive help for SQL Anywhere tools. The online documentation is updated with each maintenance release of the product, and is the most complete and up-to-date source of documentation.

To access the online documentation on Windows operating systems, choose Start > Programs > SQL Anywhere 9 > Online Books. You can navigate the online documentation using the HTML Help table of contents, index, and search facility in the left pane, as well as using the links and menus in the right pane.

To access the online documentation on UNIX operating systems, see the HTML documentation under your SQL Anywhere installation.

• **Printable books** The SQL Anywhere books are provided as a set of PDF files, viewable with Adobe Acrobat Reader.

The PDF files are available on the CD ROM in the *pdf_docs* directory. You can choose to install them when running the setup program.

◆ Printed books The complete set of books is available from Sybase sales or from eShop, the Sybase online store. You can access eShop by clicking How to Buy ➤ eShop at http://www.ianywhere.com.

Finding out more and providing feedback

We would like to receive your opinions, suggestions, and feedback on this documentation.

You can provide feedback on this documentation and on the software through newsgroups set up to discuss SQL Anywhere technologies. These newsgroups can be found on the *forums.sybase.com* news server.

The newsgroups include the following:

- sybase.public.sqlanywhere.general.
- sybase.public.sqlanywhere.linux.
- sybase.public.sqlanywhere.mobilink.
- sybase.public.sqlanywhere.product_futures_discussion.
- sybase.public.sqlanywhere.replication.
- sybase.public.sqlanywhere.ultralite.

Newsgroup disclaimer

iAnywhere Solutions has no obligation to provide solutions, information or ideas on its newsgroups, nor is iAnywhere Solutions obliged to provide anything other than a systems operator to monitor the service and insure its operation and availability.

iAnywhere Solutions Technical Advisors as well as other staff assist on the newsgroup service when they have time available. They offer their help on a volunteer basis and may not be available on a regular basis to provide solutions and information. Their ability to help is based on their workload.

CHAPTER 1

Introduction to ODBC Drivers

About this chapter	This chapter introduces the iAnywhere Solutions ODBC drivers that are included in SQL Anywhere Studio for use with MobiLink synchronization or Adaptive Server Anywhere remote data access.		
	Note The ODBC driver for Adaptive Server Anywhere is described in the main Adaptive Server Anywhere documentation, and not in this book. For more information, see "Working with ODBC data sources" [<i>ASA Database</i> <i>Administration Guide</i> , page 53].		
Contents	Торіс:	page	
	About iAnywhere Solutions ODBC drivers	2	
	Quick start for Windows	3	
	Quick start for UNIX	4	
	Key driver settings for MobiLink	9	
	Error messages	12	

About iAnywhere Solutions ODBC drivers

iAnywhere Solutions ODBC drivers are compliant with the Open Database Connectivity (ODBC) specification. ODBC is a specification for an application program interface (API) that enables applications to access multiple database management systems using SQL.

The iAnywhere Solutions ODBC drivers are for use only with MobiLink synchronization server and administration tools, and with Adaptive Server Anywhere remote data access. Other ODBC client applications cannot use these drivers.

iAnywhere Solutions ODBC drivers are provided for the following operating systems:

- Windows operating systems, except Windows CE
- UNIX operating systems: Solaris, AIX, and Linux

The iAnywhere Solutions ODBC drivers are installed by the SQL Anywhere Studio Setup program.

Note

The ODBC driver for Adaptive Server Anywhere is described in the main Adaptive Server Anywhere documentation, and not in this book. For more information, see "Working with ODBC data sources" [ASA Database Administration Guide, page 53].

Quick start for Windows

The iAnywhere Solutions ODBC drivers support the following Windows operating systems:

- Windows 95, Windows 98, Windows Me, Windows XP
- Windows NT with Service Pack 5 and higher
- Windows 2000 with Service Pack 1 and higher

On Windows, the ODBC drivers are 32-bit drivers. All required network software supplied by your database system vendors must be 32-bit compliant.

Starting the ODBC Administrator

The ODBC Administrator is used to manage data sources created for each driver.

To start the ODBC Administrator (Windows)

1. Double-click the ODBC Data Sources icon in the Control Panel.

To start the ODBC Administrator (command line)

1. At a command prompt, enter the command odbcad32.

Once in the ODBC Administrator, you can create a new data source by clicking Add and choosing one of the iAnywhere Solutions ODBC drivers.

For a list of key driver settings when using MobiLink, see "Key driver settings for MobiLink" on page 9.

Driver file names

The prefix for all iAnywhere Solutions ODBC driver file names is *wq*. The file extension is *.dll*. This indicates that they are dynamic link libraries.

Quick start for UNIX

The following UNIX operating systems are supported:

- Solaris
- ♦ Linux (Red Hat, Caldera, and SuSE)
- ♦ AIX

For information about the versions that are supported, see the UNIX Readme First, which is installed in your SQL Anywhere installation directory in two formats: readme.txt and readme.pdf.

The system information file (.odbc.ini)

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 your *\$HOME* directory and is usually called *.odbc.ini*. This file is maintained using any text editor, and defines data source entries as described in the following sections:

- Sybase Adaptive Server Enterprise "Connecting to a data source using a connection string" on page 20.
- Oracle "Connecting to a data source using a connection string" on page 37.
- IBM DB2 "Connecting to a data source using a connection string" on page 51.

You must use the long name of connection string attributes when defining data source entries. A sample file (*odbc.ini.sample*) is located in the driver installation directory.

For a list of key driver settings when using MobiLink, see "Key driver settings for MobiLink" on page 9.

System information fileThere 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=/opt/sybase/SYBSsa9/drivers

Sample system The following is a sample system information file for Solaris: information file

```
; This is only a sample .odbc.ini file
; IT WILL NOT WORK AS IS.
; You need to make a copy of this file
; into the user's home directory and modify
; the DSNs (Data Source Names) to your needs:
; cp odbc.ini.sample ~/.odbc.ini
; vi ~/.odbc.ini
; You also need to modify the default
; value of your ODBCINI environment variable.
; It should be set to point to your local
; copy of the .odbc.ini file:
; export ODBCINI=~/.odbc.ini
[ODBC Data Sources]
SybaseASA=Adaptive Server Anywhere 9 Driver
Oracle Wire Protocol=iAnywhere Solutions 9 - Oracle Wire
        Protocol Driver
Sybase ASE Wire Protocol=iAnywhere Solutions 9 - Sybase ASE
        Driver
DB2 Wire Protocol=iAnywhere Solutions 9 - DB2 Wire Protocol
        Driver
[SybaseASA]
Driver=/opt/sybase/SYBSsa9/lib/dbodbc9_r.so
UID=dba
PWD=sql
ServerName=asademo
CommLinks=tcpip
[Oracle Wire Protocol]
Driver=/opt/sybase/SYBSsa9/drivers/lib/wgora18.so
Description=iAnywhere Solutions 9 - Oracle Wire Protocol
LogonID=uid
Password=pwd
HostName=oracleserver
PortNumber=1521
SID=oraclesid
CatalogOptions=0
ProcedureRetResults=0
EnableDescribeParam=0
EnableStaticCursorsForLongData=0
ApplicationUsingThreads=1
```

```
[Sybase ASE Wire Protocol]
Driver=/opt/sybase/SYBSsa9/drivers/lib/wqase18.so
Description=iAnywhere Solutions 9 - Sybase ASE Driver
Database=db
LogonID=uid
Password=pwd
NetworkAddress=serverhost,4100
EnableDescribeParam=1
EnableQuotedIdentifiers=0
OptimizePrepare=2
RaiseErrorPositionBehavior=0
SelectMethod=0
ApplicationUsingThreads=1
DefaultLongDataBuffLen=1024
[DB2 Wire Protocol]
Driver=/opt/sybase/SYBSsa9/drivers/lib/wqdb218.so
Description=iAnywhere Solutions 9 - DB2 Wire Protocol Driver
LogonID=uid
Password=pwd
DB2AppCodePage=1252
ServerCharSet=1252
IpAddress=db2host
Database=db
TcpPort=50000
Package=db2package
Action=REPLACE
QueryBlockSize=8
CharSubTypeType=SYSTEM_DEFAULT
ConversationType=SINGLE_BYTE
CloseConversation=DEALLOC
UserBufferSize=32
MaximumClients=35
GrantExecute=1
GrantAuthid=PUBLIC
OEMANSI=1
DecimalDelimiter=PERIOD
DecimalPrecision=15
StringDelimiter=SINGLE_QUOTE
IsolationLevel=CURSOR_STABILITY
ResourceRelease=DEALLOCATION
DynamicSections=32
Trace=0
WithHold=1
[ODBC]
Trace=0
TraceFile=odbctrace.out
TraceDll=/opt/sybase/SYBSsa9/drivers/lib/odbctrac.so
InstallDir=/opt/sybase/SYBSsa9/drivers
ConversionTableLocation=/opt/sybase/SYBSsa9/drivers/tables
UseCursorLib=0
```

Using a centralized file UNIX support of the database drivers also 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 path name of the centralized file. For example, in the C shell you could set this variable as follows:

setenv ODBCINI /opt/odbc/system_odbc.ini

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

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

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

1. Check ODBCINI

2. Check \$HOME for .odbc.ini

Driver file names

The iAnywhere Solutions ODBC drivers are ODBC API-compliant dynamic link libraries, referred to in UNIX as **shared objects**. The prefix for all ODBC driver file names on UNIX is *wq*. On UNIX, the driver file names are lowercase and the extension is *.so*.

Setting the Library Path environment variable

	You must include the full path to the dynamic link libraries in the environment variable LD_LIBRARY_PATH. For example, if you install the ODBC drivers in the system directory /opt/sybase/SYBSsa9/drivers, then the fully qualified path for the ODBC pack is /opt/sybase/SYBSsa9/drivers/lib.
	If you do not include the path /opt/sybase/SYBSsa9/drivers/lib in the environment variable LD_LIBRARY_PATH, then your applications are unable to load the ODBC drivers dynamically at runtime or to display error message text.
	iAnywhere Solutions ODBC drivers require several environment variables to be set.
Additional environment variables	Some iAnywhere Solutions ODBC drivers must have environment variables set as required by the database client components used by the drivers. Consult the driver requirements in each of the individual driver sections for additional information pertaining to individual driver requirements.
	ODBCINI is an optional environment variable that all iAnywhere Solutions ODBC drivers recognize. ODBCINI is used to locate an ODBC information file other than the default file.

Setting the database client environment

In addition to setting the environment variables required by your particular database client, you must also add the client's library directory to your shared library path. For example, to add the Oracle lib directory /db/oracle/lib to the shared library path, C shell users would enter:

setenv LD_LIBRARY_PATH /db/oracle/lib:\${LD_LIBRARY_PATH}

Bourne or Korn shell users would use:

LD_LIBRARY_PATH=/db/oracle/lib:\$LD_LIBRARY_PATH export LD_LIBRARY_PATH

Key driver settings for MobiLink

This section lists key driver settings for using the ODBC drivers for MobiLink. Other settings can be left at their default values.

For a complete listing of all settings, see "iAnywhere Solutions ODBC Driver for Sybase Adaptive Server Enterprise" on page 15, "iAnywhere Solutions ODBC Driver for Oracle Wire Protocol" on page 31, and "iAnywhere Solutions ODBC driver for DB2" on page 47.

For more information about using ODBC drivers with MobiLink, see *http://www.ianywhere.com/developer/technotes/odbc_mobilink.html*.

Key settings for MobiLink and Adaptive Server Enterprise

The ODBC driver for Adaptive Server Enterprise does not require any Adaptive Server Enterprise client software. The following are key settings for Adaptive Server Enterprise. Other parameters are available and described elsewhere, but are less important.

- **Basic connection settings** Set the following basic connection settings:
 - **Data source name** This value is used to identify the data source. On Windows, it is located on the General tab. On UNIX, this attribute is called DataSourceName (DSN).
 - Network library name For TCP/IP networks, set this to Winsock. On Windows, it is located on the General tab. On UNIX, this attribute is called Network Library Name (NLM).
 - Network address and port of the server (General tab) These should be separated by commas; for example, server-machine,1500. On Windows, it is located on the General tab. On UNIX, this attribute is called NetworkAddress (NA).
 - Default Logon ID used for the connection This is case sensitive and must correspond to a valid logon ID on the server. On Windows, it is located on the Connection tab. On UNIX, this attribute is called LogonID (UID).
- DefaultLongDataBuffLen This parameter specifies (in Kb) the size of the largest LONG BINARY or LONG VARCHAR column value being synchronized. The default setting is 1024, which is equivalent to 1 Mb. If you are synchronizing long columns with entries larger than this, set the value higher.

This parameter appears on the Advanced tab of the Windows driver ODBC setup dialog. On UNIX, this attribute is called DefaultLongDataBuffLen (DLDBL). ◆ SelectMethod This parameter can be 0 or 1. When you use 0 (the default), database cursors are used. When set to 1, SELECT statements are run directly, without cursors.

This parameter appears on the Performance tab of the Windows driver ODBC setup dialog. On UNIX, this attribute is called SelectMethod (SM).

• **OptimizePrepare** This parameter can be 0, 1, 2, or 3. For use with MobiLink, set it to 2.

This parameter appears on the Performance tab of the Windows driver ODBC setup dialog. On UNIX, this attribute is called OptimizePrepare (OP).

For complete information about settings for the Adaptive Server Enterprise driver, see "Connecting to a data source using a connection string" on page 20.

Key settings for MobiLink and Oracle

There is one Oracle driver available: the Oracle wire protocol driver.

Oracle wire protocol The iAnywhere Solutions Oracle Wire Protocol ODBC driver does not require Oracle client software.

The following are key settings for the Oracle wire protocol driver:

- Basic connection settings Set the following basic connection settings:
 - Data source name This value is used to identify the data source. On Windows, it is located on the General tab. On UNIX, this attribute is called DataSourceName (DSN).
 - Host This value provides the location of your Oracle server. On Windows, it is located on the General tab. On UNIX, this attribute is called HostName (HOST).
 - Port Identifies the port number of your Oracle listener. On Windows, it is located on the General tab. On UNIX, this attribute is called PortNumber (PORT).
 - SID Identifies the Oracle database system ID. On Windows, it is located on the General tab. On UNIX, this attribute is called Sid (SID).
 - Default user name This must correspond to a valid user name on the server. On Windows, it is located on the Advanced tab. On UNIX, this attribute is called LoginID (UID).

For complete information about settings for the Oracle wire protocol driver, see "Connecting to a data source using a connection string" on page 37.

Key settings for MobiLink and DB2

The iAnywhere Solutions ODBC driver for IBM DB2, which is available only on AIX, does not require any DB2 client software. The following are key settings for DB2. Other parameters are available and described elsewhere, but are of less importance.

- **Basic connection settings** Set the following basic connection settings:
 - **DataSourceName (DSN)** This value is used to identify the data source.
 - **IpAddress (IP)** The IP address of the machine where the catalog tables are stored.
 - LogonID (UID) This default user ID must correspond to a valid user name on the server.
- WithHold (WH) For use with MobiLink, this attribute must be set to 1 in the ODBC configuration file.
- DecimalPrecision (DP) For use with MobiLink, this attribute must be set to 31.
- Action The first time you bind a package for a data source, set this to Create to create a package. Later, set it to Replace.

For complete information about settings for DB2, see "Connecting to a data source using a connection string" on page 51.

Bind packages Users must create the DB2 bind packages on every server to which they intend to connect with the driver. The driver will not work properly with any server that does not have the packages created. The UNIX version of the driver is provided with a program that creates the bind package.

To bind a package

1. At a command prompt, enter the following command:

bind18 dsn

where *dsn* is the ODBC data source name. You are prompted for a user ID and password if they are not stored in the system information file.

The *bind18* program is shipped with MobiLink, and can be found in the *drivers/lib* directory under your SQL Anywhere directory. You must export LD_LIBRARY_PATH or LIBPATH before running bind18.

Error messages

Error messages can come from the following sources:

- ♦ An ODBC driver
- The database system
- The ODBC driver manager

ODBC driver errors An error reported on an ODBC driver has the following format:

[vendor] [ODBC_component] message

where *ODBC_component* is the component in which the error occurred. If you receive this type of error, check the last ODBC call made by your application for possible problems or contact your ODBC application vendor.

An error that occurs in the data source includes the data store name, in the following format:

[vendor] [ODBC_component] [data_store] message

With this type of message, *ODBC_component* is the component that received the error from the data store indicated. For example, you may receive the following message from an Oracle data store:

[MERANT] [ODBC Oracle driver] [Oracle] ORA-0919: specified length too long for CHAR column

If you receive this type of error, something is wrong with the database system. Check your database system documentation for more information or consult your database administrator. In this example, you would check your Oracle documentation.

Driver Manager errors The driver manager is a DLL or shared object that establishes connections with drivers, submits requests to drivers, and returns results to applications. An error that occurs in the driver manager has the following format:

[vendor] [ODBC XXX] message

For example, an error from the Microsoft driver manager might look like this:

[Microsoft] [ODBC Driver Manager] Driver does not support this function

If you receive this type of error, consult the Programmer's Reference for the Microsoft ODBC Software Development Kit that is available from Microsoft.

UNIX error handling

UNIX error handling follows the X/Open XPG3 messaging catalog system. Localized error messages are stored in the subdirectory *locale/localized_territory_directory/LC_MESSAGES*, where *localized_territory_directory* depends on your language.

For instance, German localization files are stored in *locale/de/LC_MESSAGES*, where *de* is the locale for German.

If localized error messages are not available for your locale, then they will contain message numbers instead of text. For example:

[MERANT] [ODBC 20101 driver] 30040

CHAPTER 2

iAnywhere Solutions ODBC Driver for Sybase Adaptive Server Enterprise

About this chapter	This chapter describes how to configure and use the iAny ODBC driver for Sybase Adaptive Server Enterprise data	This chapter describes how to configure and use the iAnywhere Solutions DDBC driver for Sybase Adaptive Server Enterprise databases.	
Contents	Торіс:	page	
	Introduction	16	
	Configuring data sources	17	
	Connecting to a data source using a logon dialog box	19	
	Connecting to a data source using a connection string	20	
	Data types	29	
	Supported features	30	

Introduction

iAnywhere Solutions ODBC driver for Sybase Adaptive Server Enterprise supports Adaptive Server Enterprise 11.0 and higher database systems (including Adaptive Server 12.5) in the Windows and UNIX environments.

Driver requirements

The Adaptive Server Enterprise driver is a wire protocol driver, meaning that there is no additional client software required for the Adaptive Server Enterprise driver.

Configuring data sources

On Windows, data sources are configured and modified through the ODBC Administrator.

In the UNIX environment, there is no ODBC Administrator. To configure a data source in the UNIX environment, you must edit the system information file.

For information about editing this file, see "Quick start for UNIX" on page 4.

* To configure a Sybase Adaptive Server Enterprise data source

- 1. Start the ODBC Administrator to display a list of data sources.
- 2. If you are configuring an existing data source, select the data source name and click **Configure** to display the Adaptive Server Enterprise Driver Setup dialog box.
- If you are configuring a new data source, click Add to display a list of installed drivers. Select the iAnywhere Solutions 9 - Sybase ASE driver and click Finish to display the Adaptive Server Enterprise Driver Setup dialog box.
- 4. On the General tab, provide the following connection information. Click Apply.
 - ◆ Data Source Name Type a string that identifies this Adaptive Server Enterprise data source configuration in the system information. Examples include Accounting or Sys11-Serv1.
 - Description Type an optional long description of a data source name. For example, My Accounting Database or System 11 on Server number 1.
 - Network Library Name Select the name of the network library. This specifies which network protocol to use. The values are Winsock and NamedPipes.
 - **Network Address** Type the network address. The value you specify depends on which network protocol is chosen under Network Library Name and on the Adaptive Server Enterprise server.

If you choose Winsock for the Network Library Name, specify an IP address as follows: *servername-or-IP-address*, *port-number*. For example, if your network supports named servers, you may specify an address such as **Sybaseserver**, **5000**. You may also specify the IP address directly such as **199.226.224.34**, **5000**.

If you choose NamedPipes as the network protocol, you must specify the pipe address of the server. For example, \\machine1\sybase\pipe\query.

- ◆ Database Name Type the name of the database to which you want to connect by default. If you do not specify a value, the default is the database defined by the system administrator for each user.
- 5. Configure other settings in a manner appropriate for your application.

Most settings can be left at their default values. For a list of key driver settings when using MobiLink, see "Key driver settings for MobiLink" on page 9.

- 6. At any point during the configuration process, you can click **Test Connect** to attempt to connect to the data source using the connection properties specified in the Driver Setup dialog box. A logon dialog box is displayed. For more information, see "Connecting to a data source using a logon dialog box" on page 19. Note that the information you enter in the logon dialog box during a test connect is not saved.
 - If the driver can connect, it releases the connection and displays a "connection established" message. Click **OK**.
 - If the driver cannot connect because of an improper environment or incorrect connection value, it will display an appropriate error message. Click **OK**.
- 7. Click **OK** or **Cancel**. 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 alternate values.

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.

In this dialog box, perform the following steps:

- 1. In the Network Library field, select the name of the network library. This specifies which network protocol to use. Valid values are Winsock and NamedPipes.
- 2. In the Network Address field, type the network address, which depends on the network protocol chosen under Network Library and on the Adaptive Server Enterprise server. If you choose Winsock, specify an IP address as follows: *servername-or-IP-address*, *port-number*. For example, if your network supports named servers, you may specify an address such as **Sybaseserver**, **5000**. You may also specify the IP address directly such as **199.226.224.34**, **5000**.

If you have chosen NamedPipes as the network protocol, you must specify the pipe address of the server. For example, \\machine1\sybase\pipe\query.

- 3. If required, type your case-sensitive login ID.
- 4. If required, type your case-sensitive password for the system.
- 5. In the Database field, type the name of the database you want to access (case-sensitive) or select the name from the Database drop-down list, which displays the names you specified in the Adaptive Server Enterprise Driver Setup dialog box.
- 6. Click **OK** to complete the logon and to update the values in the system information.

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 for Adaptive Server Enterprise is:

DSN=SYS11 TABLES;SRVR=QESRVR;DB=PAYROLL;UID=JOHN;PWD=XYZZY

The following table gives the long and short names for each attribute, as well as a description. To configure a data source in the UNIX environment, you must edit the system information file. This file accepts only long names for attributes.

For information about editing this file, see "Quick start for UNIX" on page 4.

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.

Most settings can be left at their default values. For a list of key driver settings when using MobiLink, see "Key driver settings for MobiLink" on page 9.

Attribute	Description
AppCodePage (ACP)	Valid values for this attribute are listed in "Values for AppCodePage Connection String Attribute" on page 69. The code page that you specify must be the same as the code page used by your application. The driver on UNIX determines the value of the application's code page by checking for an
	AppCodePage value in the following order:
	 In the connection string In the DataSource section of the system file
	 In the ODBC section of the system file (odbc.ini)
	If no AppCodePage value is found, the driver uses the default value of 1 (ISO 8859-1 Latin- 1).
ApplicationName (APP)	The name used by Sybase to identify your application.
ApplicationUsingThreads (AUT)	ApplicationUsingThreads={0 1}. Ensures that the driver works with multi-threaded applications.
	When set to 1 (the initial default), the driver is thread-safe.
	When using the driver with single-threaded applications, you can set this option to 0 to avoid additional processing required for ODBC thread-safety standards.
ArraySize (AS)	The number of rows the driver retrieves from the server for a fetch. This is not the number of rows given to the user. This increases performance by reducing network traffic.
	The initial default is 50 rows.

Attribute	Description
Charset (CS)	The name of a character set. This character set must be installed on the Sybase server. The default is the setting on the Sybase server. For this driver to support Unicode, this attribute must be set to UTF-8. Refer to the Sybase server documentation for a list of valid character set names.
CursorCacheSize (CCS)	The number of connections that the con- nection cache can hold. To set the connection cache, you must set the SelectMethod attribute to 1. Increasing the connection cache may in- crease performance of some applications but requires additional database resources.
	The initial default is 1 (one cursor).
Database (DB)	The name of the database to which you want to connect.
DataSourceName (DSN)	A string that identifies a single connection to a Sybase database. Examples include "Accounting" or "Sys10-Serv1."
DefaultLongDataBuffLen (DLDBL)	An integer value that specifies, in 1024- byte multiples, the maximum length of data fetched from a TEXT or IMAGE column. You will need to increase this value if the total size of any long data exceeds 1 MB. The default is 1024.
DistributedTransaction- Model (DTM)	DistributedTransactionModel={XA Proto- col0 21 Native OLE}. Determines which model is used for distributed transaction sup- port. The initial default is XA Protocol. Leave this option at the default setting.

Attribute	Description
EnableDescribeParam (EDP)	EnableDescribeParam={0 1}. Determines whether the ODBC API function SQLDe- scribeParam is enabled.
	When set to 0 (the initial default), SQLDe- scribeParam is disabled.
	When set to 1, SQLDescribeParam is en- abled, which allows an application to describe parameters in SQL statements and in stored procedure calls. To use this option, Opti- mizePrepare must be set to 0 or 1, and the SQL statement must not include long param- eters. This attribute should be set to 1 when using Microsoft Remote Data Objects (RDO) to access data.
EnableQuotedIdentifiers (EQI)	EnableQuotedIdentifiers={0 1}. Enables quoted identifiers.
	When set to 0 (the initial default), quoted identifiers are disabled.
	When set to 1, quoted identifiers are enabled.
FailoverNetworkAddress (FNA)	Specifies the address of the High Availability (HA) Failover server to be used in the event of a connection loss. The driver detects the dropped connection and automatically reconnects to the HA Failover server specified by this attribute. This attribute is valid only for Sybase version 12 or higher servers that have the High Availability Failover feature enabled.
	See the description of the Network Address attribute for an explanation of valid values.
InitializationString (IS)	InitializationString={Sybase set commands;}. Supports the execu- tion of Sybase commands at connect time. Multiple commands must be separated by semicolons.
Language (LANG)	The national language. This language must be installed on the Sybase server.
	The initial default is English.

Attribute	Description
LogonID (UID)	The default logon ID used to connect to your Sybase database. This ID is case-sensitive. A logon ID is required only if security is enabled on your database. If so, contact your system administrator to get your logon ID.
NetworkAddress (NA)	The network address depends on which net- work protocol is chosen under Network Li- brary Name and on the Sybase server. If you have chosen Winsock, specify an IP ad- dress as follows: "servername-or-IP-address, port-number". For example, if your network supports named servers, you may specify an address such as "Sybaseserver, 5000". You may also specify the IP address directly such as "199.226.224.34, 5000".
	If you have chosen NamedPipes as the net- work protocol, you must specify the pipe address of the server. For example, "\\ma- chine1\sybase\pipe\query".
NetworkLibraryName (NLM)	NetworkLibrary- Name={Winsock NamedPipes}. The name of the network library. This specifies which network protocol to use. The initial default is Winsock.
	This option has no effect on UNIX; on UNIX, TCP/IP is used.

Attribute	Description
OptimizePrepare (OP)	OptimizePrepare= $\{0 1 2 3\}$. Determines whether stored procedures are created on the server for calls to SQLPrepare.
	For use with MobiLink, set this parameter to 2.
	When set to 0, stored procedures are created for every call to SQLPrepare. This setting can result in decreased performance when processing statements that do not contain parameters.
	When set to 1 (the initial default), the driver creates stored procedures only if the statemen contains parameters. Otherwise, the statemen is cached and run directly at the time of SQLExecute.
	When set to 2, stored procedures are never created. The driver caches the statement, executes it directly at the time of SQLExecute and reports any syntax or similar errors at the time of SQLExecute.
	When set to 3, stored procedures are never created. This is identical to value 2 except tha any syntax or similar errors are returned at th time of SQLPrepare instead of SQLExecute.
	Use this setting only if you must have syntax errors reported at the time of SQLPrepare.

Attribute	Description
PacketSize (PS)	When set to -1, the driver computes the maximum allowable packet size on the first connect to the data source and saves the value in the system information.
	When set to 0 (the initial default), the driver uses the default packet size as specified in the Sybase server configuration.
	When set to x, an integer from 1 to 1024, the driver uses a packet size represented by x times 512 bytes. For example, PacketSize=6 means to set the packet size to 6 * 512 bytes (3072 bytes).
	To take advantage of this connection attribute, you must configure the Sybase server for a maximum network packet size greater than or equal to the value you specified for PacketSize. For example:
	sp_configure "maximum network packet size", 5120 reconfigure Restart Sybase Server
	NOTE: The ODBC specification specifies a connect option, SQL_PACKET_SIZE, that offers this same functionality. To avoid conflicts with applications that may set both the connection string attribute and the ODBC connect option, they have been defined as mutually exclusive. If PacketSize is specified, you will receive a message "Driver Not Capable" if you attempt to call SQL_PACKET_SIZE. If you do not set PacketSize, then application calls to SQL_PACKET_SIZE are accepted by the driver.
Password (PWD)	A case-sensitive password.
Attribute	Description
--------------------------------------	--
RaiseErrorPositionBehavior (REPB)	RaiseErrorPositionBehavior={0 1}. Speci- fies when the error is returned and where the cursor is positioned when raiserror is encountered.
	When set to 0 (the initial default), raiserror is handled separately from surrounding state- ments. The error is returned when raiserror is processed via SQLExecute, SQLExecDirect, or SQLMoreResults. The result set is empty.
	When set to 1 (MS compatible), raiserror is handled with the next statement. The error is returned when the next statement is processed and the cursor is positioned on the first row of subsequent result set. This could result in multiple raiserrors being returned on a single execute.
SelectMethod (SM)	SelectMethod={0 1}. Determines whether database cursors are used for Select statements.
	When set to 0 (the initial default), database cursors are used. In some cases performance degradation can occur when performing large numbers of sequential Select statements be- cause of the amount of overhead associated with creating database cursors.
	When set to 1, Select statements are run di- rectly without using database cursors, and the data source is limited to one active statement.

Attribute	Description
TightlyCoupled Distributed- Transactions (TCDT)	TightlyCoupledDistributedTransac- tions={0 1}. Determines whether the driver uses tightly coupled distributed transactions when connected to an Adaptive Server Enter- prise version 12 or higher database. When set to 1 (the initial default), the driver uses this type of transaction and multiple connections within the same distributed transaction do not obey each other's locks.
	When set to 0, the overall performance of the driver is better, but multiple connections within the same distributed transaction may hang each other because the connections do not obey each other's locks.
	This attribute is valid only when the driver is enlisted in a distributed transaction or when it is connected to a Sybase version 12 or higher database. Otherwise, this attribute is ignored.
WorkstationID (WKID)	The workstation ID used by the client.

Data types

The following table shows how the Sybase data types are mapped to the standard ODBC data types.

Adaptive Server Enterprise	ODBC
binary	SQL_BINARY
bit	SQL_BIT
char	SQL_CHAR
datetime	SQL_TYPE_TIMESTAMP
decimal	SQL_DECIMAL
float	SQL_FLOAT
image	SQL_LONGVARBINARY
int	SQL_INTEGER
money	SQL_DECIMAL
numeric	SQL_NUMERIC
real	SQL_REAL
smalldatetime	SQL_TYPE_TIMESTAMP
smallint	SQL_SMALLINT
smallmoney	SQL_DECIMAL
sysname	SQL_VARCHAR
text	SQL_LONGVARCHAR
timestamp	SQL_TYPE_TIMESTAMP
tinyint	SQL_TINYINT
varbinary	SQL_VARBINARY
varchar	SQL_VARCHAR

Adaptive Server Enterprise 12.5

The iAnywhere Solutions ODBC driver supports extended new limits (XNL) for character and binary columns—columns with lengths greater than 255.

Supported features

This section lists features supported by the ODBC driver for Sybase.

Support for query timeout

The Adaptive Server Enterprise driver supports the QUERY_TIMEOUT statement attribute on Windows only.

Number of connections and statements supported

The Sybase database system supports multiple connections and multiple statements per connection. If SelectMethod=1, Sybase data sources are limited to one active statement in manual commit mode.

CHAPTER 3

iAnywhere Solutions ODBC Driver for Oracle Wire Protocol

About this chapter	iAnywhere Solutions ODBC driver for Oracle Wire Protocol (the "Oracle Wire Protocol driver") supports Oracle 8.1.6 and higher database systems (including Oracle 9i). The Oracle Wire Protocol driver is currently supported in AIX 4.3.3 only.		
Contents	Торіс:	page	
	Driver requirements	32	
	Configuring data sources	33	
	Connecting to a Data Source Using a Logon Dialog Box	36	
	Connecting to a data source using a connection string	37	
	Data types	42	
	Unicode support	43	
	Default Unicode Mapping	44	
	Stored procedure results	45	
	Supported features	46	

Driver requirements

There are no client requirements for the Oracle Wire Protocol driver.

Configuring data sources

On Windows, data sources are configured and modified through the ODBC Administrator.

In the UNIX environment, there is no ODBC Administrator. To configure a data source in the UNIX environment, you must edit the system information file using the attributes in "Values for AppCodePage Connection String Attribute" on page 69. You must also edit this file to perform a translation.

For more information, see "The system information file (.odbc.ini)" on page 4.

* To configure an Oracle Wire Protocol data source

- 1. Start the ODBC Administrator to display a list of data sources.
- If you are configuring an existing data source, select the data source name and click Configure to display the ODBC Oracle Wire Protocol Driver Setup dialog box.

If you are configuring a new data source, click Add to display a list of installed drivers. Select the Oracle Wire Protocol driver of your choice and click Finish to display the ODBC Oracle Wire Protocol Driver Setup dialog box.

NOTE: The General tab displays only fields that are required for creating a data source. The fields on all other tabs are optional, unless noted otherwise.

- 3. On the General tab, provide the following information and then click Apply.
 - ◆ Data Source Name Type a string that identifies this Oracle Wire Protocol data source configuration in the system information. Examples include "Accounting" or "Oracle-Serv1."
 - **Description** Type an optional long description of a data source name. For example, "My Accounting Database" or "Oracle on Server number 1."
 - ♦ Host Type either the name or the IP address of the server to which you want to connect. For example, if your network supports named servers, you can specify a server name such as Oracleserver. Or, you can specify an IP address such as 199.226.224.34.
 - **Port number** Type the port number of your Oracle listener. Check with your database administrator for the correct number.
 - **SID** Type the Oracle System Identifier that refers to the instance of Oracle running on the server.

- 4. Optionally, click the Advanced tab to specify data source settings. On this tab, provide any of the following optional information; then, click Apply.
 - ◆ Default User Name Type the default user name used to connect to your Oracle database. A default user name is required only if security is enabled on your database. Your ODBC application may override this value or you may override this value in the logon dialog box or connection string.
 - ◆ Array Size Type the number of bytes the driver uses for fetching multiple rows. Values can be an integer from 1 up to 4 GB; the default is 60000. Larger values increase throughput by reducing the number of times the driver fetches data across the network. Smaller values increase response time, as there is less of a delay waiting for the server to transmit data.
 - ◆ Lock Timeout Type 0, -1, or any integer value greater than 0. The value 0 specifies that Oracle does not wait for a lock to be freed before raising an error when processing a Select...For Update statement. The value -1 waits forever. When connected to an Oracle9i server, you can specify the number of seconds to wait by setting this option to an integer greater than 0. If you are connected to an Oracle8i or lower server, any value greater than 0 is equivalent to the value -1.
 - ◆ Default Buffer Size for Long/LOB Columns (in Kb) Type an integer value that specifies the maximum length of data fetched from Long/LOB columns. The value must be in multiples of 1024 (for example, 1024, 2048). The default is 1024 KB. You will need to increase this value if the total size of any long data exceeds 1 MB.
 - ♦ Local Timezone Offset This feature is not supported. Do not enter anything.
 - Enable Timestamp with Timezone This feature is not supported. Do not select it. By default, the check box is not selected.
 - ◆ Catalog Options Select this check box if you want the result column REMARKS for the catalog functions SQLTables and SQLColumns, and the result column COLUMN_DEF for the catalog function SQLColumns to have meaning for Oracle. Selecting this box reduces the performance of your queries. By default, the check box is not selected, which returns SQL_NULL_DATA for the result columns COLUMN_DEF and REMARKS.
 - Enable SQLDescribeParam Select this check box to enable the SQLDescribeParam function, which results in all parameters being described with a data type of SQL_VARCHAR. This option should be selected when using Microsoft Remote Data Objects (RDO) to access data. By default, the check box is not selected.

- ♦ Application Using Threads This check box must be selected. Selecting this option ensures that the driver works with multi-threaded applications. By default, the check box is selected.
- **Procedure Returns Results** Select this check box to enable the driver to return result sets from stored procedures/functions. If this check box is selected and you execute a stored procedure that does not return result sets, you will incur a small performance penalty. By default, the check box is not selected.
- Enable Static Cursors for Long Data: Select this check box to enable the driver to support long columns when using a static cursor. Selecting this check box causes a performance penalty at the time of execution when reading long data. By default, the check box is not selected.

NOTE: You must select this check box if you want to persist a result set that contains LONG data into an XML data file.

- Use Current Schema for SQLProcedures Select this check box to specify only the current user when executing SQLProcedures. When this check box is selected (the default), the call for SQLProcedures is optimized, but only procedures owned by the user are returned.
- Catalog Functions Include Synonyms By default, the check box is selected.
- **Describe at Prepare** Select this check box to enable the driver to describe the SQL statement at prepare time. By default, the check box is not selected.
- **Translate** Click Translate to display the Select Translator dialog box, which lists the translators specified in the ODBC Translators section of the system information. DataDirect provides a translator named "OEM TO ANSI" that translates your data from the IBM PC character set to the ANSI character set. Select a translator; then, click OK to close this dialog box and perform the translation.
- 5. At any point during the configuration process, you can click Test Connect to attempt to connect to the data source using the connection properties specified in the Driver Setup dialog box. A logon dialog box is displayed. Note that the information you enter in the logon dialog box during a test connect is not saved. If the driver can connect, it releases the connection and displays a "connection established" message. Click OK. If the driver cannot connect because of an improper environment or incorrect connection value, it will display an appropriate error message. Click OK.

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.

In this dialog box, perform the following steps:

In the Host field, type either the name or the IP address of the server to which you want to connect.

In the Port Number field, type the number of your Oracle listener. Check with your database administrator for the correct number.

In the SID field, type the Oracle System Identifier that refers to the instance of Oracle running on the server.

If required, type your Oracle user name.

If required, type your Oracle password.

Click OK to log on to the Oracle database installed on the server you specified and to update the values in the system information.

NOTE: Oracle has a feature that allows you to connect to Oracle via the operating system user name and password. To connect, use a slash (/) for the user name and leave the password blank. To configure the Oracle server, refer to the Oracle server documentation.

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 for Oracle is:

DSN=Accounting;SRVR=QESRVR;UID=JOHN;PWD=XYZZY

If the server name contains a semicolon, enclose it in quotation marks:

DSN=Accounting;SRVR="QE;SRVR";UID=JOHN;PWD=XYZZY

To configure a data source in the UNIX environment, you must edit the system information file. This file accepts only long names for attributes.

For more information, see "Quick start for UNIX" on page 4.

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.

Most settings can be left at their default values. For a list of key driver settings when using MobiLink, see "Key driver settings for MobiLink" on page 9.

Attribute	Description
AppCodePage (ACP)	Valid values for this attribute are listed in "Values for AppCodePage Connection String Attribute" on page 69. The code page that you specify must be the same as the code page used by your application.
	the application's code page by checking for an AppCodePage value in the following order:
	 In the connection string
	 In the DataSource section of the system file (odbc.ini)
	 In the ODBC section of the system file (odbc ini)
	If no AppCodePage value is found, the driver uses the default value of 1 (ISO 8859-1 Latin-1).
ApplicationUsingTh- reads (AUT)	ApplicationUsingThreads= $\{0 1\}$. Ensures that the driver works with multi-threaded applications.
	When set to 1 (the initial default), the driver is thread-safe.
	When using the driver with single-threaded applications, you can set this option to 0 to avoid additional processing required for ODBC thread-safety standards.
	Set to 1 for DSNs used with MobiLink server.
ArraySize (AS)	The number of bytes the driver uses for fetching multiple rows. Values can be an integer from 1 up to 4 GB. Larger values increase throughput by reducing the number of times the driver fetches data across the network. Smaller values increase response time, as there is less of a delay waiting for the server to transmit data. The initial default is 60 000
	110 milliar default 15 00,000.

Attribute	Description
CatalogOptions (CO)	CatalogOptions={0 1}. Determines whether the result column REMARKS for the catalog functions SQLTables and SQLColumns and COLUMN_DEF for the catalog function SQL- Columns have meaning for Oracle. If you want to obtain the actual default value, set CO=1. The initial default is 0.
DataSourceName (DSN)	A string that identifies an Oracle data source con- figuration in the system information. Examples include "Accounting" or "Oracle-Serv1."
DescribeAtPrepare (DAP)	DescribeAtPrepare= $\{0 1\}$. Determines whether the driver describes the SQL statement at prepare time.
	When set to 0 (the initial default), the driver does not describe the SQL statement at prepare time.
EnableDescribeParam (EDP)	EnableDescribeParam={0 1}. Determines whether the ODBC API function SQLDe- scribeParam is enabled, which results in all parameters being described with a data type of SQL_VARCHAR.
	This attribute should be set to 1 when using Microsoft Remote Data Objects (RDO) to access data. The initial default is 0.
EnableStatic CursorsFor- LongData (ESCLD)	EnableStaticCursorsForLongData= $\{0 1\}$. De- termines whether the driver supports long columns when using a static cursor. Using this attribute causes a performance penalty at the time of execution when reading long data.
	The initial default is 0.
HostName (HOST)	HostName={servername IP_address}. Identi- fies the Oracle server to which you want to con- nect. If your network supports named servers, you can specify a host name such as Oracle- server. Otherwise, specify an IP address such as 199.226.224.34.

Attribute	Description
LockTimeOut (LTO)	LockTimeOut={0 -1}. Determines whether Oracle should wait for a lock to be freed before raising an error when processing a SelectFor Update statement.
	When set to 0, Oracle does not wait.
	When set to -1 (the initial default), Oracle waits forever.
LogonID (UID)	The default logon ID (user name) that the appli- cation uses to connect to your Oracle 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.
Password (PWD)	The password that the application uses to connect to your Oracle database.
PortNumber (PORT)	Identifies the port number of your Oracle listener. The initial default value is 1521. Check with your database administrator for the correct number.
ProcedureRetResults (PRR)	ProcedureRetResults={0 1}. Determines whether the driver returns result sets from stored procedure functions.
	When set to 0 (the initial default), the driver does not return result sets from stored procedures.
	When set to 1, the driver returns result sets from stored procedures. When set to 1 and you execute a stored procedure that does not return result sets, you will incur a small performance penalty. For details, see "Stored procedure results" on page 45.
Sid (SID)	The Oracle System Identifier that refers to the instance of Oracle running on the server. This item is required when connecting to servers that support more than one instance of an Oracle database.

Attribute	Description
UseCurrentSchema (UCS)	UseCurrentSchema={0 1}. Determines whether the driver specifies only the current user when executing SQLProcedures. When set to 0, the driver does not specify only
	the current user.
	When set to 1 (the initial default), the call for SQLProcedures is optimized, but only proce-
	dures owned by the user are returned.

Data types

The following table shows how the Oracle data types are mapped to the standard ODBC data types.

Oracle	ODBC
Bfile	SQL_LONGVARBINARY ^{1,2}
Blob	SQL_LONGVARBINARY ²
Char	SQL_CHAR
Clob	SQL_LONGVARCHAR ²
Date	SQL_TYPE_TIMESTAMP
Long	SQL_LONGVARCHAR
Long Raw	SQL_LONGVARBINARY
Number	SQL_DOUBLE
Number(p,s)	SQL_DECIMAL
Raw	SQL_VARBINARY
Varchar ¹	SQL_VARCHAR

 1 Valid when connecting to Oracle 8 servers; these data types support output parameters to stored procedures.

The Oracle Wire Protocol driver does not support any Abstract Data Types. When the driver encounters an Abstract Data Type during data retrieval, it will return an Unknown Data Type error (SQL State HY000).

Unicode support

The Oracle Wire Protocol driver automatically determines whether the Oracle database is a unicode database.

The Oracle Wire Protocol driver maps the Oracle data types as follows:

Oracle Data Type	Mapped to
Char	SQL_WCHAR
Varchar2	SQL_WVARCHAR
Long	SQL_WLONGVARCHAR

CLOB data types are not supported.

This driver supports the Unicode ODBC function calls, called W (Wide) calls (for example, SQLConnectW). These calls are used to accept Unicode datastreams.

Default Unicode Mapping

The default Unicode mapping for an application's SQL_C_WCHAR variable is as follows:

.

Platform	Default Unicode Mapping
Windows	UTF-16
AIX	UTF-8
Solaris	UTF-8
Linux	UTF-8

Stored procedure results

When the connection option Procedure Returns Results is active, the driver returns result sets from stored procedures/functions. In addition, SQLGetInfo(SQL_MULT_RESULTS_SETS) will return Y and SQLGetInfo(SQL_BATCH_SUPPORT) will return SQL_BS_SELECT_PROC. If this option is on and you execute a stored procedure that does not return result sets, you will incur a small performance penalty.

This feature requires that stored procedures be in a certain format. First, a package must be created to define all of the cursors used in the procedure, then the procedure can be created using the new cursor. For example:

```
Create or replace package GEN_PACKAGE as

CURSOR G1 is select CHARCOL from GTABLE2;

type GTABLE2CHARCOL is ref cursor return Gl%rowtype;

end GEN_PACKAGE;

Create or replace procedure GEN_PROCEDURE1 (

rset IN OUT GEN_PACKAGE.GTABLE2

CHARCOL, icol INTEGER) as

begin

open rset for select CHARCOL from GTABLE2

where INTEGERCOL <= icol order by INTEGERCOL;

end;
```

For more information, consult your Oracle SQL manual.

Ŧ

Supported features

This section lists features supported by the iAnywhere Solutions 9 — Oracle Wire Protocol ODBC driver.

Isolation and lock levels supported

Oracle supports isolation level 1 (read committed) and isolation level 3 (serializable isolation). Oracle supports record-level locking.

ODBC conformance level

For a list of the API functions supported by the Oracle Wire Protocol driver, see "ODBC Functions" on page 61.

The Oracle Wire Protocol driver also support the following functions:

- ♦ SQLColumnPrivileges
- SQLDescribeParam (if EnableDescribeParam=1)
- ♦ SQLForeignKeys
- ♦ SQLPrimaryKeys
- SQLProcedures
- ♦ SQLProcedureColumns
- ♦ SQLSetPos
- ♦ SQLTablePrivileges

The drivers support the core SQL grammar.

Number of connections and statements supported

The Oracle driver supports multiple connections and multiple statements per connection.

CHAPTER 4

iAnywhere Solutions ODBC driver for DB2

About this chapter	This chapter describes how to configure and use the iAnywhere Solutions ODBC driver for IBM DB2 databases.	
Contents	Торіс:	page
	Introduction	48
	Configuring data sources	49
	Connecting to a data source using a logon dialog box	50
	Connecting to a data source using a connection string	51
	Data types	58
	Supported features	59

Introduction

iAnywhere Solutions ODBC driver for DB2 (the "DB2 driver") supports DB2 Universal Database Versions 6 and 7 on UNIX, and DB2 for OS/390 Versions 6 and 7. This driver is not available on Windows platforms.

Driver requirements

The server requirement for all platforms is the same. The DB2 database must be installed as the Server Version (not the Local Version).

There are no client requirements for the DB2 driver.

Configuring data sources

To configure a data source in the UNIX environment, you must edit the system information file using the attributes in "Values for AppCodePage Connection String Attribute" on page 69. You must also edit this file to perform a translation.

For more information, see "Quick start for UNIX" on page 4.

Creating DB2 bind Users must create the DB2 bind packages on every server to which they intend to connect with the driver. The driver will not work properly with any server that does not have the packages created. The UNIX version of the driver is provided with a program that creates the bind package. It is the equivalent of the Create Package button on the Bind tab of the DB2 driver setup. To bind a package from a command shell, enter:

bind18 dsn

where *dsn* is the ODBC data source name. You are prompted for a user ID and password if they are not stored in the system information file.

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. For DB2, the dialog box is as follows:

In this dialog box, perform the following steps:

- 1. In the IP Address field, type the IP address of the machine where the catalog tables are stored. Specify the address using the machine's numeric address (for example, 123.456.78.90) or specify its host name. If you enter a host name, the driver must find this name (with the correct address assignment) in the HOSTS file on the workstation or in a DNS server.
- 2. In the TCP Port field, type the port number that is assigned to the DB2 server on the machine where the catalog tables are stored. Specify either this port's numeric address or its service name (5179 is the default port address). If you specify a service name, the driver must find this name (with the correct port assignment) in the SERVICES file on the workstation.
- 3. If you are running DB2 on OS/390, perform the following operations.
 - In the Location field, type the DB2 location name. Use the name defined during the local DB2 installation.
 - ♦ In the Collection field, type the name that identifies a group of packages. These packages include the Connect ODBC for DB2 driver packages. The default is DATADIRECT00.
 - Type the name of the database to which you want to connect in the Database field.
 - If required, type your logon ID in the User Name field.
 - If required, type your password in the Password field.
- 4. Click OK to complete the logon and to update the values in the system information.

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 for DB2 is:

DSN=DB2MVS;LOC=TESTMVSDB2;UID=JOHN;PWD=XYZZY

To configure a data source in the UNIX environment, you must edit the system information file. This file accepts only long names for attributes.

For more information, see "Quick start for UNIX" on page 4.

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.

Most settings can be left at their default values. For a list of key driver settings when using MobiLink, see "Key driver settings for MobiLink" on page 9. For other information, about using ODBC drivers with MobiLink, see http://www.ianywhere.com/developer/technotes/odbc_mobilink.html.

Attribute	Description
Action	Can be Create or Replace. The first time you bind a package for a data source (using bind18), set this to Create to create a package. To change the package, set it to Replace.
AddStringTo CreateTable (ASCT)	A string that is automatically added to all Create Table statements. This field is primarily for users who need to add an "in database" clause.

Attribute	Description	
AppCodePage (ACP)	Valid values for this attribute are listed in "DB2 Code Page Values" on page 59. The code page that you specify must be the same as the code page used by your application.	
	The driver on UNIX determines the value of the applica- tion's code page by checking for an AppCodePage value in the following order:	
	• In the connection string	
	• In the DataSource section of the system file (odbc.ini)	
	• In the ODBC section of the system file (odbc.ini)	
	If no AppCodePage value is found, the driver uses the default value of 1 (ISO 8859-1 Latin-1).	
CharSubType- Type (CST)	CharSubTypeType={SYSTEM DEFAULT SBCS MBCS DBCS}.	
	Determines which character set is specified.	
	When set to SYSTEM_DEFAULT, the system default value of the DB2 location to which you are binding is used.	
	When set to SBCS, a single-byte character set is used.	
	When set to MBCS, a mixed-byte character set is used.	
	When set to DBCS, a double-byte character set is used.	
CloseConversa- tion (CC)	CloseConversation={DEALLOC AT_COMMIT}. Determines when the DB2 driver closes an LU 6.2 conversation.	
	When set to DEALLOC (the initial default), the conver- sation is closed when the client application terminates. Use this value unless you are tuning the system for OLTP applications, or you want to prevent an application from using host resources when a client leaves it idle for an extended period of time.	
	When set to AT_COMMIT, the conversation is closed after the client application executes a COMMIT statement.	
Collection (COL)	A name that identifies a group of packages. These pack- ages include the Connect ODBC for DB2 driver packages. The default is DATADIRECT00. This attribute is valid only if you are connecting to DB2 on OS/390.	

Attribute	Description
Conversation- Type (CT)	ConversationType={SINGLE_BYTE MIXED_BYTE}. Determines the type of system codes used by DB2. This value indicates the byte format of character data in the command area of a DB2 packet sent from the remote host. When set to SINGLE BYTE, the remote host uses a
	single-byte system code page (as specified in DSNZ-PARMS on DB2/MVS, for example).
	When set to MIXED_BYTE, the remote host is DB2 3.1, and DSNZPARMS specifies a double-byte code page where single-byte and mixed-byte values are not allowed.
Database (DB)	The name of the database to which you want to connect.
DataSource- Name (DSN)	A string that identifies a DB2 data source configuration in the system information. Examples include "Accounting" and "DB2-Serv1".
DB2AppCodePag (DACP)	Valid values for this attribute are discussed in "DB2 Code e Page Values" on page 59. Specifies a code page that represents the character set that you want data returned to your application. For example, if you are connected to a database in which the character data is stored in the EBCDIC character set but you want the data returned in the LATIN-1 character set, you would set this attribute to 1252.
	On UNIX, the default value is 1252 (LATIN-1).
DecimalDelim- iter (DD)	DecimalDelimiter={COMMA PERIOD}. Determines how the decimal point is represented. This is a required field if the DBRM file contains a decimal literal that does not match the host system's default value.
	When set to COMMA, the decimal point is represented with a comma. For example, 3,17.
	When set to PERIOD, the decimal point is represented with a period. For example, 3.17.
DecimalPreci-	Set this attribute to 31 for use with MobiLink.
sion (DP)	DecimalPrecision={15 31}. Determines the number of places following the decimal point that will be calculated.
	When set to 15 (the initial default), 15 decimal places are calculated.
	When set to 31, 31 decimal places are calculated.

Attribute	Description	
DynamicSec- tions (DS)	The number of statements that the DB2 driver package can prepare for a single user. The initial default is 32.	
GrantAuthid (GA)	A value that determines to whom execute privileges are granted.	
	The default value is grant execute privileges on the package to PUBLIC.	
GrantExecute (GE)	GrantExecute= $\{0 1\}$. Indicates whether or not to grant privileges on the package that you are creating.	
	When set to 0, privileges are not granted.	
	When set to 1, privileges are granted.	
IPAddress (IP)	The IP address of the machine where the catalog tables are stored. Enter the address using the machine's numeric address (for example, 123.456.78.90) or type its address name. If you enter an address name, the driver must find this name (with the correct address assignment) in the HOSTS file on the workstation or in a DNS server.	
IsolationLevel (IL)	IsolationLevel={ALL CHANGE CURSOR STABILITY NO_COMMIT REPEATABLE_READ}. Specifies the method by which locks are acquired and released by the system. Valid values are: All Prevents any other process from accessing data that your application has read or modified. All read or modified data is locked until the end of the transaction.	
	Change Allows other processes to read from the database. Only modified data is locked until the end of the transaction.	
	Cursor Stability Allows other processes to change a row that your application read if the cursor is not on the row that you want to change. Prevents other processes from changing records that your application has changed until your program commits them or terminates. Prevents your program from reading a modified record that has not been committed by another process.	

Attribute	Description
IsolationLevel (IL) (cont.)	No Commit Allows your program to read modified records even if they have not been committed by another person.
	Repeatable Read Prevents other processes from chang- ing records that are read or changed by your application (including phantom records) until your program commits them or terminates. Prevents the application from reading modified records that have not been committed by another process. If your program opens the same query during a single unit of work under this isolation level, the results table will be identical to the previous table; however, it can contain updates made by your program.
Location (LOC)	A path that specifies the DB2 location name. Use the name defined during the local DB2 installation. This attribute is valid and required only if you are connecting to DB2 on OS/390.
LogonID (UID)	The default logon ID used to connect to your DB2 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.
	For DB2 on UNIX, normal UNIX security is used. The LogonID value is your UNIX user ID.
Maximum- Clients (MC)	An integer that specifies the maximum number of concur- rent client sessions that the DB2 driver can carry. If the driver is used with an application server, select a value that will accommodate the number of users who will simulta- neously access the host system through the DB2 driver. In a client configuration, use a small value to reduce the DB2 driver's memory requirements. The default is 10.
Package (PCK)	The name of the package that the driver uses to process static and dynamic SQL for applications that use this data source definition. The default name is DEFxx, where xx is the version number.
PackageOwner (PO)	The AuthID assigned to the package. This DB2 AuthID must have authority to execute all the SQL in the package.
Password (PWD)	A password used to connect to your DB2 database.

Attribute	Description
QueryBlock- Size (QBS)	The number of rows the driver retrieves when fetching from the server. This is not the number of rows given to the user.
	The initial default is 8 rows.
ResourceRe- lease (RR)	ResourceRelease={DEALLOCATION COMMIT}. Spec- ifies the release of database resources.
	When set to DEALLOCATION, database resources are released when the connection is terminated.
	When set to COMMIT, database resources are released after a commit and provide a high level of concurrency.
ServerCharSet (SCS)	Valid values for this attribute are listed in "DB2 Code Page Values" on page 59. Specifies a code page that represents the character set used to store character data in the database.
	On UNIX, the default value is 1252 (LATIN-1) for UDB databases and 37 (EBCDIC) for MVS databases.
	NOTE: The code page (character set) you specify for this attribute must be the same as the character set used by the database. If the values are different, you will fail to connect and receive an error message, for example:
	[MERANT] [ODBC DB2 Wire Protocol driver] ServerCharSet '37' is incompatible with the Database character set '1252'
StringDelimiter (SD)	StringDelimiter={SINGLE_QUOTE DOUBLE QUOTE}. Specifies the type of quotation marks (single or double) used to represent constant string values that are referenced by the SQL in the DBRM. This option must match the option used to delimit the literal strings referenced by your embedded SQL.
TCPPort (PORT)	The port number that is assigned to the DB2 server on the machine where the catalog tables are stored. Specify this port's numeric address or its name (5179 is the default port address). If you specify a port name, the driver must find this name (with the correct port assignment) in the SERVICES file on the workstation.

Attribute	Description		
Trace (TR)	Trace = $\{0 1\}.$		
	Determines whether a trace file is created in the applica- tion's directory. When Trace=1, a trace file is created. The trace file name is packet.xxx, where xxx is an incremental number starting with 000.		
UserBufferSize (UBS)	The size in kilobytes of the bulk packet that the DB2 driver uses to download data from the host. Valid values are 1 to 63. For most environments, the default value of 32 is sufficient; however, adjusting the value can optimize some client applications as follows:		
	• For client applications that frequently download large amounts of data, a large buffer size can improve response time.		
	• For client applications that perform brief online trans- actions, a small buffer will maximize memory on the machine where the DB2 driver is installed.		
WithHold	Set this attribute to 1 for use with MobiLink.		
(WH)	WithHold={0 1}. Specifies the cursor behavior for the application used with this data source. Either DB2 closes all open cursors (Delete cursors) after a commit or rollback, or leaves them open (Preserve cursors). When set to 1, the cursor behavior is Preserve. When set to 0, the cursor behavior is Delete (the default).		
	If you are using the Static Bind Administrator and you want your package to use cursors WITH HOLD, you must set this attribute to 1. Note that any application using this package must use a data source with this attribute set to 1.		
	When set to 1, the Static Bind Administrator automatically adds the WITH HOLD clause to queries that it puts in the application's database resource module (DBRM). The WITH HOLD clause prevents DB2 from automatically closing the cursor when the application executes a Commit statement.		
	SQLGetInfo() returns SQL_CB_ PRESERVE for SQL_COMMIT_CURSOR_ BEHAVIOR		
	When set to 0, SQLGetInfo() returns SQL_CB_DELETE. For information about this function, refer to the Microsoft ODBC API.		

Data types

The following table shows how the DB2 data types map to the standard ODBC data types.

DB2	ODBC	
Char	SQL_CHAR	
Char() for Bit Data	SQL_BINARY	
Clob*	SQL_LONGVARCHAR	
Date	SQL_TYPE_DATE	
Decimal	SQL_DECIMAL	
Float	SQL_DOUBLE	
Graphic	SQL_BINARY	
Integer	SQL_INTEGER	
Long Varchar	SQL_LONGVARCHAR	
Long Varchar for Bit Data	SQL_LONGVARBINARY	
Long Vargraphic	SQL_LONGVARBINARY	
Smallint	SQL_SMALLINT	
Time	SQL_TYPE_TIME	
Timestamp	SQL_TYPE_TIMESTAMP	
Varchar	SQL_VARCHAR	
Varchar() for Bit Data	SQL_VARBINARY	
Vargraphic	SQL_VARBINARY	

*Only the first 32Kb of the Clob data type is returned when fetching this data type from DB2 databases. Also, only 32Kb can be inserted and updated on DB2 databases.

Supported features

This section lists features supported by the ODBC driver for DB2.

Stored procedure support

The DB2 driver supports DB2 Remote Procedure Calls (RPCs) with the following restrictions:

- Multiple result sets are not returned; only the first result set is returned.
- RPCs must take an argument list. The driver does not support RPCs that use a SQL descriptor area (SQLDA) data structure to specify the arguments.
- Literals are not supported as stored procedure parameters.

Isolation and Lock Levels Supported

DB2 supports isolation levels 0 (read uncommitted), 1 (read committed), and 2 (repeatable read). It supports record-level locking.

NOTE: An isolation level can be set only before connecting to a DB2 database.

Number of connections and statements supported

The DB2 database system supports multiple connections and multiple statements per connection.

DB2 Code Page Values

The following table lists the most commonly used DB2 code pages and their numerical values. These values are valid for the ServerCharSet connection string attribute. For a complete listing of all valid DB2 code pages and their values, refer to the IBM DB2 Administration Guide.

Code Page Value	Code Page Name
437	IBM CP437 US code set
850	IBM CP850 European code set
912	ISO 8859-2 Latin-2/Eastern Europe
914	ISO 8859-4 Latin/Estonian/Latvian
916	ISO 8859-8 Latin/Hebrew

i.

Code Page Value	Code Page Name
920	ISO 8859-9 Latin-5/Turkish
923	ISO 8859-15 Latin1 with Euro, etc.
950	Traditional Chinese
1089	ISO 8859-6 Latin/Arabic
1200	Unicode Encoding UTF-16
1208	Unicode Encoding UTF-8
1252	ISO 8858-1 Latin-1

Unicode support

The DB2 driver supports Unicode data types if the database was created with a multi-byte character set.

The driver maps the following DB2 data types to Unicode data types:

DB2 Data Type	Mapped to
Char	SQL_WCHAR
Clob*	SQL_WLONGVARCHAR
Long Varchar	SQL_WLONGVARCHAR
Varchar	SQL_WVARCHAR

*Only the first 32 KB of the Clob data type is returned when fetching this data type from DB2 databases. Also, only 32 KB can be inserted and updated on DB2 databases.

APPENDIX A

ODBC Functions

About this appendix	This appendix lists the ODBC functions that you can use in SQL statements.	
Contents	Торіс:	page
	ODBC functions	62

ODBC functions

This section lists the scalar functions that ODBC supports. Your database system may not support all of these functions. See the documentation for your database system to find out which functions are supported.

You can use these functions in SQL statements using the following 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

String functions

The following table 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, SQL_VARCHAR, or SQL_LONGVARCHAR.
- 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.

Function	Returns
ASCII(<i>string_exp</i>)	ASCII code value of the leftmost character of <i>string_exp</i> as an integer.
BIT_LENGTH(<i>string_exp</i>)	The length in bits of the string expression.
CHAR(code)	The character with the ASCII code value specified by <i>code</i> . <i>code</i> should be between 0 and 255; otherwise, the return value is data-source dependent.
Function	Returns
--	--
CHAR_LENGTH(<i>string_</i> - exp)	The length in characters of the string expres- sion, 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(<i>string_exp</i>)	The length in characters of the string expres- sion 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_exp2)	The string resulting from concatenating <i>string_exp2</i> and <i>string_exp1</i> . The string is system dependent.
DIFFERENCE(string_exp1, string_exp2)	An integer value that indicates the differ- ence between the values returned by the SOUNDEX function for <i>string_exp1</i> and <i>string_exp2</i> .
INSERT(string_exp1, start, length, string_exp2)	A string where <i>length</i> characters have been deleted from <i>string_exp1</i> beginning at start and where <i>string_exp2</i> has been inserted into <i>string_exp</i> , beginning at start.
LCASE(<i>string_exp</i>)	Uppercase characters in <i>string_exp</i> converted to lowercase.
LEFT(<i>string_exp</i> , <i>count</i>)	The count of characters of string_exp.
LENGTH(<i>string_exp</i>)	The number of characters in <i>string_exp</i> , excluding trailing blanks and the string termination character.

Function	Returns
LOCATE(<i>string_exp1</i> , <i>string_exp2</i> [<i>,start</i>])	The starting position of the first occurrence of <i>string_exp1</i> within <i>string_exp2</i> . If <i>start</i> is not specified, the search begins with the first character position in <i>string_exp2</i> . If <i>start</i> is specified, the search begins with the character position indicated by the value of <i>start</i> . The first character position in <i>string_exp2</i> is indicated by the value 1. If <i>string_exp1</i> is not found, 0 is returned.
LTRIM(string_exp)	The characters of <i>string_exp</i> , with leading blanks removed.
OCTET_LENGTH(string exp)	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(<i>string_exp</i> , <i>count</i>)	A string composed of <i>string_exp</i> repeated <i>count</i> times.
REPLACE(string_exp1, string_exp2, string_exp3)	Replaces all occurrences of <i>string_exp2</i> in <i>string_exp1</i> with <i>string_exp3</i> .
RIGHT(string_exp, count)	The rightmost count of characters in <i>stringexp</i> .
RTRIM(string_exp)	The characters of <i>string_exp</i> with trailing blanks removed.
SOUNDEX(string_exp)	A data-source-dependent string representing the sound of the words in <i>string_exp</i> .
SPACE(count)	A string consisting of <i>count</i> spaces.
SUBSTRING(string_exp, start, length)	A string derived from <i>string_exp</i> beginning at the character position <i>start</i> for <i>length</i> characters.
UCASE(<i>string_exp</i>)	Lowercase characters in <i>string_exp</i> converted to uppercase.

Numeric functions

The following table 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_BIGINT, 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, SQL_INTEGER, or SQL_BIGINT.

Function	Returns
ABS(numeric_exp)	Absolute value of <i>numeric_exp</i> .
ACOS(float_exp)	Arc-cosine of <i>float_exp</i> as an angle in radians.
ASIN(float_exp)	Arc-sine of <i>float_exp</i> as an angle in radians.
ATAN(float_exp)	Arc-tangent of <i>float_exp</i> as an angle in radians.
ATAN2(float_exp1, float exp2)	Arc-tangent of the x and y coordinates, specified by <i>float_exp1</i> and <i>float_exp2</i> as an angle in radians.
CEILING(numeric_exp)	Smallest integer greater than or equal to <i>numeric_exp</i> .
COS(float_exp)	Cosine of <i>float_exp</i> as an angle in radians.
COT(float_exp)	Cotangent of <i>float_exp</i> as an angle in radians.
DEGREES(numeric_exp)	Number if degrees converted from <i>numeric_exp</i> radians.
EXP(float_exp)	Exponential value of <i>float_exp</i> .

Function	Returns
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 <i>float_exp</i> .
MOD(integer_exp1, integer exp2)	Remainder of <i>integer_exp1</i> divided by <i>integer_exp2</i> .
PI()	Constant value of pi as a floating-point number.
POWER(numeric_exp, integer_exp)	Value of <i>numeric_exp</i> to the power of <i>integer_exp</i> .
RADIANS(numeric_exp)	Number of radians converted from numeric_exp degrees.
RAND([integer_exp])	Random floating-point value using <i>integer_exp</i> 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).
SIGN(numeric_exp)	Indicator of the sign of <i>numeric_exp</i> . If <i>numeric_exp</i> < 0 , -1 is returned. If <i>numeric_exp</i> $= 0$, 0 is returned. If <i>numeric_exp</i> > 0 , 1 is returned.
SIN(float_exp)	Sine of <i>float_exp</i> , where <i>float_exp</i> is an angle in radians.
SQRT(float_exp)	Square root of <i>float_exp</i> .
TAN(float_exp)	Tangent of <i>float_exp</i> , where <i>float_exp</i> 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).

Date and Time Functions

The following table 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.

Function	Returns
CURRENT_DATE()	Current date.
CURRENT_TIME[(time- precision)]	Current local time. The <i>time-precision</i> argument determines the seconds precision of the returned value.
CURRENT TIMESTAMP[(timestamp- precision)]	Current local date and local time as a timestamp value. The <i>timestamp-precision</i> 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 data-source- specific name of the day for the day portion of <i>date_exp</i> .
DAYOFMONTH(<i>date</i> <i>exp</i>)	Day of the month in <i>date_exp</i> as an integer value.
DAYOFWEEK(<i>date_exp</i>)	Day of the week in <i>date_exp</i> as an integer value.
DAYOFYEAR(<i>date_exp</i>)	Day of the year in <i>date_exp</i> as an integer value.
HOUR(time_exp)	Hour in <i>time_exp</i> as an integer value.
MINUTE(time_exp)	Minute in <i>time_exp</i> as an integer value.

Function	Returns
MONTH(<i>date_exp</i>)	Month in <i>date_exp</i> as an integer value.
MONTHNAME(date_exp)	Character string containing the data source-specific name of the month.
NOW()	Current date and time as a timestamp value.
QUARTER(<i>date_exp</i>)	Quarter in <i>date_exp</i> as an integer value.
SECOND(time_exp)	Second in <i>time_exp</i> as an integer value.
TIMESTAM- PADD(interval, integer exp, time_exp)	Timestamp calculated by adding <i>integerexp</i> intervals of type interval to time_exp. The 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, or SQL_TSI_YEAR Fractional seconds are expressed in bil- lionths of a second.
TIMESTAM- PDIFF(interval, time_exp1, time_exp2)	Integer number of intervals of type interval by which <i>time_exp2</i> is greater than <i>time</i> <i>exp1. interval</i> has the same values as TIMESTAMPADD. Fractional seconds are expressed in billionths of a second.
WEEK(date_exp)	Week of the year in <i>date_exp</i> as an integer value.
YEAR(<i>date_exp</i>)	Year in <i>date_exp</i> . The range is data-source dependent.

System functions

The following table lists the system functions that ODBC supports.

Function	Returns
DATABASE()	Name of the database, corresponding to the connection handle (hdbc).
IFNULL(exp, value)	<i>value</i> , if exp is null.
USER()	Authorization name of the user.

APPENDIX B

Values for AppCodePage Connection String Attribute

About this appendix	This appendix contains specific information ab the UNIX environments.	out using Connect ODBC in
Contents	Торіс:	page
	Table of AppCodePage values	70

Table of AppCodePage values

The following table lists valid values for the AppCodePage connection string attribute. This attribute is valid only for Connect ODBC drivers that run on UNIX. See the appropriate individual driver chapter for information about this attribute.

Value	Description
0	ISO 646 7-bit ASCII
1	ISO 8859-1 Latin-1
2	CP 850 - European code page
3	CP 437 - US code page
4	HP ROMAN8
5	Standard Macintosh Roman
6	Shift-JIS proper
7	EUC-JIS encoding
8	Digital UNIX JIS encoding
9	EUC-CNS encoding
10	EUC-GB encoding
11	Microsoft CP 932 = Win32J-DBCS
12	ISO 8859-2 Latin-2 Eastern Europe
13	ISO 8859-5 Latin/Cyrillic
14	ISO 8859-6 Latin/Arabic
15	ISO 8859-7 Latin/Greek
16	ISO 8859-8 Latin/Hebrew
17	ISO 8859-9 Latin-5 Turkish
18	ISO 8859-10 Latin-6 Nordic
19	ISO 8859-3 Latin/Esperanto/Galician
20	ISO 8859-4 Latin/Estonian/Latvian
21	ISO 8859-15 Latin-9 Western Europe with Euro sign
26	Macintosh Cyrillic

Value	Description
27	Macintosh Eastern European
28	Macintosh Greek
29	Macintosh Turkish
30	HP Greek
31	HP Turkish
32	KOI8 - Cyrillic
33	TIS 620 - Thai standard
34	Big5 Traditional Chinese
36	EUC-KSC Korean encoding, similar to CP 949
37	IBM EBCDIC (8859-1 convertible)
273	IBM EBCDIC Germany/Austria
277	IBM EBCDIC Denmark/Norway
278	IBM EBCDIC Finland/Sweden
280	IBM EBCDIC Italian
284	IBM EBCDIC Spain/Latin America
285	IBM EBCDIC U.K.
290	IBM EBCDIC Katakana for DB2
297	IBM EBCDIC France
420	IBM EBCDIC Arabic bilingual
500	IBM EBCDIC Western Europe
737	PC Greek
775	PC Baltic
852	PC Eastern Europe
855	PC Cyrillic
857	PC Turkish
860	PC Portuguese
861	PC Icelandic
862	PC Hebrew

Value	Description
863	PC Canadian French
864	PC Arabic
865	PC Nordic
866	PC Russian
869	PC Greek
870	IBM EBCDIC Eastern Europe
874	Microsoft Thai SB code page
875	IBM EBCDIC Greek
930	Japanese Host merged: CP 290 + CP 300
932	Japanese IBM J-DBCS: CP 897 + CP 301
933	Korean Host merged: CP 833 + CP 834
935	SimpChinese Host merged: CP 836+ CP 837
936	PC Simplified Chinese
937	TradChinese Host merged: CP 037 + CP 835
939	Japanese Host merged: CP 1027 + CP 4396
949	PC (MS) Korean, similar to EUC-KSC
950	PC (MS) Traditional Chinese (~Big5)
954	EUC-JIS
1026	IBM EBCDIC Turkish
1047	MVS Open Edition
1250	MS Windows 3.1 Eastern European
1251	MS Windows 3.1 Cyrillic
1252	MS Windows 3.1 US (ANSI)
1253	MS Windows 3.1 Greek
1254	MS Windows 3.1 Turkish
1255	MS Windows Hebrew
1256	MS Windows Arabic
1257	MS Windows Baltic

Value	Description
1258	MS Windows Vietnamese
5026	CCSID for CP 930 with only 1880 UDC
5035	CCSID for CP 939 with only 1880 UDC

Index

Α

Adaptive Server Enterprise driver	
code pages	20
connection string attributes	20
connections supported	30
data source configuration	17
data source connecting via connect	on
string	19
data source connecting via logon	
dialog box	19
data types	29
driver requirements	16
key ODBC settings	9
query timeout	30
statements supported	30
AIX	
drivers	31
ODBC drivers	4
AppCodePage connection string	
attributes	69

В

bind packages DB2 data sources	11
bind18	
creating bind packages	11
creating DB2 data sources	49

С

code pages	
Adaptive Server Enterprise driver	20
connection string attributes	69
DB2 driver	51
Oracle driver	37
Connect ODBC	
Sybase Wire Protocol	69
connection string attributes	
Adaptive Server Enterprise	20
DB2	51
Oracle	37
connections supported	

DB2	59
Oracle	46
Sybase	30

D

data source	
configuring	33
configuring Adaptive Server	
Enterprise	17
configuring DB2	49
connecting via connection string in	
Adaptive Server Enterprise	19
connecting via connection string in	
DB2	51
connecting via connection string wi	th
Oracle Wire Protocol	37
connecting via logon dialog box for	
Adaptive Server Enterprise	19
connecting via logon dialog box in	
DB2	50
data types	
DB2	58
Oracle Wire Protocol	42
Sybase	29
date and time functions	
ODBC	66
DB2 code page	
values	59
DB2 driver	
code pages	51
connection string attributes	51
connections supported	59
data source configuration	49
data source connection via connecti	on
string	51
data source connection via logon	
dialog box	50
data types	58
driver requirements	48
isolation levels	59
key ODBC settings	11
locking levels	59

statements supported	59
stored procedure support	59
Unicode support	60
documentation	
SQL Anywhere Studio	vi
driver requirements	
Adaptive Server Enterprise driver	16
DB2 driver	48
Oracle Wire Protocol driver	32
drivers	
iAnywhere Solutions ODBC	2
_	

Е

error messages	
general	12
UNIX	13

F

failover	
Adaptive Server Enterprise driver	20
feedback	
providing	ix

Н

high availability	
Adaptive Server Enterprise driver	20

I

A much and Calutions ODDC drives	
Anywhere Solutions ODBC driver	
Adaptive Server Enterprise	15
DB2	47
Oracle	31
IBM DB2 driver	
code pages	51
connection string attributes	51
data source configuration	49
data source connection via connect	ion
string	51
data source connection via logon	
dialog box	50
data types	58
driver requirements	48
key ODBC settings	11
stored procedure support	59
Unicode support	60
isolation levels	

DB2	59
Oracle	46
L	
library path environment variable	7
Linux	
ODBC drivers	4
locking levels	
DB2	59
Oracle	46
М	
Mobil ink	

MobiLink ODBC drivers 1

newsgroups technical support ix numeric functions ODBC 65

0

ODBC	
iAnywhere Solutions drivers	2
MobiLink drivers	2
scalar functions	62
ODBC drivers	
AIX	4
Linux	4
Solaris	4
ODBCINI environment variable	7
Oracle driver	
code pages	37
connection string attributes	37
connections supported	46
isolation levels	46
key ODBC settings	10
locking levels	46
statements supported	46
Oracle Wire Protocol driver	
data source configuration	33
data source connection via conne	ction
string	37
data types	42
default unicode mapping	44
driver requirements	32

unicode mapping	44	DB2 driver	60
unicode support	43	Oracle Wire Protocol driver	43
0		UNIX	
Q		code page connection string attribu	ites
query timeout		driver normes	7
setting for the Adaptive Server		driver	16
Enterprise driver	30	anvironment library noth	10
_		environment ODPCINI	7
R		environment system information fi	10
remote data access		(odbaini)	10
ODBC drivers	1	(.oubc.iii)	47
ODDC unvers	1		12
S		UNIX drivers	15
•		DD2	47
scalar functions		DB2	47
ODBC	62	W	
Solaris		••	
ODBC drivers	4	Windows	
SQL Anywhere Studio		driver names	3
documentation	vi	starting the ODBC Administrator	3
statements supported		Windows drivers	
Adaptive Server Enterprise	30	Adaptive Server Enterprise	16
DB2	59	DB2	47
Oracle	46		
stored procedures			
DB2 driver	59		
stored results			
Oracle Wire Protocol driver	45		
string functions			
ODBC	62		
support			
newsgroups	ix		
system functions			
ODBC support	68		
system information file (.odbc.ini)	4		
т			
4 1 1 1 4			
recumcal support			
time functions	1X		
ume functions			

echnical support	
newsgroups	ix
ime functions	
ODBC	66

U

unicode mapping	
Oracle Wire Protocol driver	44
Unicode support	