

# PROGRESS<sup>®</sup> DATADIRECT<sup>®</sup>

**DataDirect Connect<sup>®</sup> Series** *for ODBC*  
User's Guide

Release 7.1.0

© 2012 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.

These materials and all Progress® software products are copyrighted and all rights are reserved by Progress Software Corporation. The information in these materials is subject to change without notice, and Progress Software Corporation assumes no responsibility for any errors that may appear therein. The references in these materials to specific platforms supported are subject to change.

Actional, Apama, Artix, Business Empowerment, Business Making Progress, Corticon, Corticon (and design), DataDirect (and design), DataDirect Connect, DataDirect Connect64, DataDirect Technologies, DataDirect XML Converters, DataDirect XQuery, DataXtend, Dynamic Routing Architecture, Empowerment Center, Fathom, Fuse Mediation Router, Fuse Message Broker, Fuse Services Framework, IONA, Making Software Work Together, Mindreef, ObjectStore, OpenEdge, Orbix, PeerDirect, Powered by Progress, PowerTier, Progress, Progress DataXtend, Progress Dynamics, Progress Business Empowerment, Progress Empowerment Center, Progress Empowerment Program, Progress OpenEdge, Progress Profiles, Progress Results, Progress Software Business Making Progress, Progress Software Developers Network, Progress Sonic, ProVision, PS Select, RulesCloud, RulesWorld, Savvion, SequeLink, Shadow, SOAPscope, SOAPStation, Sonic, Sonic ESB, SonicMQ, Sonic Orchestration Server, SpeedScript, Stylus Studio, Technical Empowerment, WebSpeed, Xcalia (and design), and Your Software. Our Technology-Experience the Connection are registered trademarks of Progress Software Corporation or one of its affiliates or subsidiaries in the U.S. and/or other countries. AccelEvent, Apama Dashboard Studio, Apama Event Manager, Apama Event Modeler, Apama Event Store, Apama Risk Firewall, AppsAlive, AppServer, ASpen, ASP-in-a-Box, BusinessEdge, Cache-Forward, CloudEdge, DataDirect Spy, DataDirect SupportLink, Fuse, FuseSource, Future Proof, GVAC, High Performance Integration, Integration Everywhere, ObjectStore Inspector, ObjectStore Performance Expert, OpenAccess, Orbacus, Pantero, POSSE, ProDataSet, Progress Arcade, Progress CloudEdge, Progress Cloudware, Progress Control Tower, Progress ESP Event Manager, Progress ESP Event Modeler, Progress Event Engine, Progress RFID, Progress RPM, Progress Responsive Cloud, Progress Responsive Process Management, Progress Software, PSE Pro, SectorAlliance, SeeThinkAct, Shadow z/Services, Shadow z/Direct, Shadow z/Events, Shadow z/Presentation, Shadow Studio, SmartBrowser, SmartComponent, SmartDataBrowser, SmartDataObjects, SmartDataView, SmartDialog, SmartFolder, SmartFrame, SmartObjects, SmartPanel, SmartQuery, SmartViewer, SmartWindow, Sonic Business Integration Suite, Sonic Process Manager, Sonic Collaboration Server, Sonic Continuous Availability Architecture, Sonic Database Service, Sonic Workbench, Sonic XML Server, The Brains Behind BAM, WebClient, and Who Makes Progress are trademarks or service marks of Progress Software Corporation and/or its subsidiaries or affiliates in the U.S. and other countries. Java is a registered trademark of Oracle and/or its affiliates. Any other marks contained herein may be trademarks of their respective owners.

#### Third Party Acknowledgments:

Progress DataDirect Connect for ODBC v7.1, Progress DataDirect Connect64 for ODBC v7.1, Progress DataDirect Connect XE for ODBC v7.1, Progress DataDirect Connect64 XE for ODBC v7.1 each incorporate ICU v4.2.1 from International Business Machines Corporation. Such technology is subject to the following terms and conditions: ICU License - ICU 1.8.1 and later ICU License - ICU 1.8.1 and later COPYRIGHT AND PERMISSION NOTICE. Copyright (c) 1995-2010 International Business Machines Corporation and others. All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder. All trademarks and registered trademarks mentioned herein are the property of their respective owners.

Progress DataDirect Connect for ODBC v7.1, Progress DataDirect Connect64 for ODBC v7.1, Progress DataDirect Connect XE for ODBC v7.1, Progress DataDirect Connect64 XE for ODBC v7.1 each incorporate HSQL database v1.8.0.10 from The HSQL Development Group. Such technology is subject to the following terms and conditions: Copyright (c) 2001-2005, The HSQL Development Group All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of the HSQL Development Group nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL HSQL DEVELOPMENT GROUP, HSQLDB.ORG, OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Progress DataDirect Connect for ODBC v7.1, Progress DataDirect Connect64 for ODBC v7.1, Progress DataDirect Connect XE for ODBC v7.1, Progress DataDirect Connect64 XE for ODBC v7.1 each incorporate OpenLDAP v2.2.8 from The OpenLDAP Foundation. Such technology is subject to the following terms and conditions: The OpenLDAP Public License Version 2.8, 17 August 2003 - Redistribution and use of this software and associated documentation ("Software"), with or without modification, are permitted provided that the following conditions are met: 1. Redistributions in source form must retain copyright statements and notices, 2. Redistributions in binary form must reproduce applicable copyright statements and notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution, and 3. Redistributions must contain a verbatim copy of this document. The OpenLDAP Foundation may revise this license from time to time. Each revision is distinguished by a version number. You may use this Software under terms of this license revision or under the terms of any subsequent revision of the license. THIS SOFTWARE IS PROVIDED BY THE OPENLDAP FOUNDATION AND ITS CONTRIBUTORS "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OPENLDAP FOUNDATION, ITS CONTRIBUTORS, OR THE AUTHOR(S) OR OWNER(S) OF THE SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE (POSSIBILITY OF SUCH DAMAGE. The names of the authors and copyright holders must not be used in advertising or otherwise to promote the sale, use or other dealing in this Software without specific, written prior permission. Title to copyright in this Software shall at all times remain with copyright holders. OpenLDAP is a registered trademark of the OpenLDAP Foundation. Copyright 1999-2003 The OpenLDAP Foundation, Redwood City, California, USA. All Rights Reserved. Permission to copy and distribute verbatim copies of this document is granted.

Progress DataDirect Connect for ODBC v7.1, Progress DataDirect Connect64 for ODBC v7.1, Progress DataDirect Connect XE for ODBC v7.1, Progress DataDirect Connect64 XE for ODBC v7.1 each incorporate OpenSSL v1.0.0d from The OpenSSL Project. Such technology is subject to the following terms and conditions: LICENSE ISSUES ===== The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL, please contact openssl-core@openssl.org. OpenSSL License -----

Copyright (c) 1998-2011 The OpenSSL Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (<http://www.openssl.org/>)"
4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.
5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project.
6. Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>)" THIS SOFTWARE IS PROVIDED BY THE OPENSSL PROJECT "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OPENSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

===== This product includes cryptographic software written by Eric Young ([ey@cryptsoft.com](mailto:ey@cryptsoft.com)). This product includes software written by Tim Hudson ([tjh@cryptsoft.com](mailto:tjh@cryptsoft.com)).

Original SSLeay License ----- Copyright (C) 1995-1998 Eric Young ([ey@cryptsoft.com](mailto:ey@cryptsoft.com)) All rights reserved.

This package is an SSL implementation written by Eric Young ([ey@cryptsoft.com](mailto:ey@cryptsoft.com)). The implementation was written so as to conform with Netscapes SSL. This library is free for commercial and non-commercial use as long as the following conditions are adhered to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson ([tjh@cryptsoft.com](mailto:tjh@cryptsoft.com)). Copyright remains Eric Young's, and as such any copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement: "This product includes cryptographic software written by Eric Young ([ey@cryptsoft.com](mailto:ey@cryptsoft.com))" The word 'cryptographic' can be left out if the routines from the library being used are not cryptographic related :-).
4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement: "This product includes software written by Tim Hudson ([tjh@cryptsoft.com](mailto:tjh@cryptsoft.com))" THIS SOFTWARE IS PROVIDED BY ERIC YOUNG "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The licence and distribution terms for any publicly available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution licence [including the GNU Public Licence.]

Progress DataDirect Connect for ODBC v7.1, Progress DataDirect Connect64 for ODBC v7.1, Progress DataDirect Connect XE for ODBC v7.1, Progress DataDirect Connect64 XE for ODBC v7.1 each incorporate PostgreSQL Database Management System v8.3.3. Such technology is subject to the following terms and conditions: PostgreSQL is released under the BSD license. PostgreSQL Database Management System (formerly known as Postgres, then as Postgres95) Portions Copyright (c) 1996-2005, The PostgreSQL Global Development Group Portions Copyright (c) 1994, The Regents of the University of California Permission to use, copy, modify, and distribute this software and its documentation for any purpose, without fee, and without a written agreement is hereby granted, provided that the above copyright notice and the following two paragraphs appear in all copies. IN NO EVENT SHALL THE UNIVERSITY OF CALIFORNIA BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE UNIVERSITY OF CALIFORNIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. THE UNIVERSITY OF CALIFORNIA SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND THE UNIVERSITY OF CALIFORNIA HAS NO OBLIGATIONS TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS. Why not the GNU General Public License? People often ask why PostgreSQL is not released under the GNU General Public License. The simple answer is because we like the BSD license and do not want to change it. If you are keen to read more about this topic, then please take a look in the Archives at any of the many threads on this subject, but please don't start yet another debate on the subject!

# Table of Contents

<b>List of Tables</b> .....	<b>19</b>
<b>Preface</b> .....	<b>23</b>
Product Platforms .....	23
Product Matrix .....	24
What's New in This Release? .....	25
Using this Book .....	27
Conventions Used in This Book. ....	28
Typographical Conventions. ....	28
Environment-Specific Information .....	29
About the Product Documentation. ....	29
HTML Version .....	29
PDF Version .....	30
Contacting Customer Support .....	31
<b>Part 1: Getting Started</b>	
<b>1 Quick Start Connect</b> .....	<b>35</b>
Configuring and Connecting on Windows .....	35
Setting the Library Path Environment Variable (Salesforce Driver on Windows) ..	35
Configuring a Data Source .....	36
Testing the Connection .....	36
Configuring and Connecting on UNIX and Linux .....	37
Environment Configuration .....	37
Test Loading the Driver .....	38
Setting the Library Path Environment Variable (Salesforce Driver on UNIX/Linux)	38
Configuring a Data Source .....	39
Testing the Connection .....	40
Using the Performance Wizard .....	40
Starting the Wizard .....	41
Tuning Performance Using the Wizard .....	41
<b>2 Using The Product</b> .....	<b>43</b>
What Is ODBC? .....	43
How Does It Work? .....	44
Why Do Application Developers Need ODBC? .....	44
About the Product .....	44
Environment-Specific Information .....	45
For Windows Users .....	45
For UNIX and Linux Users .....	47
Using IP Addresses .....	54

Binding Parameter Markers . . . . .	55
Driver Threading Information . . . . .	56
Version String Information . . . . .	57
Retrieving Data Type Information . . . . .	59
Persisting a Result Set as an XML Data File . . . . .	60
Using the Windows XML Persistence Demo Tool . . . . .	61
Using the UNIX/Linux XML Persistence Demo Tool . . . . .	63
Translators . . . . .	63
<b>3 Advanced Features . . . . .</b>	<b>65</b>
Using Failover . . . . .	65
Connection Failover . . . . .	66
Extended Connection Failover . . . . .	67
Select Connection Failover . . . . .	69
Guidelines for Primary and Alternate Servers . . . . .	70
Using Client Load Balancing . . . . .	71
Using Connection Retry . . . . .	72
Summary of Failover-Related Options . . . . .	72
Using Client Information . . . . .	75
How Databases Store Client Information . . . . .	75
Storing Client Information . . . . .	76
Using Security . . . . .	77
Authentication . . . . .	77
Data Encryption Across the Network . . . . .	78
SSL Encryption . . . . .	79
Summary of Security-Related Options . . . . .	81
Using DataDirect Connection Pooling . . . . .	82
Creating a Connection Pool . . . . .	82
Adding Connections to a Pool . . . . .	83
Removing Connections from a Pool . . . . .	83
Handling Dead Connections in a Pool . . . . .	84
Connection Pool Statistics . . . . .	84
Summary of Pooling-Related Options . . . . .	85
Using DataDirect Bulk Load . . . . .	85
Bulk Export and Load Methods . . . . .	87
Exporting Data from a Database . . . . .	87
Bulk Loading to a Database . . . . .	89
The Bulk Load Configuration File . . . . .	91
Sample Applications . . . . .	94
Determining the Bulk Load Protocol . . . . .	94
Character Set Conversions . . . . .	94
External Overflow Files . . . . .	95
Using Bulk Load for Batch Inserts . . . . .	95
Using Bulk Load for Single Inserts/Updates/Deletes (Salesforce Driver) . . . . .	96
Summary of Bulk Load Related Options . . . . .	96

<b>4</b>	<b>Configuring the Product on UNIX/Linux</b>	<b>97</b>
	Environment Variables	97
	Library Search Path	97
	ODBCINI	98
	ODBCINST	98
	DD_INSTALLDIR	99
	The Test Loading Tool	99
	Data Source Configuration	100
	Configuration Through the Administrator	100
	Configuration Through the System Information (odbc.ini) File	103
	The demoodbc Application	119
	The example Application	119
	DSN-less Connections	119
	File Data Sources	124
	UTF-16 Applications on UNIX and Linux	125

## Part 2: The 32-Bit/64-Bit Drivers

<b>5</b>	<b>The DB2 Wire Protocol Driver</b>	<b>129</b>
	Driver Requirements	129
	Binding	130
	Creating DB2 Packages Using List Files	131
	Creating DB2 Packages Manually	131
	Configuring and Connecting to Data Sources	132
	Data Source Configuration in the UNIX/Linux odbc.ini File	132
	Data Source Configuration through a GUI	132
	Using a Connection String	148
	Using a Logon Dialog Box	149
	Connection Option Descriptions	150
	Performance Considerations	179
	IBM to IANA Code Page Values	181
	Data Types	181
	Using the XML Data Type	182
	Unicode Support	183
	Advanced Features	183
	Failover	183
	Client Information	183
	Security	185
	Connection Pooling	186
	DataDirect Bulk Load	186
	Cursor Stability Isolation Level	186
	XQuery Expressions	187
	Stored Procedure Support	187

Unexpected Characters . . . . .	187
Support for DB2 pureScale . . . . .	188
Persisting a Result Set as an XML Data File . . . . .	189
Isolation and Lock Levels Supported . . . . .	189
SQL Support . . . . .	189
ODBC Conformance Level . . . . .	189
Number of Connections and Statements Supported . . . . .	189
Using Arrays of Parameters . . . . .	190
<b>6 The Informix Wire Protocol Driver . . . . .</b>	<b>191</b>
Driver Requirements . . . . .	191
Configuring and Connecting to Data Sources . . . . .	191
Data Source Configuration in the UNIX/Linux odbc.ini File . . . . .	191
Data Source Configuration through a GUI . . . . .	192
Using a Connection String . . . . .	196
Using a Logon Dialog Box . . . . .	197
Connection Option Descriptions . . . . .	197
Performance Considerations . . . . .	204
Data Types . . . . .	205
Advanced Features . . . . .	206
MTS Support . . . . .	206
Persisting a Result Set as an XML Data File . . . . .	206
Isolation and Lock Levels Supported . . . . .	207
SQL Support . . . . .	207
ODBC Conformance Level . . . . .	207
Number of Connections and Statements Supported . . . . .	207
<b>7 The MySQL Wire Protocol Driver . . . . .</b>	<b>209</b>
Driver Requirements . . . . .	209
Configuring and Connecting to Data Sources . . . . .	209
Data Source Configuration in the UNIX/Linux odbc.ini File . . . . .	210
Data Source Configuration through a GUI . . . . .	210
Using a Connection String . . . . .	217
Using a Logon Dialog Box . . . . .	218
Connection Options Descriptions . . . . .	219
Performance Considerations . . . . .	235
Data Types . . . . .	236
Unicode Support . . . . .	237

Advanced Features . . . . .	238
Failover . . . . .	238
Security . . . . .	238
Connection Pooling . . . . .	238
Persisting a Result Set as an XML Data File . . . . .	238
Isolation and Lock Levels Supported . . . . .	238
SQL Support . . . . .	239
ODBC Conformance Level . . . . .	239
Number of Connections and Statements Supported . . . . .	239
<b>8 The Oracle Wire Protocol Driver . . . . .</b>	<b>241</b>
Driver Requirements . . . . .	241
Configuring and Connecting to Data Sources . . . . .	241
Data Source Configuration in the UNIX/Linux odbc.ini File . . . . .	242
Data Source Configuration through a GUI . . . . .	242
Using a Connection String . . . . .	261
Using a Logon Dialog Box . . . . .	262
Connection Option Descriptions . . . . .	264
Performance Considerations . . . . .	300
Data Types . . . . .	304
XMLType . . . . .	305
Unicode Support . . . . .	306
Advanced Features . . . . .	307
Failover . . . . .	307
Client Information . . . . .	307
Security . . . . .	307
Connection Pooling . . . . .	309
DataDirect Bulk Load . . . . .	309
MTS Support . . . . .	309
OS Authentication . . . . .	310
Support for Oracle RAC . . . . .	310
Support of Materialized Views . . . . .	310
Stored Procedure Results . . . . .	311
Unexpected Characters . . . . .	311
Persisting a Result Set as an XML Data File . . . . .	312
Isolation and Lock Levels Supported . . . . .	313
SQL Support . . . . .	313
ODBC Conformance Level . . . . .	313
Number of Connections and Statements Supported . . . . .	313
Using Parameter Arrays . . . . .	314

<b>9</b>	<b>The PostgreSQL Wire Protocol Driver</b>	<b>315</b>
	Driver Requirements	315
	Configuring and Connecting to Data Sources	315
	Data Source Configuration in the UNIX/Linux odbc.ini File	315
	Data Source Configuration through a GUI	316
	Using a Connection String	323
	Using a Logon Dialog Box	324
	Connection Option Descriptions	324
	Performance Considerations	344
	Data Types	345
	Using the XML Data Type	346
	Unicode Support	346
	Advanced Features	346
	Failover	347
	Security	347
	Connection Pooling	347
	Stored Procedure Results	347
	Persisting a Result Set as an XML Data File	349
	Isolation and Lock Levels Supported	349
	SQL Support	349
	ODBC Conformance Level	350
	Number of Connections and Statements Supported	350
	Using Arrays of Parameters	350
<b>10</b>	<b>The Progress OpenEdge® Wire Protocol Driver</b>	<b>351</b>
	Driver Requirements	351
	Configuring and Connecting to Data Sources	351
	Data Source Configuration in the UNIX odbc.ini File	351
	Data Source Configuration through a GUI	352
	Using a Connection String	357
	Using a Logon Dialog Box	358
	Connection Option Descriptions	358
	Performance Considerations	370
	Data Types	371
	Unicode Support	371
	Advanced Features	372
	Failover	372
	Security	372
	Isolation and Lock Levels Supported	372
	SQL Grammar Support	373



ODBC Conformance Level .....	373
Number of Connections and Statements Supported .....	373
<b>11 The SQL Server Wire Protocol Driver .....</b>	<b>375</b>
Driver Requirements .....	375
Configuring and Connecting to Data Sources .....	375
Data Source Configuration in the UNIX/Linux odbc.ini File .....	376
Data Source Configuration through a GUI .....	376
Using a Connection String .....	388
Using a Logon dialog Box .....	389
Connection Option Descriptions .....	390
Performance Considerations .....	413
Data Types .....	415
Unicode Support .....	416
Using the XML Data Type .....	417
Advanced Features .....	417
Failover .....	417
Security .....	417
Connection Pooling .....	418
DataDirect Bulk Load .....	418
Persisting a Result Set as an XML Data File .....	419
Isolation and Lock Levels Supported .....	419
Using The Snapshot Isolation Level .....	419
SQL Support .....	419
ODBC Conformance Level .....	420
Number of Connections and Statements Supported .....	420
Using Arrays of Parameters .....	420
<b>12 The Sybase Wire Protocol Driver .....</b>	<b>421</b>
Driver Requirements .....	421
Configuring and Connecting to Data Sources .....	421
Data Source Configuration in the UNIX/Linux odbc.ini File .....	421
Data Source Configuration through a GUI .....	422
Using a Connection String .....	436
Using a Logon Dialog Box .....	437
Connection Option Descriptions .....	438
Performance Considerations .....	465
Data Types .....	467
Unicode Support .....	468

Advanced Features . . . . .	469
Failover . . . . .	469
Security . . . . .	469
Connection Pooling . . . . .	470
DataDirect Bulk Load . . . . .	470
Unexpected Characters . . . . .	471
MTS Support . . . . .	473
NULL Values . . . . .	473
Persisting a Result Set as an XML Data File . . . . .	474
Isolation and Lock Levels Supported . . . . .	474
SQL Grammar Support . . . . .	474
ODBC Conformance Level . . . . .	474
Number of Connections and Statements Supported . . . . .	475
Using Arrays of Parameters . . . . .	475
<b>13 The Oracle Driver . . . . .</b>	<b>477</b>
Driver Requirements . . . . .	477
Windows . . . . .	478
UNIX and Linux . . . . .	478
Configuring and Connecting to Data Sources . . . . .	479
Data Source Configuration in the UNIX/Linux odbc.ini File . . . . .	479
Data Source Configuration through a GUI . . . . .	479
Using a Connection String . . . . .	487
Using a Logon Dialog Box . . . . .	488
Connection Option Descriptions . . . . .	489
Performance Considerations . . . . .	504
Data Types . . . . .	506
XMLType . . . . .	507
Unicode Support . . . . .	509
Advanced Features . . . . .	510
Unexpected Characters . . . . .	510
MTS Support . . . . .	511
OS Authentication . . . . .	511
Support for Oracle RAC . . . . .	511
Support of Materialized Views . . . . .	512
Stored Procedure Results . . . . .	512
Persisting a Result Set as an XML Data File . . . . .	513
Isolation and Lock Levels Supported . . . . .	513
SQL Support . . . . .	513
ODBC Conformance Level . . . . .	513

Number of Connections and Statements Supported . . . . .	514
Using Arrays of Parameters. . . . .	514
<b>14 The SQL Server Legacy Wire Protocol Driver . . . . .</b>	<b>515</b>
Driver Requirements . . . . .	515
Windows . . . . .	515
UNIX and Linux . . . . .	515
Configuring and Connecting to Data Sources . . . . .	516
Data Source Configuration in the UNIX/Linux odbc.ini File. . . . .	516
Data Source Configuration through a GUI . . . . .	516
Using a Connection String. . . . .	520
Using a Login Dialog Box . . . . .	521
Connection Option Descriptions . . . . .	522
Windows . . . . .	523
UNIX and Linux . . . . .	523
Performance Considerations . . . . .	535
Data Types. . . . .	536
Unicode Support . . . . .	537
Advanced Features . . . . .	538
Isolation and Lock Levels Supported. . . . .	538
Using The Snapshot Isolation Level . . . . .	538
SQL Support . . . . .	538
ODBC Conformance Level . . . . .	539
Number of Connections and Statements Supported . . . . .	539
Using Arrays of Parameters. . . . .	539
<b>Part 3: The 32-Bit Drivers</b>	
<b>15 The Btrieve (Pervasive.SQL) Driver. . . . .</b>	<b>543</b>
Driver Requirements . . . . .	543
Managing Databases. . . . .	544
Transactions . . . . .	544
Configuring and Connecting to Data Sources . . . . .	544
Using a Connection String. . . . .	548
Connection Option Descriptions . . . . .	549
Defining Table Structure. . . . .	553
Data Types. . . . .	555
Indexes . . . . .	556
Column Names . . . . .	557
Select Statement . . . . .	557
Alter Table Statement. . . . .	558

Create and Drop Index Statements . . . . .	558
Create Index . . . . .	558
Drop Index. . . . .	559
Isolation and Lock Levels Supported. . . . .	559
SQL Support . . . . .	559
ODBC Conformance Level . . . . .	559
Number of Connections and Statements Supported . . . . .	559
<b>16 The dBASE Driver. . . . .</b>	<b>561</b>
Driver Requirements . . . . .	561
Configuring and Connecting to Data Sources . . . . .	561
Data Source Configuration in the UNIX/Linux odbc.ini File. . . . .	562
Data Source Configuration through a GUI . . . . .	562
dBASE. . . . .	563
FoxPro 3.0 DBC . . . . .	566
Using a Connection String. . . . .	568
Connection Option Descriptions . . . . .	569
Defining Index Attributes on Windows. . . . .	576
Defining Index Attributes on UNIX and Linux . . . . .	578
Data Types. . . . .	578
Column Names . . . . .	579
Select Statement . . . . .	580
Alter Table Statement. . . . .	580
Create and Drop Index Statements . . . . .	581
Create Index . . . . .	581
Drop Index. . . . .	582
Pack Statement . . . . .	583
SQL Statements for FoxPro 3.0 Database Containers . . . . .	583
Locking . . . . .	584
Levels of Database Locking . . . . .	584
Limit on Number of Locks . . . . .	584
How Transactions Affect Record Locks . . . . .	584
Isolation and Lock Levels Supported. . . . .	585
SQL Support . . . . .	585
ODBC Conformance Level . . . . .	585
Number of Connections and Statements Supported . . . . .	585

<b>17 The Informix Driver</b>	<b>587</b>
Driver Requirements	587
Windows	587
UNIX (AIX, HP-UX PA-RISC, and Solaris)	587
Configuring and Connecting to Data Sources	588
Data Source Configuration in the UNIX/Linux odbc.ini File	588
Data Source Configuration through a GUI	589
Using a Connection String	593
Using a Logon Dialog Box	594
Connection Option Descriptions	595
Performance Considerations	602
Data Types	602
MTS Support	604
Persisting a Result Set as an XML Data File	605
Isolation and Lock Levels Supported	605
SQL Support	605
ODBC Conformance Level	606
Number of Connections and Statements Supported	606
<b>18 The Text Driver</b>	<b>607</b>
Driver Requirements	607
Formats for Text Files	607
Configuring Data Sources	608
Data Source Configuration in the UNIX/Linux odbc.ini File	608
Data Source Configuration through a GUI	609
Using a Connection String	612
Connection Option Descriptions	613
Defining Table Structure on Windows	621
Defining Table Structure on UNIX and Linux	624
Date Masks	625
Data Types	626
Select Statement	627
Alter Table Statement	627
SQL Support	627
ODBC Conformance Level	628
Number of Connections and Statements Supported	628

<b>19 The XML Driver</b>	<b>629</b>
Driver Requirements	629
Supported Tabular Formats for XML Documents	630
Hierarchical-Formatted XML Document Support	630
Defining Locations	633
Specifying Table Names in SQL Statements	633
Configuring and Connecting to Data Sources	635
Using a Connection String	641
Using a Logon Dialog Box	641
Connection Option Descriptions	642
Driver Setup Dialog Box Descriptions	642
Configure Location Dialog Box Descriptions	647
Using Hints for Tabular-Formatted XML Documents	653
Data Types	655
Unicode Support	657
Persisting a Result Set as an XML Data File	657
ODBC Conformance Level	657
Number of Connections and Statements Supported	658
SQL Support	658
SQL Statements	658
Extensions to SQL Standards	658
Grammar Token Definitions	659
 <b>Part 4: The Connect XE Drivers</b>	
<b>20 The Greenplum Wire Protocol Driver</b>	<b>669</b>
Driver Requirements	669
Configuring and Connecting to Data Sources	669
Data Source Configuration in the UNIX/Linux odbc.ini File	669
Data Source Configuration through a GUI	670
Using a Connection String	676
Using a Logon Dialog Box	677
Connection Option Descriptions	677
Performance Considerations	693
Data Types	694
Unicode Support	695
Advanced Features	695
Failover	695
Connection Pooling	695
Stored Procedure Results	696
Isolation and Lock Levels Supported	697
SQL Support	698

ODBC Conformance Level .....	698
Number of Connections and Statements Supported .....	698
Using Arrays of Parameters.....	698
<b>21 The Salesforce Driver .....</b>	<b>699</b>
Driver Requirements .....	701
Configuring and Connecting to Data Sources .....	701
Data Source Configuration in the UNIX/Linux odbc.ini File.....	701
Data Source Configuration through a GUI .....	702
Using a Connection String.....	716
Using a Logon Dialog Box.....	717
Connection Option Descriptions .....	718
Performance Considerations .....	741
Data Types.....	742
Mapping Objects to Tables.....	745
Client-Side Caches .....	745
Creating a Cache .....	746
Modifying a Cache Definition.....	746
Dropping a Cache .....	748
Cache MetaData .....	748
Catalog Tables .....	748
SYSTEM_CACHES Catalog Table .....	748
SYSTEM_CACHE_REFERENCES Catalog Table .....	750
SYSTEM_REMOTE_SESSIONS Catalog Table.....	750
SYSTEM_SESSIONINFO Catalog Table .....	752
SYSTEM_SESSIONS Catalog Table .....	752
Timeouts .....	753
Views and Remote/Local Tables .....	754
Using Identifiers .....	754
Database Configuration File .....	755
Mapping Objects to Tables.....	757
Reports .....	758
Connecting Through a Proxy Server .....	758
Configuring the SQL Engine Server .....	759
Configuring Server Mode .....	759
Starting the SQL Engine Server .....	761
Stopping the SQL Engine Server .....	762
Configuring Java Logging for the SQL Engine Server .....	762
Unicode Support .....	763
Advanced Features .....	763
Failover .....	763
Security .....	763
Connection Pooling .....	764

Parameter Metadata Support . . . . .	764
Insert and Update Statements . . . . .	764
Select Statements . . . . .	764
Using DataDirect Bulk Load With the Salesforce Driver . . . . .	765
Bulk Operation Support . . . . .	765
Using Column Mapping . . . . .	766
Using Bulk Load for Single Inserts/Updates/Deletes . . . . .	766
Error Handling . . . . .	767
Driver Errors . . . . .	767
Data Source Errors . . . . .	767
Isolation and Lock Levels Supported . . . . .	767
SQL Support . . . . .	768
ODBC Conformance Level . . . . .	768
Number of Connections and Statements Supported . . . . .	768
<b>22 The Sybase IQ Wire Protocol Driver . . . . .</b>	<b>769</b>
Driver Requirements . . . . .	769
Configuring and Connecting to Data Sources . . . . .	769
Data Source Configuration in the UNIX/Linux odbc.ini File . . . . .	769
Data Source Configuration through a GUI . . . . .	770
Using a Connection String . . . . .	778
Using a Logon Dialog Box . . . . .	779
Connection Option Descriptions . . . . .	780
Performance Considerations . . . . .	797
Data Types . . . . .	799
Unicode Support . . . . .	800
Advanced Features . . . . .	801
Failover . . . . .	801
Connection Pooling . . . . .	801
Unexpected Characters . . . . .	801
NULL Values . . . . .	802
Persisting a Result Set as an XML Data File . . . . .	803
Isolation and Lock Levels Supported . . . . .	803
SQL Support . . . . .	803
ODBC Conformance Level . . . . .	804
Number of Connections and Statements Supported . . . . .	804
Using Arrays of Parameters . . . . .	804



<b>23 The Driver for Apache Hive™</b>	<b>805</b>
Driver Requirements	805
Configuring and Connecting to Data Sources	805
Data Source Configuration in the UNIX odbc.ini File	806
Data Source Configuration through a GUI	806
Using a Connection String	810
Using a Logon Dialog Box	811
Connection Option Descriptions	811
Performance Considerations	817
Data Types	817
Materialized Views	818
Stored Procedures	818
Unicode Support	818
Isolation and Lock Levels Supported	819
SQL Support	819
ODBC Conformance Level	819
Using Arrays of Parameters	819
Limitations on Apache Hive™ Functionality	820
Best Practices	820
<b>24 The Driver for the Teradata Database</b>	<b>821</b>
Driver Requirements	821
Configuring and Connecting to Data Sources	821
Data Source Configuration in the UNIX/Linux odbc.ini File	822
Data Source Configuration through a GUI	822
Using a Connection String	827
Using a Logon Dialog Box	828
Connection Option Descriptions	829
Data Types	842
Unicode Support	843
Persisting a Result Set as an XML Data File	844
Isolation and Lock Levels Supported	844
SQL Support	844
ODBC Conformance Level	844
Number of Connections and Statements Supported	844
<b>Index</b>	<b>845</b>



# List of Tables

Table 2-1.	IP Address Formats Supported by 32- and 64-bit DataDirect Connect for ODBC Drivers. . . . .	54
Table 2-2.	IP Address Formats Supported by 32- and 64-bit DataDirect Connect XE for ODBC Drivers . . . . .	54
Table 2-3.	Threading Information for DataDirect Connect for ODBC . . . . .	56
Table 2-4.	Threading Information for DataDirect Connect XE for ODBC . . . . .	56
Table 3-1.	Summary: Failover and Related Connection Options . . . . .	72
Table 3-2.	Database Locations for Storing Client Information . . . . .	76
Table 3-3.	Driver Support for NTLM Authentication . . . . .	78
Table 3-4.	Summary: Security Connection Options . . . . .	81
Table 3-5.	Pool Reset Connection Attributes . . . . .	83
Table 3-6.	Pool Statistics Connection Attribute . . . . .	84
Table 3-7.	Summary: Connection Pooling Connection Options . . . . .	85
Table 3-8.	Bulk Load Behavior for DataDirect Connect for ODBC . . . . .	85
Table 3-9.	Bulk Load Behavior for DataDirect Connect XE for ODBC . . . . .	86
Table 3-10.	Summary: Bulk Load Connection Options . . . . .	96
Table 5-1.	DB2 Wire Protocol Attribute Names . . . . .	151
Table 5-2.	DB2 Data Types . . . . .	181
Table 5-3.	WLM Attributes for DB2 V9.5 and V9.7 for Linux/UNIX/Windows . . . . .	184
Table 5-4.	WLM Attributes for DB2 z/OS. . . . .	185
Table 5-5.	Kerberos Authentication Requirements for the DB2 Wire Protocol Driver . . . . .	185
Table 5-6.	Cursor Stability Examples . . . . .	186
Table 6-1.	Informix Wire Protocol Attribute Names . . . . .	198
Table 6-2.	Informix Data Types . . . . .	205
Table 7-1.	MySQL Wire Protocol Attribute Names . . . . .	219
Table 7-2.	MySQL Data Types . . . . .	236
Table 8-1.	Oracle Wire Protocol Attribute Names . . . . .	264
Table 8-2.	Oracle Data Types . . . . .	304
Table 8-3.	Kerberos Authentication Requirements for the Oracle Wire Protocol Driver . . . . .	308
Table 9-1.	PostgreSQL Wire Protocol Attribute Names . . . . .	325
Table 9-2.	PostgreSQL Data Types. . . . .	345
Table 10-1.	Progress OpenEdge Wire Protocol Attribute Names . . . . .	359
Table 10-2.	Progress OpenEdge Data Types . . . . .	371

Table 11-1.	SQL Server Wire Protocol Attribute Names .....	390
Table 11-2.	Microsoft SQL Server Data Types .....	415
Table 11-3.	Mapping Microsoft SQL Server and Windows Azure SQL Database Data Types to Unicode Data Types .....	416
Table 11-4.	Kerberos Authentication Requirements for the SQL Server Wire Protocol Driver .....	418
Table 12-1.	Sybase Wire Protocol Attribute Names .....	438
Table 12-2.	Sybase Data Type Mapping .....	467
Table 12-3.	Kerberos Authentication Requirements for the Sybase Wire Protocol Driver .....	470
Table 12-4.	Summary of Fast and Slow Bulk Copy Mode Characteristics .....	471
Table 13-1.	Oracle Attribute Names .....	489
Table 13-2.	Oracle Data Types .....	506
Table 14-1.	SQL Server Legacy Wire Protocol Attribute Names on Windows .....	523
Table 14-2.	SQL Server Legacy Wire Protocol Attribute Names on UNIX/Linux .....	524
Table 14-3.	Microsoft SQL Server Data Types .....	536
Table 15-1.	Btrieve Attribute Names .....	549
Table 15-2.	Btrieve Data Types .....	555
Table 16-1.	dBASE Attribute Names .....	569
Table 16-2.	dBASE Data Types .....	579
Table 16-3.	Additional FoxPro 3.0 Data Types .....	579
Table 16-4.	dBASE-Compatible Index Summary .....	582
Table 17-1.	Informix Attribute Names .....	595
Table 17-2.	Informix Data Types .....	602
Table 18-1.	Common Text File Formats .....	607
Table 18-2.	Text Attribute Names .....	613
Table 18-3.	Date Masks for Text Driver .....	625
Table 18-4.	Date Mask Examples .....	626
Table 18-5.	Text Data Types .....	626
Table 19-1.	Common Tabular Formats for XML Documents .....	630
Table 19-2.	XML Attribute Names .....	642
Table 19-3.	XML Configure Location Attribute Names .....	647
Table 19-4.	Data Islands Data Types .....	655
Table 19-5.	ADO 2.5 Persisted Files Data Types .....	655
Table 19-6.	DataDirect Format Data Types .....	656
Table 19-7.	SQL Extensions .....	658
Table 19-8.	Time and Date Literals .....	661

Table 19-9.	SQL Operators and Symbols . . . . .	661
Table 19-10.	Reserved Keywords . . . . .	662
Table 20-1.	Greenplum Wire Protocol Attribute Names . . . . .	678
Table 20-2.	Greenplum Data Types . . . . .	694
Table 21-1.	Salesforce Attribute Names . . . . .	718
Table 21-2.	Salesforce Data Types for Local Tables . . . . .	743
Table 21-3.	Salesforce Data Types for Remote Tables . . . . .	744
Table 21-4.	SYSTEM_CACHES Catalog Table . . . . .	749
Table 21-5.	SYSTEM_CACHE_REFERENCES . . . . .	750
Table 21-6.	SYSTEM_REMOTE_SESSIONS Catalog Table . . . . .	750
Table 21-7.	SYSTEM_SESSIONINFO Catalog Table . . . . .	752
Table 21-8.	SYSTEM_SESSIONS . . . . .	752
Table 21-9.	Extended Functionality for the SQLColumns Function. . . . .	768
Table 22-1.	Sybase IQ Wire Protocol Attribute Names . . . . .	780
Table 22-2.	Sybase IQ Data Type Mapping . . . . .	799
Table 22-3.	Mapping Sybase IQ Data Types to Unicode Data Types . . . . .	800
Table 23-1.	Supported Apache Hive Versions and Distribution Versions . . . . .	805
Table 23-2.	Attribute Names for the Driver for Apache Hive . . . . .	812
Table 23-3.	Apache Hive Data Types . . . . .	817
Table 24-1.	Teradata Attribute Names . . . . .	830
Table 24-2.	Teradata Data Types . . . . .	842
Table 24-3.	Teradata Unicode Data Types . . . . .	843



# Preface

This book is your user's guide to Progress® DataDirect Connect® Series *for* ODBC, which includes the following products:

- DataDirect Connect *for* ODBC
- DataDirect Connect64 *for* ODBC
- DataDirect Connect XE (Extended Edition) *for* ODBC
- DataDirect Connect64 XE *for* ODBC

These products consist of a number of database *drivers* that are compliant with the Open Database Connectivity (ODBC) specification. A number of these drivers are available in both 32- and 64-bit formats. See the ["Product Matrix" on page 24](#) for details.

---

## Product Platforms

DataDirect Connect Series *for* ODBC drivers enable you to connect to a variety of databases from these platforms:

### Windows (32-bit)

- Windows 7
- Windows Server 2008
- Windows Vista
- Windows XP
- Windows Server 2003
- Windows 2000

### Windows (64-bit)

- Windows 7
- Windows Server 2008
- Windows Vista
- Windows XP Professional
- Windows Server 2003

### UNIX and Linux (32-bit)

- AIX
- HP-UX aCC Enabled
- Linux
- Oracle Solaris

### UNIX and Linux (64-bit)

- AIX
- HP-UX aCC Enabled
- Linux
- Oracle Solaris

See ["Environment-Specific Information" on page 45](#) for detailed information regarding these platforms.

---

## Product Matrix

The DataDirect Connect Series *for* ODBC products include 32- and 64-bit drivers. DataDirect Connect *for* ODBC (32-bit) and DataDirect Connect64 *for* ODBC (64-bit) are detailed in the following table.

Driver	Connect for ODBC	Connect64 for ODBC
DB2 Wire Protocol	X	X
Informix Wire Protocol	X	X
MySQL Wire Protocol	X	X
Oracle Wire Protocol	X	X
PostgreSQL Wire Protocol	X	X
Progress OpenEdge Wire Protocol	X	X
SQL Server Wire Protocol	X	X
Sybase Wire Protocol	X	X
Oracle (client)	X	X
SQL Server Legacy Wire Protocol	X	X
Btrieve	X	
dBASE	X	
Informix (client)	X	
Text	X	
XML	X	

DataDirect Connect XE *for* ODBC consists of the drivers detailed in the following table.

Driver	Connect XE for ODBC	Connect64 XE for ODBC
Greenplum Wire Protocol	X	X
Salesforce	X	X
Sybase IQ	X	X
Driver for Apache Hive	X	X
Driver for the Teradata Database	X	X



# What's New in This Release?

The highlights of Release 7.1.0 are:

## ■ New database version support

- DB2 UDB V10.1 for Linux, UNIX, and Windows
- DB2 pureScale
- Greenplum 4.2
- Microsoft SQL Server 2012
- Microsoft Windows Azure for SQL Database
- Salesforce API Version 25
- Teradata 14.0

## ■ New driver: DataDirect Connect XE and DataDirect Connect64 XE for ODBC for Apache Hive Wire Protocol

- Supported with the following Apache Hive versions:
  - Amazon Elastic MapReduce (Amazon EMR)
  - Apache Hadoop Hive
  - Cloudera's Distribution Including Apache Hadoop (CDH)
  - MapR Distribution for Apache Hadoop
- Returns result set metadata for parameterized statements that have been prepared but not yet executed.
- Supports parameter arrays, processing the arrays as a series of executions, one execution for each row in the array.
- Provides a connection option that allows you to configure the driver to report that it supports transactions, although Hive does not support transactions. This provides a workaround for applications that do not operate with a driver that reports that transactions are not supported.
- Supports the following standard SQL functionality:
  - Create Index, Create Table, and Create View
  - Insert, Update, and Delete
  - Drop Index, Drop Table, and Drop View

## ■ Support for setting the value of undocumented connection options using the setup dialog for the following drivers on Windows:

- |                                   |                           |
|-----------------------------------|---------------------------|
| ■ DB2 Wire Protocol               | ■ Driver for Apache Hive  |
| ■ MySQL Wire Protocol             | ■ Greenplum Wire Protocol |
| ■ Oracle Wire Protocol            | ■ Salesforce              |
| ■ PostgreSQL Wire Protocol        | ■ Sybase IQ Wire Protocol |
| ■ Progress OpenEdge Wire Protocol |                           |
| ■ SQL Server Wire Protocol        |                           |
| ■ Sybase Wire Protocol            |                           |
| ■ Oracle                          |                           |

**■ DB2 Wire Protocol driver enhancements**

- Certified with DB2 V10.1 for Linux, UNIX, and Windows

**■ Oracle Wire Protocol driver enhancements**

- Oracle Advanced Security support, which can be configured using the following connection options:
  - Data Integrity Level sets the level of OAS data integrity used for data sent between the driver and database server.
  - Data Integrity Types specifies one or multiple algorithms to protect against attacks that intercept and modify data being transmitted between the client and server when OAS data integrity is enabled using the Data Integrity Level option.
  - Encryption Level determines whether data is encrypted and decrypted when transmitted over the network between the driver and database server using OAS encryption.
  - Encryption Types specifies one or multiple algorithms to use if OAS encryption is enabled using the Encryption Level property.

**■ PostgreSQL driver enhancements**

- The Max Char Size connection option specifies the maximum size of columns of type SQL\_CHAR that the driver describes through result set descriptions and catalog functions.
- The Max Long Varchar Size connection option specifies the maximum size of columns of type SQL\_LONGVARCHAR that the driver describes through result set descriptions and catalog functions.
- The Enable Keyset Cursors connection option enables emulated Keyset cursors to provide scrollable cursors to an ODBC application.
- The Keyset Cursor Options connection option determines which columns are used to comprise the keyset that the driver uses to create the initial keyset on which cursor operations are based.

**■ Greenplum driver enhancements**

- The Max Char Size connection option specifies the maximum size of columns of type SQL\_VARCHAR that the driver describes through result set descriptions and catalog functions.
- The Max Long Varchar Size connection option specifies the maximum size of columns of type SQL\_LONGVARCHAR that the driver describes through result set descriptions and catalog functions.
- The Enable Keyset Cursors connection option enables emulated Keyset cursors to provide scrollable cursors to an ODBC application.
- The Keyset Cursor Options connection option determines which columns are used to comprise the keyset that the driver uses to create the initial keyset on which cursor operations are based.

### ■ Salesforce driver enhancements

- The Refresh Schema connection option specifies whether the driver automatically refreshes the remote object mapping and other information contained in a remote schema the first time a user connects to the specified embedded database.
- The KeywordConflictSuffix *key=value* pair in the Config Options connection option can specify a string of up to five alphanumeric characters that the driver appends to any object or field name that conflicts with a SQL engine keyword.

NOTE: For the Salesforce Web Service API versions supported by the Salesforce driver, refer to the product matrix on the Progress DataDirect Web site:

<http://www.datadirect.com/products/odbc/matrix/connectodbc.htm>

---

## Using this Book

The content of this book assumes that you are familiar with your operating system and its commands. It contains the following information:

- [Chapter 1 “Quick Start Connect” on page 35](#) explains the basics for quickly configuring and testing the drivers.
- [Chapter 2 “Using The Product” on page 43](#) explains the drivers and ODBC, and discusses environment-specific subjects.
- [Chapter 3 “Advanced Features” on page 65](#) explains at a general level advanced driver features such as failover, security, connection pooling, and bulk load.
- [Chapter 4 “Configuring the Product on UNIX/Linux” on page 97](#) discusses UNIX and Linux environment variables and configuration of the drivers. It also provides a sample system information file, as well as discussing other driver tools for UNIX and Linux.
- A chapter for each database driver. Each driver’s chapter is structured in the same way. First, it lists which versions of the databases the driver supports, the operating environments in which the driver runs, and the driver requirements for your operating environment. Next, it explains how to configure a data source and how to connect to that data source. Finally, the chapter provides information about data types, ODBC conformance levels, isolation and lock levels supported, and other driver-specific information.

If you are writing programs to access ODBC drivers, you need to obtain a copy of the *ODBC Programmer’s Reference* for the Microsoft Open Database Connectivity Software Development Kit, available from Microsoft Corporation.

Database drivers are continually being added to each operating environment. For the latest information about the specific drivers available for your platform, refer to the Progress DataDirect database support matrix Web pages at:

<http://www.datadirect.com/products/odbc/matrix/connectodbc.htm> (32-bit)

<http://www.datadirect.com/products/odbc64/matrix/connect64odbc.htm> (64-bit)

NOTE: This book refers the reader to Web pages using URLs for more information about specific topics, including Web URLs not maintained by Progress DataDirect. Because it is the nature of Web content to change frequently, Progress DataDirect can guarantee only that the URLs referenced in this book were correct at the time of publication.

---

# Conventions Used in This Book

The following sections describe the typography and other conventions used in this book.

## Typographical Conventions

This book uses the following typographical conventions:

Convention	Explanation
<i>italics</i>	Introduces new terms with which you may not be familiar, and is used occasionally for emphasis.
<b>bold</b>	Emphasizes important information. Also indicates button, menu, and icon names on which you can act. For example, click <b>Next</b> .
UPPERCASE	Indicates keys or key combinations that you can use. For example, press the ENTER key.  Also used for SQL reserved words.
monospace	Indicates syntax examples, values that you specify, or results that you receive.
<i>monospaced italics</i>	Indicates names that are placeholders for values that you specify. For example, <i>filename</i> .
forward slash /	Separates menus and their associated commands. For example, Select File / Copy means that you should select Copy from the File menu.  The slash also separates directory levels when specifying locations under UNIX.
vertical rule	Indicates an "OR" separator used to delineate items.
brackets [ ]	Indicates optional items. For example, in the following statement: SELECT [DISTINCT], DISTINCT is an optional keyword.  Also indicates sections of the Windows Registry.
braces { }	Indicates that you must select one item. For example, {yes   no} means that you must specify either yes or no.
ellipsis . . .	Indicates that the immediately preceding item can be repeated any number of times in succession. An ellipsis following a closing bracket indicates that all information in that unit can be repeated.

## Environment-Specific Information

The drivers are supported in the Windows, UNIX, and Linux environments. When the information provided is not applicable to all supported environments, the following symbols are used to identify that information:



The Windows symbol signifies text that is applicable only to Windows.



The UNIX symbol signifies text that is applicable only to UNIX and Linux.

---

## About the Product Documentation

The product library consists of the following books:

- *DataDirect Connect Series for ODBC Installation Guide* details requirements and procedures for installing the product.
- *DataDirect Connect Series for ODBC User's Guide* provides information about configuring and using the product.
- *DataDirect Connect Series for ODBC Reference* provides detailed reference information about the product.
- *DataDirect Connect Series for ODBC Troubleshooting Guide* provides information about error messages and troubleshooting procedures for the product.

### HTML Version

This library, except for the installation guide, is placed on your system as HTML-based online help during a normal installation of the product. It is located in the help subdirectory of the product installation directory. To use online help, you must have one of the following Internet browsers installed.

- Internet Explorer 5.x, 6.x, 7.x, 8.x, and 9.x
- Mozilla Firefox 1.x, 2.x, 3.x, 8.0, and 10.0
- Netscape 4.x, 7.x 8.x, and 9.0
- Safari 1.x, 2.x, 3.x, and 5.1.7
- Opera 7.54u2, 8.x, 9.x, and 12.0



On Windows, you can access the entire Help system by selecting the help icon that appears in the DataDirect program group.

On all platforms, you can access the entire Help system by opening the following file from within your browser:

```
install_dir/help/help.htm
```

where *install\_dir* is the path to the product installation directory.

Or, from a command-line environment, at a command prompt, enter:

```
browser_exe install_dir/help/help.htm
```

where *browser\_exe* is the name of your browser executable and *install\_dir* is the path to the product installation directory.

After the browser opens, the left pane displays the Table of Contents, Index, and Search tabs for the entire documentation library. When you have opened the main screen of the Help system in your browser, you can bookmark it in the browser for quick access later.

NOTE: Security features set in your browser can prevent the Help system from launching. A security warning message is displayed. Often, the warning message provides instructions for unblocking the Help system for the current session. To allow the Help system to launch without encountering a security warning message, the security settings in your browser can be modified. Check with your system administrator before disabling any security features.

Help is also available from the setup dialog box for each driver. When you click **Help**, your browser opens to the correct topic without opening the help Table of Contents. A grey toolbar appears at the top of the browser window.



This tool bar contains previous and next navigation buttons. If, after viewing the help topic, you want to see the entire library, click:



on the left side of the toolbar, which opens the left pane and displays the Table of Contents, Index, and Search tabs.

## PDF Version

The product documentation is also provided in PDF format. You can view or print the documentation, and perform text searches in the files. The PDF documentation is available on the Progress DataDirect Web site at:

<http://www.datadirect.com/support/product-info/documentation/by-product.html>

You can download the entire library in a compressed file. When you uncompress the file, it appears in the correct directory structure.

Maintaining the correct directory structure allows cross-book text searches and cross-references. If you download or copy the books individually outside of their normal directory structure, their cross-book search indexes and hyperlinked cross-references to other volumes will not work. You can view a book individually, but it will not automatically open other books to which it has cross-references.

To help you navigate through the library, a file, called *books.pdf*, is provided. This file lists each online book provided for the product. We recommend that you open this file first and, from this file, open the book you want to view.

NOTE: To use the cross-book search feature, you must use Adobe Reader 8.0 or higher. If you are using a version of Adobe Reader that does not support the cross book search feature or are using a version of Adobe Reader earlier than 8.0, you can still view the books and use the Find feature within a single book.

---

## Contacting Customer Support

Progress DataDirect offers a variety of options to meet your customer support needs. Please visit our Web site for more details and for contact information:

<http://www.datadirect.com/support/index.html>

The Progress DataDirect Web site provides the latest support information through our global service network. The SupportLink program provides access to support contact details, tools, patches, and valuable information, including a list of FAQs for each product. In addition, you can search our Knowledgebase for technical bulletins and other information.

When you contact us for assistance, please provide the following information:

- Your customer number or the serial number that corresponds to the product for which you are seeking support, or a case number if you have been provided one for your issue. If you do not have a SupportLink contract, the SupportLink representative assisting you will connect you with our Sales team.
- Your name, phone number, email address, and organization. For a first-time call, you may be asked for full customer information, including location.
- The Progress DataDirect product and the version that you are using.
- The type and version of the operating system where you have installed your product.
- Any database, database version, third-party software, or other environment information required to understand the problem.
- A brief description of the problem, including, but not limited to, any error messages you have received, what steps you followed prior to the initial occurrence of the problem, any trace logs capturing the issue, and so on. Depending on the complexity of the problem, you may be asked to submit an example or reproducible application so that the issue can be re-created.
- A description of what you have attempted to resolve the issue. If you have researched your issue on Web search engines, our Knowledgebase, or have tested additional configurations, applications, or other vendor products, you will want to carefully note everything you have already attempted.
- A simple assessment of how the severity of the issue is impacting your organization.





# Part 1: Getting Started

This part contains the following chapters:

- [Chapter 1 “Quick Start Connect” on page 35](#)
- [Chapter 2 “Using The Product” on page 43](#)
- [Chapter 3 “Advanced Features” on page 65](#)
- [Chapter 4 “Configuring the Product on UNIX/Linux” on page 97](#)



# 1 Quick Start Connect

This chapter provides basic information about configuring your drivers immediately after installation and testing your connection. To take full advantage of the features of the drivers, read ["About the Product" on page 44](#) and the driver-specific chapters.

Information that the driver needs to connect to a database is stored in a *data source*. The ODBC specification describes three types of data sources: user data sources, system data sources (not a valid type on UNIX/Linux), and file data sources. On Windows, user and system data sources are stored in the registry of the local computer. The difference is that only a specific user can access user data sources, whereas any user of the machine can access system data sources. On Windows, UNIX, and Linux, file data sources, which are simply text files, can be stored locally or on a network computer, and are accessible to other machines.

When you define and configure a data source, you store default connection values for the driver that are used each time you connect to a particular database. You can change these defaults by modifying the data source.

This chapter contains the following sections:

- ["Configuring and Connecting on Windows" on page 35](#)
- ["Configuring and Connecting on UNIX and Linux" on page 37](#)
- ["Using the Performance Wizard" on page 40](#)

---

## Configuring and Connecting on Windows

The following basic information enables you to configure a data source and test connect with a driver immediately after installation. On Windows, you can configure and modify data sources through the ODBC Administrator using a driver Setup dialog box. Default connection values are specified through the options on the tabs of the Setup dialog box and are stored either as a user or system data source in the Windows Registry, or as a file data source in a specified location.

### Setting the Library Path Environment Variable (Salesforce Driver on Windows)

Before you can use the Salesforce driver, you must set the PATH environment variable to the path of the jvm.dll file of your Java Virtual Machine (JVM).

## Configuring a Data Source

- 1 From the DataDirect program group, start the ODBC Administrator and click either the **User DSN**, **System DSN**, or **File DSN** tab to display a list of data sources.

- **User DSN:** If you installed default DataDirect ODBC user data sources as part of the installation, select the appropriate data source name and click **Configure** to display the driver Setup dialog box.

If you are configuring a new user data source, click **Add** to display a list of installed drivers. Select the appropriate driver and click **Finish** to display the driver Setup dialog box.

- **System DSN:** To configure a new system data source, click **Add** to display a list of installed drivers. Select the appropriate driver and click **Finish** to display the driver Setup dialog box.
- **File DSN:** To configure a new file data source, click **Add** to display a list of installed drivers. Select the driver and click **Advanced** to specify attributes; otherwise, click **Next** to proceed. Specify a name for the data source and click **Next**. Verify the data source information; then, click **Finish** to display the driver Setup dialog box.

The General tab of the Setup dialog box appears by default.

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 in this book.

- 2 The following two options appear on the General tab of all driver Setup dialog boxes:

**Data Source Name:** Type a string that identifies this data source configuration, such as Accounting.

**Description:** Type an optional long description of a data source name, such as My Accounting Database.

Provide the requested information for all other options on the General tab; then, click **Apply** to configure the data source.

## Testing the Connection

- 1 After you have configured the data source, you can click **Test Connect** on the Setup dialog box to attempt to connect to the data source using the connection options specified in the dialog box. Some drivers immediately return a message indicating success or failure. For most drivers, a logon dialog box appears as described in each individual driver chapter.
- 2 Supply the requested information in the logon dialog box and click **OK**. 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 incorrect environment or connection value, it displays an appropriate error message. Click **OK**.

- 3 On the driver Setup dialog box, click **OK**. The values you have specified are saved and are the defaults used when you connect to the data source. You can change these defaults by using the previously described procedure to modify your data source. You can override these defaults by connecting to the data source using a connection string with alternate values. See individual driver chapters for information about using connection strings.

---

## Configuring and Connecting on UNIX and Linux

The following basic information enables you to configure a data source and test connect with a driver immediately after installation. See [Chapter 4 “Configuring the Product on UNIX/Linux”](#) for detailed information about configuring the UNIX/Linux environment and data sources.

NOTE: In the following examples, xx in a driver filename represents the driver level number.

### Environment Configuration

- 1 Check your permissions: You must log in as a user with full r/w/x permissions recursively on the entire product installation directory.
- 2 From your login shell, determine which shell you are running by executing:
 

```
echo $SHELL
```
- 3 Run one of the following product setup scripts from the installation directory to set variables: `odbc.sh` or `odbc.csh`. For Korn, Bourne, and equivalent shells, execute `odbc.sh`. For a C shell, execute `odbc.csh`. After running the setup script, execute:
 

```
env
```

to verify that the *installation\_directory/lib* directory has been added to your shared library path.
- 4 Set the ODBCINI environment variable. The variable must point to the path from the root directory to the system information file where your data source resides. The system information file can have any name, but the product is installed with a default file called `odbc.ini` in the product installation directory. For example, if you use an installation directory of `/opt/odbc` and the default system information file, from the Korn or Bourne shell, you would enter:

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

From the C shell, you would enter:

```
setenv ODBCINI /opt/odbc/odbc.ini
```

## Test Loading the Driver

The `ivtestlib` (32-bit drivers) and `ddtestlib` (64-bit drivers) test loading tools are provided to test load drivers and help diagnose configuration problems in the UNIX and Linux environments, such as environment variables not correctly set or missing database client components. This tool is installed in the `/bin` subdirectory in the product installation directory. It attempts to load a specified ODBC driver and prints out all available error information if the load fails.

For example, if the drivers are installed in `/opt/odbc/lib`, the following command attempts to load the 32-bit Oracle Wire Protocol driver on Solaris, where `xx` represents the version number of the driver:

```
ivtestlib /opt/odbc/lib/ivoraxx.so
```

NOTE: On Solaris, AIX, and Linux, the full path to the driver does not have to be specified for the tool. The HP-UX version, however, requires the full path.

If the load is successful, the tool returns a success message along with the version string of the driver. If the driver cannot be loaded, the tool returns an error message explaining why.

## Setting the Library Path Environment Variable (Salesforce Driver on UNIX/Linux)

Before you can use the Salesforce driver, you must set the library path environment variable for your UNIX/Linux operating system to the directory containing your JVM's `libjvm.so` [`sl` | `a`] file, and that directory's parent directory.

NOTE FOR HP-UX: You also must set the `LD_PRELOAD` environment variable to the fully qualified path of the `libjvm.so`.

### 32-bit Salesforce Driver: Library Path Environment Variable

Set the library path environment variable to the directory containing your 32-bit JVM's `libjvm.so` [`sl` | `a`] file, and that directory's parent directory.

- `LD_LIBRARY_PATH` on Solaris, Linux, and HP-UX (Itanium)
- `SHLIB_PATH` on HP PA-RISC
- `LIBPATH` on AIX

### 64-bit Salesforce Driver: Library Path Environment Variable

Set the library path environment variable to the directory containing your 64-bit JVM's `libjvm.so` [`sl` | `a`] file, and that directory's parent directory.

- `LD_LIBRARY_PATH` on Solaris, HP-UX (Itanium), and Linux
- `LIBPATH` on AIX

## Configuring a Data Source

If you have Motif 2.0.3 or higher and one of the supported Linux operating systems, you can use the DataDirect ODBC Data Source Administrator for Linux (the Linux ODBC Administrator) to configure a data source. If you do not, see ["Configuration Through the System Information \(odbc.ini\) File" on page 103](#) for instructions on configuring the system information file.

NOTE: The Linux ODBC Administrator is currently supported only on Linux for x86 and x64 processors with Motif 2.0.3 or higher. It is not supported on Linux for Itanium II or other UNIX platforms.

The Linux ODBC Administrator is located in the /tools directory of the product installation directory. For example:

```
/opt/odbc/tools/odbcadmin
```

### To configure a data source:

- 1 To start the Linux ODBC Administrator, change to the *install\_dir*/tools directory, where *install\_dir* is the path to the product installation directory. At a command prompt, enter:
 

```
./odbcadmin
```
- 2 Click either the **User DSN** or **File DSN** tab to display a list of data sources.
  - **User DSN**: Select the appropriate data source name and click **Configure** to display the driver Setup dialog box.
 

If you are configuring a new user data source, click **Add** to display a list of installed drivers. Select the appropriate driver and click **Finish** to display the driver Setup dialog box.
  - **File DSN**: To configure a new file data source, click **Add** to display a list of installed drivers. Select the appropriate driver and click **Advanced** to specify attributes; otherwise, click **Next** to proceed. Specify a name for the data source and click **Next**. Verify the data source information; then, click **Finish** to display the driver Setup dialog box.

The General tab of the Setup dialog box appears by default.

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 in this book.

- 3 The following two options appear on the General tab of all driver Setup dialog boxes:
 

**Data Source Name**: Type a string that identifies this data source configuration, such as Accounting.

**Description**: Type an optional long description of a data source name, such as My Accounting Database.

Provide the requested information for all other options on the General tab; then, click **Apply** to configure the data source.

## Testing the Connection

- 1 After you have configured the data source, you can click **Test Connect** on the Setup dialog box to attempt to connect to the data source using the connection options specified in the dialog box. Some drivers immediately return a message indicating success or failure. For most drivers, a logon dialog box appears as described in each individual driver chapter.
- 2 Supply the requested information in the logon dialog box and click **OK**. 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 incorrect environment or connection value, it displays an appropriate error message. Click **OK**.
- 3 On the driver Setup dialog box, click **OK**. The values you have specified are saved and are the defaults used when you connect to the data source. You can change these defaults by using the previously described procedure to modify your data source. You can override these defaults by connecting to the data source using a connection string with alternate values. See individual driver chapters for information about using connection strings.

---

## Using the Performance Wizard

The Performance Wizard leads you step-by-step through a series of questions about your application. Based on your answers, the Wizard provides the optimal settings for performance-related connection string options. The Wizard applies to the following drivers:

- DB2 Wire Protocol
- Informix Wire Protocol
- MySQL Wire Protocol
- Oracle Wire Protocol
- SQL Server Wire Protocol
- Sybase Wire Protocol
- Oracle

The Wizard runs as an applet within a browser window. The browser must be configured to run applets. Refer to your browser's documentation for instructions on configuring your browser.

**NOTE:** Security features set in your browser can prevent the Performance Wizard from launching. If this is the case, a security warning message is displayed. Often, the warning message provides instructions for unblocking the Performance Wizard for the current session. To allow the Performance Wizard to launch without encountering a security warning message, the security settings in your browser can be modified. Check with your system administrator before disabling any security features.



## Starting the Wizard

You can start the Wizard in the following ways:

- On Windows, you can start the Wizard by selecting it from the product program group.
- On all platforms, you can start the Wizard by launching the following file from your browser window, where *install\_dir* is your product installation directory:

*install\_dir*/wizards/index.html

## Tuning Performance Using the Wizard

After you start the Wizard, a Welcome window appears. Click **Start** to start the process and select a driver.

The following is an example of one of the questions you may be asked to answer for the DB2 Wire Protocol driver.

Progress | DataDirect Connect THE WORLD LEADER IN DATA CONNECTIVITY

PROGRESS SOFTWARE

### PERFORMANCE WIZARD

DataDirect Connect® for ODBC  
DataDirect Connect64® for ODBC

**DB2 Wire Protocol**

- ☒ Choose Driver
- ☐ Stored Procedures
- ☐ Connection Pooling
- ☐ Multi-Threaded Application
- ☐ Failover
- ☐ Encryption
- ☐ Result

Do you need to access database objects (such as tables or stored procedures) that are grouped in different schemas (as opposed to accessing objects that are contained in a single schema)?

☒ Yes  
☐ No

< Back    Next >

**Detail:**

Applicable connection string attribute: UseCurrentSchema. If your application needs to access database objects owned only by the current user, performance of your application can be improved. In this case, the UseCurrentSchema attribute should be enabled (set to 1). When this attribute is enabled, the driver returns only database objects owned by the current user when executing catalog functions. Calls to catalog functions are optimized by grouping queries. Enabling this attribute is equivalent to passing the Logon ID used on the connection as the SchemaName argument to the catalog functions.

When you have answered all questions for a driver, the results appear in the form of a connection string, as shown in the following example:

Progress | DataDirect Connect THE WORLD LEADER IN DATA CONNECTIVITY PROGRESS SOFTWARE

## PERFORMANCE WIZARD

DataDirect Connect® for ODBC  
DataDirect Connect64® for ODBC

**DB2 Wire Protocol**

☒ Choose Driver

☒ Stored Procedures

☒ Connection Pooling

☒ Multi-Threaded Application

☒ Failover

☒ Encryption

☐ Result

**To save these results, copy and paste the result text into a file.**

Add or replace the following values in your application's connection string for the DB2 Wire Protocol driver.

```
MaxPoolSize=125;MinPoolSize=5;ApplicationUsingThreads=1;LoadBalanceTimeout=1800;ConnectionReset=0;UseCurrentSchema=0;ConnectionPooling=1;
```

For values to set in the ODBC Administrator, click :  
Administrator

For values to set in the odbc.ini (UNIX and Linux) file, click :  
ODBC.INI

< Back
Tune Another Driver
Exit

You can copy these results to an existing connection string for immediate use or to a text file for later reference.

You can also either click **Administrator**, if you are using the Windows or Linux ODBC administrator, or **ODBC.INI**, if you are editing the odbc.ini file. Clicking either of these buttons displays a window that provides the values to use for configuring a data source.

See ["Data Source Configuration" on page 100](#) for details about configuring data sources through the odbc.ini file.

## 2 Using The Product

This chapter contains the following sections:

- ["What Is ODBC?" on page 43](#)
- ["About the Product" on page 44](#)
- ["Environment-Specific Information" on page 45](#)
- ["Using IP Addresses" on page 54](#)
- ["Binding Parameter Markers" on page 55](#)
- ["Driver Threading Information" on page 56](#)
- ["Version String Information" on page 57](#)
- ["Retrieving Data Type Information" on page 59](#)
- ["Persisting a Result Set as an XML Data File" on page 60](#)
- ["Translators" on page 63](#)

---

### What Is ODBC?

The Open Database Connectivity (ODBC) interface by Microsoft allows applications to access data in database management systems (DBMS) using SQL as a standard for accessing the data. ODBC permits maximum interoperability, which means a single application can access different DBMS. Application end users can then add ODBC database drivers to link the application to their choice of DBMS.

The ODBC interface defines:

- A library of ODBC function calls of two types:
  - Extended functions that support additional functionality, including scrollable cursors
  - Core functions that are based on the X/Open and SQL Access Group Call Level Interface specification
- SQL syntax based on the X/Open and SQL Access Group SQL CAE specification (1992)
- A standard set of error codes
- A standard way to connect and logon to a DBMS
- A standard representation for data types

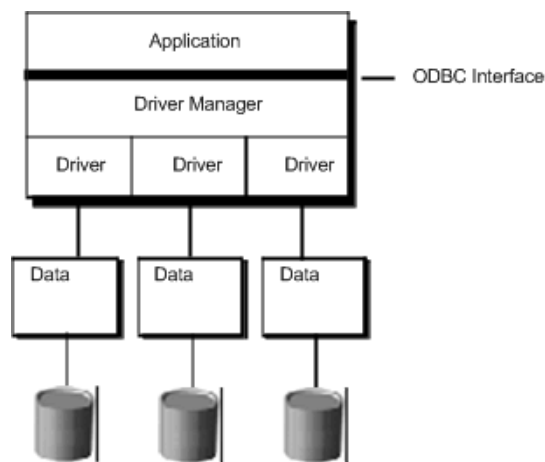
The ODBC solution for accessing data led to ODBC database drivers, which are dynamic-link libraries on Windows and shared objects on UNIX and Linux. These drivers allow an application to gain access to one or more data sources. ODBC provides a standard interface to allow application developers and vendors of database drivers to exchange data between applications and data sources.

## How Does It Work?

The ODBC architecture has four components:

- An application, which processes and calls ODBC functions to submit SQL statements and retrieve results
- A Driver Manager, which loads drivers for the application
- A driver, which processes ODBC function calls, submits SQL requests to a specific data source, and returns results to the application
- A data source, which consists of the data to access and its associated operating system, DBMS, and network platform (if any) used to access the DBMS

The following figure shows the relationship among the four components:



## Why Do Application Developers Need ODBC?

Using ODBC, you, as an application developer can develop, compile, and ship an application without targeting a specific DBMS. In this scenario, you do not need to use embedded SQL; therefore, you do not need to recompile the application for each new environment.

---

## About the Product

The DataDirect Connect Series *for* ODBC drivers are compliant with the Open Database Connectivity (ODBC) specification.

Progress DataDirect provides ODBC drivers for both relational and flat-file database systems. The flat-file drivers provide full SQL support; refer to [Chapter 12 “SQL Statements for Flat-File Drivers”](#) in the *DataDirect Connect Series for ODBC Reference* for details.

## Support for Multiple Environments

Progress DataDirect provides ODBC-compliant database drivers for Windows, UNIX, and Linux operating systems. See ["Environment-Specific Information" on page 45](#) for an explanation of the environment-specific differences when using the database drivers in your operating environment.

Database drivers are continually being added to each operating environment. For the latest information about the specific drivers available for your platform, refer to the Progress DataDirect database support matrix Web pages at:

<http://www.datadirect.com/products/odbc/matrix/connectodbc.htm> (32-bit)

<http://www.datadirect.com/products/odbc64/matrix/connect64odbc.htm> (64-bit)

---

## Environment-Specific Information

The sections ["For Windows Users" on page 45](#) and ["For UNIX and Linux Users" on page 47](#) contain information specific to your operating environment.

The following sections refer to threading models. Refer to [Chapter 3 "Threading"](#) in the *DataDirect Connect Series for ODBC Reference* for an explanation of threading.

### For Windows Users

The following are requirements for the 32- and 64-bit drivers on Windows operating systems.

#### 32-Bit Drivers

- All required network software that is supplied by your database system vendors must be 32-bit compliant.
- If your application was built with 32-bit system libraries, you must use 32-bit drivers. If your application was built with 64-bit system libraries, you must use 64-bit drivers (see ["64-Bit Drivers" on page 46](#)). The database to which you are connecting can be either 32-bit or 64-bit enabled.
- The following processors are supported:
  - x86: Intel
  - x64: Intel and AMD
- The following operating systems are supported for DataDirect Connect *for* ODBC. All editions are supported unless otherwise noted.
  - Windows 7
  - Windows Server 2008
  - Windows Vista
  - Windows XP, Service Pack 1 and higher

- Windows Server 2003
  - Windows 2000, Service Pack 1 and higher
- The following operating systems are supported for DataDirect Connect XE *for* ODBC. All editions are supported unless otherwise noted.
- Windows 7
  - Windows Server 2008 Enterprise, Datacenter, Web and Small Business Editions
  - Windows Vista
  - Windows XP Professional, Service Pack 2 and higher
  - Windows Server 2003, Service Pack 2 and higher
  - Windows 2000 Professional and Server, Service Pack 4
- For the Salesforce driver: A 32-bit Java Virtual Machine (JVM), J2SE 5 or higher, is required. Also, you must set the PATH environment variable to the directory containing your 32-bit JVM's jvm.dll file, and that directory's parent directory.
- An application that is compatible with components that were built using Microsoft Visual Studio 2010 compiler and the standard Win32 threading model.
- You must have ODBC header files to compile your application. For example, Microsoft Visual Studio includes these files.

## 64-Bit Drivers

- All required network software that is supplied by your database system vendors must be 64-bit compliant.
- The following processors are supported:
- Intel
  - AMD
- The following operating systems are supported for DataDirect Connect64 *for* ODBC. All editions are supported unless otherwise noted.
- Windows 7
  - Windows Server 2008
  - Windows Vista
  - Windows Server 2003
  - Microsoft Windows XP Professional Edition
- The following operating systems are supported for DataDirect Connect64 XE *for* ODBC. All editions are supported unless otherwise noted.
- Windows 7
  - Windows Server 2008 Enterprise, Standard, or Datacenter Editions
  - Windows Vista
  - Windows Server 2003, Service Pack 2 and higher
  - Windows XP Professional, Service Pack 2 and higher

- An application that is compatible with components that were built using Microsoft C/C++ Optimizing Compiler Version 14.00.40310.41 and the standard Windows 64 threading model.
- For the Salesforce driver: A 64-bit JVM, J2SE 5 or higher, is required. Also, you must set the PATH environment variable to the directory containing your 32-bit JVM's jvm.dll file, and that directory's parent directory.
- You must have ODBC header files to compile your application. For example, Microsoft Visual Studio includes these files.

## Setup of the Drivers

The drivers must be configured before they can be used. See [Chapter 1 "Quick Start Connect" on page 35](#) for information about using the Windows ODBC Administrator. See the individual driver chapters for details about driver configuration.

## Driver Names

The prefix for all 32-bit driver file names is IV. The prefix for all 64-bit driver file names is DD. The file extension is .DLL, which indicates dynamic link libraries. For example, the 32-bit DB2 Wire Protocol driver file name is IVDB2 $nn$ .DLL, where  $nn$  is the revision number of the driver.

Refer to the readme file shipped with the product for the file name of each driver.

## For UNIX and Linux Users

The following are requirements for the 32- and 64-bit drivers on UNIX/Linux operating systems.

### 32-Bit Drivers

- All required network software that is supplied by your database system vendors must be 32-bit compliant.
- If your application was built with 32-bit system libraries, you must use 32-bit drivers. If your application was built with 64-bit system libraries, you must use 64-bit drivers (see ["64-Bit Drivers" on page 50](#)). The database to which you are connecting can be either 32-bit or 64-bit enabled.
- For the Salesforce driver: A 32-bit Java Virtual Machine (JVM), J2SE 5 or higher, is required. Also, you must set the library path environment variable of your operating system to the directory containing your JVM's libjvm.so [sl | a] file and that directory's parent directory.

The library path environment variable is:

- LD\_LIBRARY\_PATH on Linux, HP-UX Itanium, and Oracle Solaris
- SHLIB\_PATH on HP-UX PA-RISC
- LIBPATH on AIX

**AIX**

- IBM POWER processor
- AIX 5L operating system, version 5.3 fixpack 5 and higher, 6.1, and 7.1
- An application compatible with components that were built using Visual Age C++ 6.0.0.0 and the AIX native threading model

NOTE FOR SALESFORCE USERS: When compiling an application on AIX for use with the driver for Salesforce, you must **not** use the `-brtl` option.

NOTE FOR TERADATA USERS: When compiling an application on AIX for use with the driver for the Teradata database, you must use the `-brtl` option. For example:

```
cc -o pgm pgm.o -brtl -lodbcc
```

or

```
ld -o pgm -brtl pgm.o -lodbcc
```

**HP-UX**

- The following processors are supported:
  - PA-RISC
  - Intel Itanium II (IPF)
- The following operating systems are supported:
  - For PA-RISC: HP-UX 11i Versions 2 and 3 (B.11.23 and B.11.31), 11i (B.11.11), and 11
  - For IPF: HP-UX IPF 11i Versions 2 and 3 (B.11.23 and B.11.31)
- For PA-RISC: An application compatible with components that were built using HP aC++ 3.30 and the HP-UX 11 native (kernel) threading model (posix draft 10 threads)
- For IPF: An application compatible with components that were built using HP aC++ 5.36 and the HP-UX 11 native (kernel) threading model (posix draft 10 threads)

NOTE: All of the standard 32-bit UNIX drivers are supported on HP PA-RISC. For IPF, the following drivers are supported:

DataDirect Connect *for* ODBC drivers

- DB2 Wire Protocol
- Informix Wire Protocol
- MySQL Wire Protocol
- Oracle Wire Protocol
- PostgreSQL Wire Protocol
- Progress OpenEdge Wire Protocol
- SQL Server Wire Protocol
- Sybase Wire Protocol
- Oracle
- SQL Server Legacy Wire Protocol

DataDirect Connect XE *for* ODBC drivers

- Driver for Apache Hive
- Salesforce
- Sybase IQ Wire Protocol



**NOTES FOR SALESFORCE:**

- PA-RISC: Set the LD\_PRELOAD environment variable to the libjvm.sl from your JVM installation.
- Itanium:
  - Do not link with the -lc linker option.
  - Set the LD\_PRELOAD environment variable to the libjvm.so from your JVM installation.

**Linux**

- The following processors are supported:
  - x86: Intel
  - x64: Intel and AMD
- The following operating systems are supported:
  - Red Hat Enterprise Linux 4.x, 5.x, and 6.x
  - SUSE Linux Enterprise Server 10.x, 11, and 12
- An application compatible with components that were built using g++ GNU project C++ Compiler version 3.4.6 and the Linux native pthread threading model (Linuxthreads).

NOTE: All drivers are supported on Linux except for the Informix driver.

**Oracle Solaris**

- The following processors are supported:
  - Oracle SPARC
  - x86: Intel
  - x64: Intel and AMD
- The following operating systems are supported:
  - For Oracle SPARC: Oracle Solaris 8, 9, and 10
  - For x86/x64: Oracle Solaris 10, Oracle Solaris 11
- For Oracle SPARC: An application compatible with components that were built using Oracle Workshop v. 6 update 2 and the Solaris native (kernel) threading model
- For x86/x64: An application compatible with components that were built using Oracle C++ 5.8 and the Solaris native (kernel) threading model

NOTE: All of the standard 32-bit UNIX drivers are supported on Solaris SPARC. For x86, the following drivers are supported:

DataDirect Connect <i>for</i> ODBC drivers	<ul style="list-style-type: none"> <li>■ DB2 Wire Protocol</li> <li>■ MySQL Wire Protocol</li> <li>■ Oracle Wire Protocol</li> <li>■ PostgreSQL Wire Protocol</li> <li>■ SQL Server Wire Protocol</li> <li>■ Sybase Wire Protocol</li> <li>■ SQL Server Legacy Wire Protocol</li> </ul>
DataDirect Connect XE <i>for</i> ODBC drivers	<ul style="list-style-type: none"> <li>■ Driver for Apache Hive</li> <li>■ Salesforce</li> <li>■ Sybase IQ Wire Protocol</li> </ul>

## 64-Bit Drivers

All required network software that is supplied by your database system vendors must be 64-bit compliant.

- For the Salesforce driver: A 64-bit Java Virtual Machine (JVM), J2SE 5 or higher, is required. Also, you must set the library path environment variable of your operating system to the directory containing your JVM's libjvm.so [sl | a] file and that directory's parent directory.
- The library path environment variable is:
  - LD\_LIBRARY\_PATH on Linux, HP-UX Itanium, and Oracle Solaris
  - LIBPATH on AIX

## AIX

- IBM POWER Processor
- AIX 5L operating system, version 5.3 fixpack 5 and higher, 6.1, and 7.1
- An application compatible with components that were built using Visual Age C++ version 6.0.0.0 and the AIX native threading model

NOTE FOR SALESFORCE USERS: When compiling an application on AIX for use with the driver for Salesforce, you must **not** use the `-brtl` option.

## HP-UX

- Intel Itanium II (IPF) processor
- HP-UX IPF 11i operating system, Versions 2 and 3 (B.11.23 and B.11.31)
- HP aC++ v. 5.36 and the HP-UX 11 native (kernel) threading model (posix draft 10 threads)

NOTE: The following drivers are supported on IPF:

DataDirect Connect64 <i>for</i> ODBC drivers	<ul style="list-style-type: none"> <li>■ DB2 Wire Protocol</li> <li>■ Informix Wire Protocol</li> <li>■ MySQL Wire Protocol</li> <li>■ Oracle Wire Protocol</li> <li>■ PostgreSQL Wire Protocol</li> <li>■ Progress OpenEdge Wire Protocol</li> <li>■ SQL Server Wire Protocol</li> <li>■ Sybase Wire Protocol</li> <li>■ Oracle</li> <li>■ SQL Server Legacy Wire Protocol</li> </ul>
DataDirect Connect64 XE <i>for</i> ODBC drivers	<ul style="list-style-type: none"> <li>■ Greenplum Wire Protocol</li> <li>■ Driver for Apache Hive</li> <li>■ Salesforce</li> <li>■ Sybase IQ Wire Protocol</li> <li>■ Teradata</li> </ul>

#### NOTES FOR SALESFORCE:

- Do not link with the `-lc` linker option.
- Set the `LD_PRELOAD` environment variable to the `libjvm.so` of your JVM installation.

#### Linux

- The following processors are supported:
  - Intel Itanium II (IPF)
  - x64: Intel and AMD
- The following operating systems are supported:
  - For Itanium II: Red Hat Enterprise Linux AS, ES, and WS versions 4.x and 5.x
  - For x64: Red Hat Enterprise Linux AS, ES, and WS version 4.x, 5.x, and 6.x

NOTE: The Oracle (client) driver is not supported on the Red Hat x64 operating system.

  - For x64: SUSE Linux Enterprise Server 10.x, 11, and 12
- For Itanium II: an application compatible with components that were built using g++ GNU project C++ Compiler version 3.3.2 and the Linux native pthread threading model (Linuxthreads)
- For x64: an application compatible with components that were built using g++ GNU project C++ Compiler version 3.4 and the Linux native pthread threading model (Linuxthreads)

NOTE: The Salesforce Driver, the Driver for Apache Hive, and the Driver for Teradata are not supported on Linux Itanium. II.

**Oracle Solaris**

- The following processors are supported:
  - Oracle SPARC
  - x64: Intel and AMD
- The following operating systems are supported:
  - For Oracle SPARC: Oracle Solaris 8, 9, and 10
  - For x64: Oracle Solaris 10 and Oracle Solaris 11 Express
- For Oracle SPARC: An application compatible with components that were built using Oracle Workshop v. 6 update 2 and the Solaris native (kernel) threading model
- For x64: An application compatible with components that were built using Oracle C++ Compiler version 5.8 and the Solaris native (kernel) threading model

NOTE: All of the standard 32-bit UNIX drivers are supported on Solaris SPARC. For x64, The following drivers are supported for Oracle Solaris:

**DataDirect Connect for ODBC drivers**

- DB2 Wire Protocol
- MySQL Wire Protocol
- Oracle Wire Protocol
- PostgreSQL Wire Protocol
- SQL Server Wire Protocol
- Sybase Wire Protocol
- SQL Server Legacy Wire Protocol

**DataDirect Connect XE for ODBC drivers**

- Greenplum Wire Protocol
- Driver for Apache Hive
- Salesforce
- Sybase IQ Wire Protocol

**AIX**

If you are building 64-bit binaries, you must pass the define ODBC64. Demoodbc provides an example of this. See the installed file demoodbc.txt and ["The demoodbc Application" on page 119](#) for details.

You must also include the correct compiler switches if you are building 64-bit binaries. For example, to build demoodbc, you would use:

```
xlC_r -DODBC64 -q64 -qlonglong -qlongdouble -qvftable -o demoodbc
-I../include demoodbc.c -L../lib -lc_r -lC_r -lodbc
```

**HP-UX 11 aCC**

The ODBC drivers require certain runtime library patches. The patch numbers are listed in the readme file for your product. HP-UX patches are publicly available from the HP Web site [www.hp.com](http://www.hp.com).

HP updates the patch database regularly; therefore, the patch numbers in the readme file may be superseded by newer versions. If you search for the specified patch on an HP site

and receive a message that the patch has been superseded, download and install the replacement patch.

If you are building 64-bit binaries, you must pass the define ODBC64. Demoodbc provides an example of this. See the installed file demoodbc.txt and ["The demoodbc Application" on page 119](#) for details. You must also include the +DD64 compiler switch if you are building 64-bit binaries. For example, to build demoodbc, you would use:

```
aCC -Wl,+s +DD64 -DODBC64 -o demoodbc -I../include demoodbc.c -L../lib -lodbc
```

## Linux

If you are building 64-bit binaries, you must pass the define ODBC64. Demoodbc provides an example of this. See the installed file demoodbc.txt and ["The demoodbc Application" on page 119](#) for details.

You must also include the correct compiler switches if you are building 64-bit binaries. For example, to build demoodbc, you would use:

```
g++ -o demoodbc -DODBC64 -I../include demoodbc.c -L../lib -lodbc -lodbcinst  
-lc
```

## Oracle Solaris

If you are building 64-bit binaries, you must pass the define ODBC64. Demoodbc provides an example of this. See the installed file demoodbc.txt and ["The demoodbc Application" on page 119](#) for details.

You must also include the -xarch=v9 compiler switch if you are building 64-bit binaries. For example, to build demoodbc, you would use:

```
CC -mt -DODBC64 -xarch=v9 -o demoodbc -I../include demoodbc.c -L../lib -lodbc  
-lCrun
```

## Setup of the Environment and the Drivers

On UNIX and Linux, several environment variables and the system information file must be configured before the drivers can be used. See [Chapter 1 "Quick Start Connect" on page 35](#) for a brief description of these variables and information about using the DataDirect ODBC Data Source Administrator for Linux. See the individual driver chapters for details about driver configuration. See [Chapter 4 "Configuring the Product on UNIX/Linux" on page 97](#) for complete information about using the drivers on UNIX and Linux.

## Driver Names

The drivers are ODBC API-compliant dynamic link libraries, referred to in UNIX and Linux as shared objects. The prefix for all 32-bit driver file names is iv. The prefix for all 64-bit driver file names is dd. The driver file names are lowercase and the extension is .so, the standard form for a shared object. For example, the 32-bit DB2 Wire Protocol driver file name is ivdb2nn.so, where nn is the revision number of the driver. For drivers on HP-UX PA-RISC only, the extension is .sl, for example, ivdb2nn.sl.

Refer to the readme file shipped with your DataDirect product for the file name of each driver.

## Using IP Addresses

The drivers support Internet Protocol (IP) addresses in IPv4 and IPv6 format as shown in [Table 2-1](#) and [Table 2-2](#).

**Table 2-1. IP Address Formats Supported by 32- and 64-bit DataDirect Connect for ODBC Drivers**

Driver	IPv4	IPv6
DB2 Wire Protocol	All supported versions	DB2 9.1 for Linux/UNIX/Windows and higher DB2 V5R3 for iSeries and higher
Informix Wire Protocol	All supported versions	Informix 10 and higher
MySQL Wire Protocol	All supported versions	Not supported
Oracle Wire Protocol	All supported versions	Oracle 11gR2
Microsoft SQL Server Legacy Wire Protocol (UNIX/Linux only)	All supported versions	Microsoft SQL Server 2005 and higher
Microsoft SQL Server Wire Protocol	All supported versions	Microsoft SQL Server 2005 and higher
PostgreSQL Wire Protocol	All supported versions	PostgreSQL 8.2 and higher
Progress OpenEdge Wire Protocol	All supported versions	Progress OpenEdge 10.0 and higher
Sybase Wire Protocol	All supported versions	Sybase 12.5.2 and higher

**Table 2-2. IP Address Formats Supported by 32- and 64-bit DataDirect Connect XE for ODBC Drivers**

Driver	IPv4	IPv6
Driver for Apache Hive	All supported versions	All supported versions
Greenplum Wire Protocol	All supported versions	not supported
Salesforce	All supported versions	All supported versions
Sybase IQ Wire Protocol	All supported versions	All supported versions

If your network supports named servers, the server name specified in the data source can resolve to an IPv4 or IPv6 address.

In the following connection string example, the IP address for the DB2 server is specified in IPv6 format:

```
DRIVER=DataDirect DB2 Wire Protocol;
IpAddress=2001:DB8:0000:0000:8:800:200C:417A;PORT=5179;
DB=DB2ACCT;UID=JOHN;PWD=XYZZYYou
```

In addition to the normal IPv6 format, the drivers in [Table 2-1](#) and [Table 2-2](#) support IPv6 alternative formats for compressed and IPv4/IPv6 combination addresses. For example, the

following connection string specifies the server using IPv6 format, but uses the compressed syntax for strings of zero bits:

```
DRIVER=DataDirect DB2 Wire Protocol;
IpAddress=2001:DB8:0:0:8:800:200C:417A;PORT=5179;
DB=DB2ACCT;UID=JOHN;PWD=XYZZYYou
```

Similarly, the following connection string specifies the server using a combination of IPv4 and IPv6:

```
DRIVER=DataDirect DB2 Wire Protocol;
IpAddress=2001:DB8:0:0:8:800:123.1.23.4;PORT=5179;
DB=DB2ACCT;UID=JOHN;PWD=XYZZYYou
```

For complete information about IPv6 formats, go to the following URL:

<http://tools.ietf.org/html/rfc4291#section-2.2>

---

## Binding Parameter Markers

An ODBC application can prepare a query that contains dynamic parameters. Each parameter in a SQL statement must be associated, or bound, to a variable in the application before the statement is executed. When the application binds a variable to a parameter, it describes that variable and that parameter to the driver. Therefore, the application must supply the following information:

- The data type of the variable that the application maps to the dynamic parameter
- The SQL data type of the dynamic parameter (the data type that the database system assigned to the parameter marker)

The two data types are identified separately using the `SQLBindParameter` function. You can also use descriptor APIs as described in the Descriptor section of the ODBC specification (version 3.0 and higher).

The driver relies on the binding of parameters to know how to send information to the database system in its native format. If an application furnishes incorrect parameter binding information to the ODBC driver, the results will be unpredictable. For example, the statement might not be executed correctly.

To ensure interoperability, the DataDirect Connect Series *for* ODBC drivers use only the parameter binding information that is provided by the application. Some DBMSs cannot publish dynamic parameter information back to an ODBC driver. For example, both the Microsoft SQL Server and Oracle databases can determine that a parameter is an integer; however, the Oracle query processor cannot publish this information back to the driver.

## Driver Threading Information

[Table 2-3](#) and [Table 2-4](#) summarize the threading information available at this time for the drivers. Always consult the readme file for the most up-to-date information as threading information is subject to change with new database transport and server revisions. Currently, the XML driver is the only thread-impaired driver. Refer to [Chapter 3 “Threading”](#) in the *DataDirect Connect Series for ODBC Reference* for more information about threading and how to obtain reading information through SQLGetInfo.

**Table 2-3. Threading Information for DataDirect Connect for ODBC**

Driver	Fully Threaded	Thread Per Connect
Btrieve	X	
dBASE	X	
DB2 Wire Protocol		X
Informix Wire Protocol		X
Informix		X
MySQL Wire Protocol		X
Oracle Wire Protocol		X
Oracle		X
PostgreSQL Wire Protocol		X
Progress OpenEdge		X
SQL Server Legacy Wire Protocol		X
SQL Server Wire Protocol		X
Sybase Wire Protocol		X
Text	X	

**Table 2-4. Threading Information for DataDirect Connect XE for ODBC**

Driver	Fully Threaded	Thread Per Connect
Driver for Apache Hive		X <sup>1</sup>
Greenplum Wire Protocol		X
Salesforce		X
Sybase IQ Wire Protocol		X
Teradata	X	

1. The Apache Hive server process has known problems with handling concurrent operations. We recommend that you start a new Hive server per connection until these issues are resolved.



## Version String Information

All drivers, except the flat-file drivers and the Salesforce driver, have a version string of the format:

```
XX.YY.ZZZZ(BAAAA, UBBBB)
```

or

```
XX.YY.ZZZZ(bAAAA, uBBBB)
```

All flat-file drivers have a version string of the format:

```
XX.YY.ZZZZ(bAAAA, uBBBB, FCCCC)
```

The Salesforce driver has a version string of the format:

```
XX.YY.ZZZZ(BAAAA, UBBBB, SDDDDDD)
```

The Driver Manager on UNIX and Linux has a version string of the format:

```
XX.YY.ZZZZ(UBBBB)
```

The component for the Unicode conversion tables (ICU) has a version string of the format:

```
XX.YY.ZZZZ
```

where:

*XX* is the major version of the product.

*YY* is the minor version of the product.

*ZZZZ* is the build number of the driver or ICU component.

*AAAA* is the build number of the driver's bas component.

*BBBB* is the build number of the driver's utl component.

*CCCC* is the build number of a flat-file driver's flt component.

*DDDDDD* is the version of the Java components used by the Salesforce driver.

For example:

```
07.10.0002 (b0001, u0002, F0001)
    |__|  |__|  |__|  |__|
    Driver Bas  Utl  Flt
```

On Windows, you can check the version string through the properties of the driver DLL. Right-click the driver DLL and select **Properties**. The Properties dialog box appears. On the Version tab, click **File Version** in the Other version information list box.

You can always check the version string of a driver on Windows by looking at the About tab of the driver's Setup dialog.

On UNIX and Linux, you can check the version string by using the test loading tool shipped with the product. This tool, `ivtestlib` for 32-bit drives and `ddtestlib` for 64-bit drivers, is located in `install_directory/bin`.

The syntax for the tool is:

```
ivtestlib shared_object
```

or

```
ddtestlib shared_object
```

For example, for the 32-bit Oracle Wire Protocol driver on Oracle Solaris:

```
ivtestlib ivora27.so
```

returns:

```
07.10.0001 (B0002, U0001)
```

Note that the Oracle Wire Protocol driver is not a flat-file driver; therefore, there is no fit component listed in the example.

For example, for the 64-bit Driver Manager on Solaris:

```
ddtestlib libodbc.so
```

returns:

```
07.10.0001 (U0001)
```

For example, for 32-bit ICU component on Solaris:

```
ivtestlib libivicu27.so
```

```
07.10.0001
```

NOTE: On AIX, Linux, and Solaris, the full path to the driver does not have to be specified for the test loading tool. The HP-UX version of the tool, however, requires the full path.

## getFileVersionString Function

Version string information can also be obtained programmatically through the function `getFileVersionString`. This function can be used when the application is not directly calling ODBC functions.

This function is defined as follows and is located in each driver's shared object:

```
const unsigned char* getFileVersionString();
```

This function is prototyped in the `qesqlx.h` file shipped with the product.

## Retrieving Data Type Information

At times, you might need to get information about the data types that are supported by the data source, for example, precision and scale. You can use the ODBC function `SQLGetTypeInfo` to do this.

On Windows, you can use ODBC Test to call `SQLGetTypeInfo` against the ODBC data source to return the data type information. Refer to [Chapter 1 “Diagnostic Tools”](#) in the *DataDirect Connect Series for ODBC Troubleshooting Guide* for details about ODBC Test.

On UNIX, Linux, or Windows, an application can call `SQLGetTypeInfo`. Here is an example of a C function that calls `SQLGetTypeInfo` and retrieves the information in the form of a SQL result set.

```
void ODBC_GetTypeInfo(SQLHANDLE hstmt, SQLSMALLINT dataType)
{
    RETCODE rc;

    // There are 19 columns returned by SQLGetTypeInfo.
    // This example displays the first 3.
    // Check the ODBC 3.x specification for more information.

    // Variables to hold the data from each column
    char          typeName[30];
    short         sqlDataType;
    unsigned long columnSize;

    SQLINTEGER    strlenTypeName,
                  strlenSqlDataType,
                  strlenColumnSize;

    rc = SQLGetTypeInfo(hstmt, dataType);
    if (rc == SQL_SUCCESS) {

        // Bind the columns returned by the SQLGetTypeInfo result set.
        rc = SQLBindCol(hstmt, 1, SQL_C_CHAR, &typeName,
                        (SDWORD)sizeof(typeName), &strlenTypeName);
        rc = SQLBindCol(hstmt, 2, SQL_C_SHORT, &sqlDataType,
                        (SDWORD)sizeof(sqlDataType), &strlenSqlDataType);
        rc = SQLBindCol(hstmt, 3, SQL_C_LONG, &columnSize,
                        (SDWORD)sizeof(columnSize), &strlenColumnSize);

        // Print column headings
        printf ("TypeName      DataType      ColumnSize\n");
        printf ("-----\n");

        do {
            // Fetch the results from executing SQLGetTypeInfo
            rc = SQLFetch(hstmt);
            if (rc == SQL_ERROR) {
```

```

// Procedure to retrieve errors from the SQLGetTypeInfo function
        ODBC_GetDiagRec(SQL_HANDLE_STMT, hstmt);
        break;
    }

// Print the results
    if ((rc == SQL_SUCCESS) || (rc == SQL_SUCCESS_WITH_INFO)) {
printf ("%30s %10i %10u\n", typeName, sqlDataType, columnSize);
    }

    } while (rc != SQL_NO_DATA);
}
}

```

For information about how a database's data types map to the standard ODBC data types, see the appropriate driver chapter in this book.

---

## Persisting a Result Set as an XML Data File

The DataDirect Connect Series *for* ODBC drivers allow you to persist a result set as an XML data file with embedded schema. To implement XML persistence, a client application must do the following:

- 1 Turn on STATIC cursors. For example:

```
SQLSetStmtAttr (hstmt, SQL_ATTR_CURSOR_TYPE, SQL_CURSOR_STATIC,
SQL_IS_INTEGER)
```

**NOTE:** A result set can be persisted as an XML data file only if the result set is generated using STATIC cursors. Otherwise, the following error is returned:

Driver only supports XML persistence when using driver's static cursors.

- 2 Execute a SQL statement. For example:

```
SQLExecDirect (hstmt, "SELECT * FROM GTABLE", SQL_NTS)
```

- 3 Persist the result set as an XML data file. For example:

```
SQLSetStmtAttr (hstmt, SQL_PERSIST_AS_XML, "C:\temp\GTABLE.XML", SQL_NTS)
```

**NOTE:** A statement attribute is available to support XML persistence, SQL\_PERSIST\_AS\_XML. A client application must call SQLSetStmtAttr with this attribute as an argument. See the following table for the definition of valid arguments for SQLSetStmtAttr.

Argument	Definition
<i>StatementHandle</i>	The handle of the statement that contains the result set to persist as XML.
<i>Attribute</i>	SQL_PERSIST_AS_XML. This statement attribute can be found in the file qesqlxt.h, which is installed with the driver.

Argument	Definition
<i>ValuePtr</i>	Pointer to a URL that specifies the full path name of the XML data file to be generated. The directory specified in the path name must exist, and if the specified file name exists, the file will be overwritten.
<i>StringLength</i>	The length of the string pointed to by ValuePtr or SQL_NTS if ValuePtr points to a NULL-terminated string.

A client application can choose to persist the data at any time that the statement is in an executed or cursor-positioned state. At any other time, the driver returns the following message:

Function Sequence Error

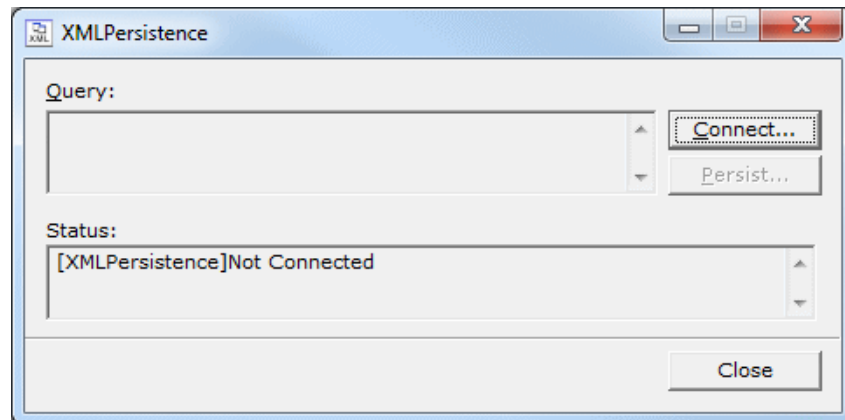
## Using the Windows XML Persistence Demo Tool

The 32-bit drivers for Windows are shipped with an XML persistence demo tool. This tool is installed in the product installation directory.

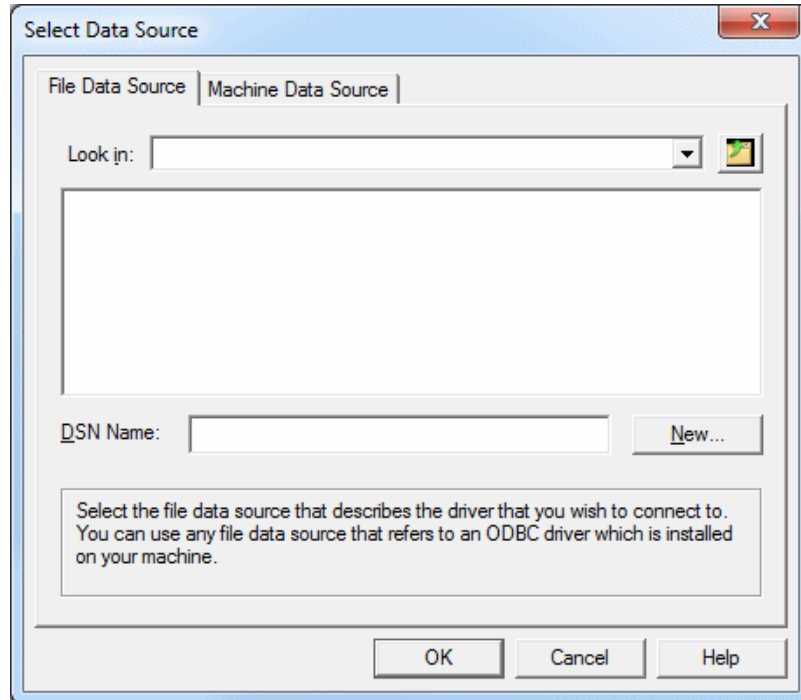
The tool has a graphical user interface and allows you to persist data as an XML data file.

### To use the Windows XML Persistence Demo tool:

- 1 From the product program group, select **XML Persistence Demo**. The XMLPersistence dialog box appears.



- 2 First, you must connect to the database. Click **Connect**. The Select Data Source dialog box appears.



- 3 You must either select an existing data source or create a new one. Take one of the following actions:
  - Select an existing data source and click **OK**.
  - Create a new file data source by clicking **New**. The Create New Data Source dialog box appears. Follow the instructions in the dialog box.
  - Create a new machine data source by clicking the **Machine Data Source** tab and clicking **New**. The Create New Data Source dialog box appears. Follow the instructions in the dialog box.
- 4 After you have connected to a database, type a SQL Select statement in the Query text box of the XML Persistence dialog box. Then, click **Persist**. The Save As dialog box appears.
- 5 Specify a name and location for the XML data file that will be created. Then, click **OK**.  
Note that the Status box in the XML Persistence dialog box displays whether the action failed or succeeded.
- 6 Click **Disconnect** to disconnect from the database.
- 7 Click **Close** to exit the tool.

## Using the UNIX/Linux XML Persistence Demo Tool

On UNIX and Linux, the drivers are shipped with an XML persistence demo tool named demoodbc. This tool is installed in the installation directory, in the /samples/demo subdirectory. For information about how to use this tool, refer to the demoodbc.txt file installed in the demo subdirectory.

---

## Translators

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

On Windows, refer to the readme.trn file in the \TRANSLAT subdirectory in the product installation directory.

On UNIX and Linux, refer to the readme.trn file in the product installation directory, in the /samples/trn subdirectory.





## 3 Advanced Features

This chapter contains the following sections:

- ["Using Failover" on page 65](#)
- ["Using Client Information" on page 75](#)
- ["Using Security" on page 77](#)
- ["Using DataDirect Connection Pooling" on page 82](#)
- ["Using DataDirect Bulk Load" on page 85](#)

---

### Using Failover

To ensure continuous, uninterrupted access to data, the DataDirect Connect Series *for* ODBC drivers provide the following levels of failover protection, listed from basic to more comprehensive:

- *Connection failover* provides failover protection for new connections only. The driver fails over new connections to an alternate, or backup, database server if the primary database server is unavailable, for example, because of a hardware failure or traffic overload. If a connection to the database is lost, or dropped, the driver does not fail over the connection. This failover method is the default.
- *Extended connection failover* provides failover protection for new connections and lost database connections. If a connection to the database is lost, the driver fails over the connection to an alternate server, preserving the state of the connection at the time it was lost, but not any work in progress.
- *Select Connection failover* provides failover protection for new connections and lost database connections. In addition, it provides protection for Select statements that have work in progress. If a connection to the database is lost, the driver fails over the connection to an alternate server, preserving the state of the connection at the time it was lost and preserving the state of any work being performed by Select statements.

The method you choose depends on how failure tolerant your application is. For example, if a communication failure occurs while processing, can your application handle the recovery of transactions and restart them? Your application needs the ability to recover and restart transactions when using either extended connection failover mode or select connection failover mode. The advantage of select mode is that it preserves the state of any work that was being performed by the Select statement at the time of connection loss. If your application had been iterating through results at the time of the failure, when the connection is reestablished the driver can reposition on the same row where it stopped so that the application does not have to undo all of its previous result processing. For example, if your application were paging through a list of items on a Web page when a failover occurred, the next page operation would be seamless instead of starting from the beginning. Performance, however, is a factor in selecting a failover mode. Select mode incurs additional overhead when tracking what rows the application has already processed.

You can specify which failover method you want to use by setting the [Failover Mode](#) connection option. Read the following sections for details on each failover method:

- ["Connection Failover" on page 66](#)
- ["Extended Connection Failover" on page 67](#)
- ["Select Connection Failover" on page 69](#)

Regardless of the failover method you choose, you must configure one or multiple alternate servers using the [Alternate Servers](#) connection option. See ["Guidelines for Primary and Alternate Servers" on page 70](#) for information about primary and alternate servers.

## Connection Failover

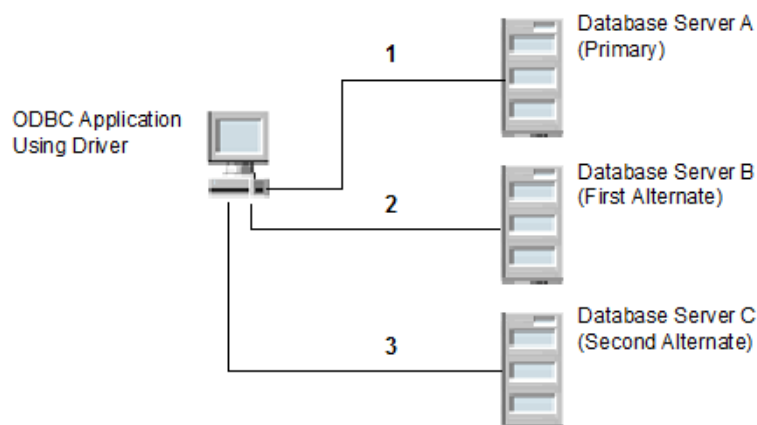
Connection failover is available in the following DataDirect Connect Series *for* ODBC drivers:

- DB2 Wire Protocol
- Greenplum Wire Protocol
- Informix Wire Protocol
- MySQL Wire Protocol
- Oracle Wire Protocol
- Oracle
- PostgreSQL Wire Protocol
- Progress OpenEdge Wire Protocol
- SQL Server Wire Protocol
- SQL Server Legacy Wire Protocol (UNIX only)
- Sybase Wire Protocol
- Sybase IQ Wire Protocol

Connection failover allows an application to connect to an alternate, or backup, database server if the primary database server is unavailable, for example, because of a hardware failure or traffic overload. Connection failover provides failover protection for new connections only and does not provide protection for lost connections to the database, nor does it preserve states for transactions or queries.

You can customize the drivers for connection failover by configuring a list of alternate database servers that are tried if the primary server is not accepting connections. Connection attempts continue until a connection is successfully established or until all the alternate database servers have been tried the specified number of times.

For example, suppose you have the environment shown in the following illustration with multiple database servers: Database Server A, B, and C. Database Server A is designated as the primary database server, Database Server B is the first alternate server, and Database Server C is the second alternate server.



First, the application attempts to connect to the primary database server, Database Server A (1). If connection failover is enabled and Database Server A fails to accept the connection, the application attempts to connect to Database Server B (2). If that connection attempt also fails, the application attempts to connect to Database Server C (3).

In this scenario, it is probable that at least one connection attempt would succeed, but if no connection attempt succeeds, the driver can retry each alternate database server (primary and alternate) for a specified number of attempts. You can specify the number of attempts that are made through the *connection retry* feature. You can also specify the number of seconds of delay, if any, between attempts through the *connection delay* feature. See ["Using Connection Retry" on page 72](#) for more information about connection retry.

A driver fails over to the next alternate database server only if a successful connection cannot be established with the current alternate server. If the driver successfully establishes communication with a database server and the connection request is rejected by the database server because, for example, the login information is invalid, then the driver generates an error and does not try to connect to the next database server in the list. It is assumed that each alternate server is a mirror of the primary and that all authentication parameters and other related information are the same.

For details on configuring connection failover for your driver, see the appropriate driver chapter in this book.

## Extended Connection Failover

Extended connection failover is available in the following DataDirect Connect Series *for* ODBC drivers:

- DB2 Wire Protocol
- Greenplum Wire Protocol
- MySQL Wire Protocol
- Oracle Wire Protocol
- PostgreSQL Wire Protocol
- Progress OpenEdge Wire Protocol
- SQL Server Wire Protocol

- Sybase Wire Protocol
- Sybase IQ Wire Protocol

Extended connection failover provides failover protection for the following types of connections:

- New connections, in the same way as described in ["Connection Failover" on page 66](#)
- Lost connections

When a connection to the database is lost, the driver fails over the connection to an alternate server, restoring the same state of the connection at the time it was lost. For example, when reestablishing a lost connection on the alternate database server, the driver performs the following actions:

- Restores the connection using the same connection options specified by the lost connection
- Reallocates statement handles and attributes
- Logs in the user to the database with the same user credentials
- Restores any prepared statements associated with the connection
- Restores manual commit mode if the connection was in manual commit mode at the time of the failover

The driver does not preserve work in progress. For example, if the database server experienced a hardware failure while processing a query, partial rows processed by the database and returned to the client would be lost. If the driver was in manual commit mode and one or more Inserts or Updates were performed in the current transaction before the failover occurred, then the transaction on the primary server is rolled back. The Inserts or Updates done before the failover are not committed to the primary server. Your application needs to rerun the transaction after the failover because the Inserts or Updates done before the failover are not repeated by the driver on the failover connection.

When a failover occurs, if a statement is in allocated or prepared state, the next operation on the statement returns a SQL state of 01000 and a vendor code of 0. If a statement is in an executed or prepared state, the next operation returns a SQL state of 40001 and a vendor code of 0. Either condition returns an error message similar to:

```
Your connection has been terminated. However, you have been successfully
connected to the next available AlternateServer: 'HOSTNAME=Server4:PORTNUMBER=
1521:SERVICENAME=test'. All active transactions have been rolled back.
```

The driver retains all connection settings made through ODBC API calls when a failover connection is made. It does not, however, retain any session settings established through SQL statements. This can be done through the Initialization String connection option, described in the individual driver chapters.

The driver retains the contents of parameter buffers, which can be important when failing over after a fetch. All Select statements are re-prepared at the time the failover connection is made. All other statements are placed in an allocated state.

If an error occurs while the driver is reestablishing a lost connection, the driver can fail the entire failover process or proceed with the process as far as it can. For example, suppose an error occurred while reestablishing the connection because a table for which the driver had a prepared statement did not exist on the alternate connection. In this case, you may

want the driver to notify your application of the error and proceed with the failover process. You can choose how you want the driver to behave if errors occur during failover by setting the [Failover Granularity](#) connection option.

During the failover process, your application may experience a short pause while the driver establishes a connection on an alternate server. If your application is time-sensitive (a real-time customer order application, for example) and cannot absorb this wait, you can set the [Failover Preconnect](#) connection option to true. Setting the Failover Preconnect option to true instructs the driver to establish connections to the primary server and an alternate server at the same time. Your application uses the first connection that is successfully established. If this connection to the database is lost at a later time, the driver saves time in reestablishing the connection on the server to which it fails over because it can use the spare connection in its failover process.

This pre-established failover connection is not used by the driver until the driver determines that it needs to fail over. If the server to which the driver is connected or the network equipment through which the connection is routed is configured with a timeout, the pre-configured failover connection could time out. The pre-configured failover connection can also be lost if the failover server is brought down and back up again. The driver tries to establish the connection to the failover server again if the connection is lost.

## Select Connection Failover

Select connection failover is available in the following DataDirect Connect Series *for* ODBC drivers:

- DB2 Wire Protocol
- Greenplum Wire Protocol
- MySQL Wire Protocol
- Oracle Wire Protocol
- PostgreSQL Wire Protocol
- Progress OpenEdge Wire Protocol
- SQL Server Wire Protocol
- Sybase Wire Protocol
- Sybase IQ Wire Protocol

Select connection failover provides failover protection for the following types of connections:

- New connections, in the same way as described in ["Connection Failover" on page 66](#)
- Lost connections, in the same way as described in ["Extended Connection Failover" on page 67](#)

In addition, the driver can recover work in progress because it keeps track of the last Select statement the application executed on each Statement handle, including how many rows were fetched to the client. For example, if the database had only processed 500 of 1,000 rows requested by a Select statement when the connection was lost, the driver would reestablish the connection to an alternate server, re-execute the Select statement, and position the cursor on the next row so that the driver can continue fetching the balance of rows as if nothing had happened.

Performance, however, is a factor when considering whether to use Select mode. Select mode incurs additional overhead when tracking what rows the application has already processed.

NOTE: The driver only recovers work requested by Select statements. You must explicitly restart the following types of statements after a failover occurs:

- Insert, Update, or Delete statements
- Statements that modify the connection state, for example, SET or ALTER SESSION statements
- Objects stored in a temporary tablespace or global temporary table
- Partially executed stored procedures and batch statements

When in manual transaction mode, no statements are rerun if any of the operations in the transaction were Insert, Update, or Delete. This is true even if the statement in process at the time of failover was a Select statement.

By default, the driver verifies that the rows that are restored match the rows that were originally fetched and, if they do not match, generates an error warning your application that the Select statement must be reissued. By setting the [Failover Granularity](#) connection option, you can customize the driver to ignore this check altogether or fail the entire failover process if the rows do not match.

When the row comparison does not agree, the default behavior of Failover Granularity returns a SQL state of 40003 and an error message similar to:

```
Unable to position to the correct row after a successful failover attempt to
AlternateServer: 'HOSTNAME=Server4:PORTNUMBER= 1521:SERVICE=test'. You must
reissue the select statement.
```

If you have configured Failover Granularity to fail the entire failover process, the driver returns a SQL state of 08S01 and an error message similar to:

```
Your connection has been terminated and attempts to complete the failover
process to the following Alternate Servers have failed: AlternateServer:
'HOSTNAME=Server4:PORTNUMBER= 1521:SERVICE=test'. All active transactions
have been rolled back.
```

## Guidelines for Primary and Alternate Servers

Many databases provide advanced database replication technologies such as DB2 High Availability Disaster Recovery (HADR) and Oracle Real Application Clusters (RAC). The failover functionality provided by the drivers does not require any of these technologies, but can work with them to provide comprehensive failover protection. Use the following guidelines for primary and alternate servers to ensure that failover works correctly in your environment:

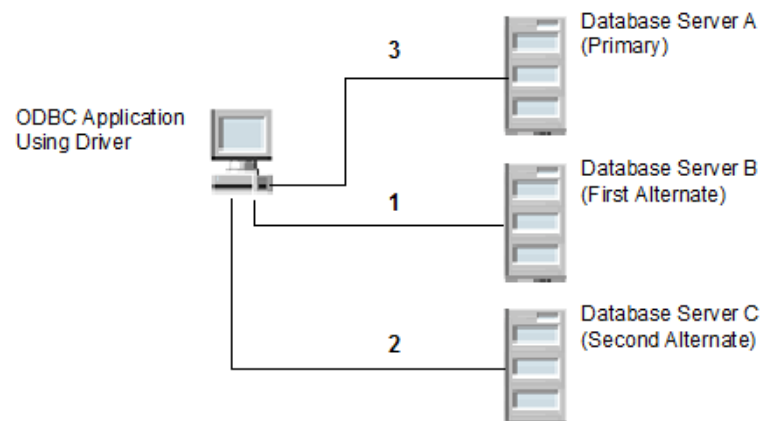
- Alternate servers should mirror data on the primary server or be part of a configuration where multiple database nodes share the same physical data.
- If using failover with DB2 HADR, the primary server must be the primary server configured in your HADR system and any alternate server must be a standby server configured in your HADR system.

## Using Client Load Balancing

Client load balancing is available in the following DataDirect Connect Series *for* ODBC drivers:

- DB2 Wire Protocol
- Greenplum Wire Protocol
- Informix Wire Protocol
- MySQL Wire Protocol
- Oracle Wire Protocol
- Oracle
- PostgreSQL Wire Protocol
- Progress OpenEdge Wire Protocol
- SQL Server Wire Protocol
- SQL Server Legacy Wire Protocol (UNIX only)
- Sybase Wire Protocol
- Sybase IQ Wire Protocol

Client load balancing helps distribute new connections in your environment so that no one server is overwhelmed with connection requests. When client load balancing is enabled, the order in which primary and alternate database servers are tried is random. For example, suppose that client load balancing is enabled as shown in the following illustration:



First, Database Server B is tried (1). Then, Database Server C may be tried (2), followed by a connection attempt to Database Server A (3). In contrast, if client load balancing were not enabled in this scenario, each database server would be tried in sequential order, primary server first, then each alternate server based on its entry order in the alternate servers list.

Client load balancing is controlled by the [Load Balancing](#) connection option. For details on configuring client load balancing, see the appropriate driver chapter in this book.

## Using Connection Retry

Connection retry is available in the following DataDirect Connect Series *for* ODBC drivers:

- DB2 Wire Protocol
- Greenplum Wire Protocol
- Informix Wire Protocol
- MySQL Wire Protocol
- Oracle Wire Protocol
- Oracle
- PostgreSQL Wire Protocol
- Progress OpenEdge Wire Protocol
- SQL Server Wire Protocol
- SQL Server Legacy Wire Protocol (UNIX only)
- Sybase Wire Protocol
- Sybase IQ Wire Protocol

Connection retry defines the number of times the driver attempts to connect to the primary server and, if configured, alternate database servers after the initial unsuccessful connection attempt. It can be used with connection failover, extended connection failover, and select failover. Connection retry can be an important strategy for system recovery. For example, suppose you have a power failure in which both the client and the server fails. When the power is restored and all computers are restarted, the client may be ready to attempt a connection before the server has completed its startup routines. If connection retry is enabled, the client application can continue to retry the connection until a connection is successfully accepted by the server.

Connection retry can be used in environments that have only one server or can be used as a complementary feature with connection failover in environments with multiple servers.

Using the connection options [Connection Retry Count](#) and [Connection Retry Delay](#), you can specify the number of times the driver attempts to connect and the time in seconds between connection attempts. For details on configuring connection retry, see the appropriate driver chapter in this book.

## Summary of Failover-Related Options

[Table 3-1](#) summarizes how failover-related connection options work with the drivers. See "Connection Option Descriptions" in each driver chapter for details about configuring the options. Not all options are available in every failover-enabled driver. The step numbers in the table refer the procedure that follows the table

**Table 3-1. Summary: Failover and Related Connection Options**

Option	Characteristic
Alternate Servers (see <a href="#">Step 1</a> )	One or multiple alternate database servers. An IP address or server name identifying each server is required.
Connection Retry Count (see <a href="#">Step 5</a> )	Number of times the driver retries the primary database server, and if specified, alternate servers until a successful connection is established.



**Table 3-1. Summary: Failover and Related Connection Options** (cont.)

Option	Characteristic
Connection Retry Delay (see <a href="#">Step 6</a> )	Wait interval, in seconds, between connection retry attempts when the Connection Retry Count option is set to a positive integer.
Failover Granularity (see <a href="#">Step 3</a> )	The type of behavior that the driver exhibits when errors are detected during the failover process.
Failover Mode (see <a href="#">Step 2</a> )	The type of failover that the driver attempts.
Failover Preconnect (see <a href="#">Step 4</a> )	Determines whether the driver makes a connection attempt to the next server in the Alternate Servers list at the time of the initial connection.
Load Balancing (see <a href="#">Step 7</a> )	Determines whether the driver uses client load balancing in its attempts to connect to primary and alternate database servers. If enabled, the driver attempts to connect to the database servers in random order.

- 1 To configure connection failover, you **must** specify one or more alternate database servers that are tried at connection time if the primary server is not accepting connections. To do this, use the Alternate Servers connection option. Connection attempts continue until a connection is successfully established or until all the database servers in the list have been tried once (the default).
- 2 Choose a failover method by setting the Failover Mode connection option. The default method is Connection (FailoverMode=0).
- 3 If Failover Mode is Extended Connection (FailoverMode=1) or Select (FailoverMode=2), set the Failover Granularity connection option to specify how you want the driver to behave if errors occur while trying to reestablish a lost connection. The default behavior of the driver is Non-Atomic (FailoverGranularity=0), which continues with the failover process and posts any errors on the statement on which they occur. Other values are:
 

Atomic (FailoverGranularity=1): the driver fails the entire failover process if an error is generated as the result of anything other than executing and repositioning a Select statement. If an error is generated as a result of repositioning a result set to the last row position, the driver continues with the failover process, but generates a warning that the Select statement must be reissued.

Atomic including Repositioning (FailoverGranularity=2): the driver fails the entire failover process if any error is generated as the result of restoring the state of the connection or the state of work in progress.

Disable Integrity Check (FailoverGranularity=3: the driver does not verify that the rows restored during the failover process match the original rows. This value applies only when Failover Mode is set to Select (FailoverMode=2).
- 4 Optionally, enable the Failover Preconnect connection option (FailoverPreconnect=1) if you want the driver to establish a connection with the primary and an alternate server at the same time. This value applies only when Failover Mode is set to Extended Connection (FailoverMode=1) or Select (FailoverMode=2). The default behavior is to

connect to an alternate server only when failover is caused by an unsuccessful connection attempt or a lost connection (FailoverPreconnect=0).

- 5 Optionally, specify the number of times the driver attempts to connect to the primary and alternate database servers after the initial unsuccessful connection attempt. By default, the driver does not retry. To set this feature, use the Connection Retry Count connection option.
- 6 Optionally, specify the wait interval, in seconds, between attempts to connect to the primary and alternate database servers. The default interval is 3 seconds. To set this feature, use the Connection Retry Delay connection option.
- 7 Optionally, specify whether the driver will use client load balancing in its attempts to connect to primary and alternate database servers. If load balancing is enabled, the driver uses a random pattern instead of a sequential pattern in its attempts to connect. The default value is not to use load balancing. To set this feature, use the Load Balancing connection option.

## A Connection String Example

The following connection string configures the Oracle Wire Protocol driver to use connection failover in conjunction with some of its optional features.

```
DSN=AcctOracleServer;AlternateServers=(HostName=AccountingOracleServer:PortNumber=1521:
SID=Accounting, HostName=255.201.11.24:PortNumber=1522:ServiceName=ABackup.NA.MyCompany) ;
ConnectionRetryCount=4;ConnectionRetryDelay=5;LoadBalancing=1;FailoverMode=0
```

Specifically, this connection string configures the driver to use two alternate servers as connection failover servers, to attempt to connect four additional times if the initial attempt fails, to wait five seconds between attempts, to try the primary and alternate servers in a random order, and to attempt reconnecting on new connections only. The additional connection information required for the alternate servers is specified in the data source AcctOracleServer.

## An odbc.ini File Example

To configure the 32-bit Oracle Wire Protocol driver to use connection failover in conjunction with some of its optional features in your odbc.ini file, you could set the following connection string attributes:

```
Driver=ODBCHOME/lib/ivoraxx.so
Description=DataDirect Oracle Wire Protocol driver
...
AlternateServers=(HostName=AccountingOracleServer:PortNumber=1521:SID=Accounting,
HostName=255.201.11.24:PortNumber=1522:ServiceName=ABackup.NA.MyCompany)
...
ConnectionRetryCount=4
ConnectionRetryDelay=5
...
LoadBalancing=0
...
FailoverMode=1
...
FailoverPreconnect=1
...
```

Specifically, this `odbc.ini` configuration tells the driver to use two alternate servers as connection failover servers, to attempt to connect four additional times if the initial attempt fails, to wait five seconds between attempts, to try the primary and alternate servers in sequential order (do not use load balancing), to attempt reconnecting on new and lost connections, and to establish a connection with the primary and alternate servers at the same time.

---

## Using Client Information

Many databases allow applications to store client information associated with a connection. For example, the following types of information can be useful for database administration and monitoring purposes:

- Name of the application currently using the connection.
- User ID for whom the application using the connection is performing work. The user ID may be different than the user ID that was used to establish the connection.
- Host name of the client on which the application using the connection is running.
- Product name and version of the driver on the client.
- Additional information that may be used for accounting or troubleshooting purposes, such as an accounting ID.

Client information is available in the following DataDirect Connect Series *for* ODBC drivers:

- DB2 Wire Protocol
- Oracle Wire Protocol

For DB2 V9.5 and V9.7 for Linux/UNIX/Windows and DB2 for z/OS, this information can feed directly into the Workload Manager (WLM) for workload management and monitoring purposes.

For Oracle 11g R2, this information is managed through the client information feature.

See ["Storing Client Information" on page 76](#) for more information about how DB2 and Oracle store client information.

## How Databases Store Client Information

Typically, databases that support storing client information do so by providing a register, a variable, or a column in a system table in which the information is stored. If an application attempts to store information and the database does not provide a mechanism for storing that information, the driver caches the information locally. Similarly, if an application returns client information and the database does not provide a mechanism for storing that information, the driver returns the locally cached value.

## Storing Client Information

Your application can store client information associated with a connection. [Table 3-2](#) shows the driver connection options that your application can use to store client information and where that client information is stored for each database. See the specific driver chapters for a description of each option.

**Table 3-2. Database Locations for Storing Client Information**

Option	Description	Database	Location
Accounting Info	Additional information that may be used for accounting or troubleshooting purposes, such as an accounting ID	DB2	CURRENT CLIENT_ACCTNG register (DB2 for Linux/UNIX/Windows) or CLIENT ACCTNG register (DB2 for z/OS).
		Oracle	CLIENT_INFO value in the V\$SESSION table.
Action	The current action within the current module.	Oracle	ACTION value in the V\$SESSION table.
Application Name	Name of the application currently using the connection	DB2	CURRENT CLIENT_APPLNAME register (DB2 for Linux/UNIX/Windows) or CLIENT APPLNAME register (DB2 for z/OS). For DB2 V9.1 and higher for Linux/UNIX/Windows, this value is also stored in the APPL_NAME value in the SYSIBMADM.APPLICATIONS table.
		Oracle	CLIENT_IDENTIFIER attribute. In addition, this value is also stored in the PROGRAM value in the V\$SESSION table.
Client Host Name	Host name of the client on which the application using the connection is running	DB2	CURRENT CLIENT_WRKSTNNAME register (DB2 for Linux/UNIX/Windows) or CLIENT WRKSTNNAME register (DB2 for z/OS).
		Oracle	MACHINE value in the V\$SESSION table.
Client ID	Additional information about the client	Oracle	CLIENT_IDENTIFIER value in the V\$SESSION table.
Client User	User ID for whom the application using the connection is performing work	DB2	CURRENT CLIENT_USERID register (DB2 for Linux/UNIX/Windows) or CLIENT USERID register (DB2 for z/OS).
		Oracle	OSUSER value in the V\$SESSION table.
Module	The name of a stored procedure or the name of the application	Oracle	MODULE value in the V\$SESSION table.
Program ID	Product name and version of the driver on the client	DB2	CLIENT_PRDID value. For DB2 V9.1 and higher for Linux/UNIX/Windows, the CLIENT_PRDID value is located in the SYSIBMADM.APPLICATIONS table.
		Oracle	PROCESS value in the V\$SESSION table.

## Using Security

The drivers support authentication and data encryption. For current information, refer to the security matrix on the Progress DataDirect Web site:

<http://www.datadirect.com/products/security/documentation/securitymatrix.htm>

The individual driver chapters provide driver-specific details, but the following sections give an overview of both authentication and data encryption, as well as discussing general requirements.

### Authentication

On most computer systems, a password is used to prove a user's identity. This password often is transmitted over the network and can possibly be intercepted by malicious hackers. Because this password is the one secret piece of information that identifies a user, anyone knowing a user's password can effectively be that user. Authentication methods protect the identity of the user.

The drivers support the following authentication methods:

- *User ID/password authentication* authenticates the user to the database using a database user name and password.
- *Client authentication* uses the user ID and password of the user logged onto the system on which the driver is running to authenticate the user to the database. The database server relies on the client to authenticate the user and does not provide additional authentication.
- *Kerberos authentication* is a trusted third-party authentication service that verifies user identities. DataDirect Connect Series *for* ODBC supports both Windows Active Directory Kerberos and MIT Kerberos implementations.
- *NTLM authentication* authenticates clients to the database through a challenge-response authentication mechanism that enables clients to prove their identities without sending a database password to the server.

### Kerberos Authentication

Kerberos authentication is available in the following DataDirect Connect Series *for* ODBC drivers:

- DB2 Wire Protocol
- Oracle Wire Protocol
- SQL Server Wire Protocol
- Sybase Wire Protocol
- Driver for the Teradata Database

Kerberos authentication can take advantage of the user name and password maintained by the operating system to authenticate users to the database or use another set of user credentials specified by the application.

The Kerberos method requires knowledge of how to configure your Kerberos environment. This method supports both Windows Active Directory Kerberos and MIT Kerberos environments.

To use Kerberos authentication, the application user first must obtain a Kerberos Ticket Granting Ticket (TGT) from the Kerberos server. The Kerberos server verifies the identity of the user and controls access to services using the credentials contained in the TGT.

If the application uses Kerberos authentication from a Windows client, the application user does not explicitly need to obtain a TGT. Windows Active Directory automatically obtains a TGT for the user.

If the application uses Kerberos authentication from a UNIX or Linux client, the user must explicitly obtain a TGT. To obtain a TGT explicitly, the user must log onto the Kerberos server using the kinit command. For example, the following command requests a TGT from the server with a lifetime of 10 hours, which is renewable for 5 days:

```
kinit -l 10h -r 5d user
```

where *user* is the application user.

Refer to your Kerberos documentation for more information about using the kinit command and obtaining TGTs for users.

NTLM Authentication

NTLM authentication is available in the following the DataDirect Connect Series for ODBC drivers:

- SQL Server Wire Protocol
- Driver for the Teradata database

Table 3-3 provides the platform support information for the drivers.

Table 3-3. Driver Support for NTLM Authentication		
Driver	Windows	Linux/UNIX
SQL Server Wire Protocol	X	X <sup>1</sup>
Driver for the Teradata database	X	
1. NTLM single sign on is not supported. To connect to SQL Server, users must use connection attributes to supply the Windows User Id, Password, and Domain to the driver.		

Data Encryption Across the Network

If your database connection is not configured to use data encryption, data is sent across the network in a format that is designed for fast transmission and can be decoded by interceptors, given some time and effort. For example, text data is often sent across the wire as clear text. Because this format does not provide complete protection from interceptors, you may want to use data encryption to provide a more secure transmission of data.

For example, you may want to use data encryption in the following scenarios:

- You have offices that share confidential information over an intranet.
- You send sensitive data, such as credit card numbers, over a database connection.
- You need to comply with government or industry privacy and security requirements.

Certain DataDirect Connect Series *for* ODBC drivers support Secure Sockets Layer (SSL). SSL is an industry-standard protocol for sending encrypted data over database connections. SSL secures the integrity of your data by encrypting information and providing client/server authentication. In addition, the DataDirect Connect Series *for* ODBC DB2 Wire Protocol driver supports DB2 database-specific encryption.

NOTE: Data encryption may adversely affect performance because of the additional overhead (mainly CPU usage) required to encrypt and decrypt data.

## SSL Encryption

SSL encryption is available in the following DataDirect Connect Series *for* ODBC drivers:

- DB2 Wire Protocol
- MySQL Wire Protocol
- Oracle Wire Protocol
- PostgreSQL Wire Protocol
- Progress OpenEdge Wire Protocol
- Salesforce
- SQL Server Wire Protocol
- Sybase Wire Protocol

NOTE: Communication between the Salesforce driver and Salesforce.com, Force.com, and Database.com is always SSL encrypted.

SSL works by allowing the client and server to send each other encrypted data that only they can decrypt. SSL negotiates the terms of the encryption in a sequence of events known as the *SSL handshake*. The drivers support the SSL v2, SSL v3, and TLS v1 protocols using OpenSSL cipher suites, and negotiate the highest SSL/TLS protocol available during the handshake. The result of this negotiation determines the encryption cipher suite to be used for the SSL session.

The encryption cipher suite defines the type of encryption that is used for any data exchanged through an SSL connection. Refer to [Chapter 8 “SSL Encryption Cipher Suites”](#) in the *DataDirect Connect Series for ODBC Reference* for a list of the encryption cipher suites supported by the drivers.

The handshake involves the following types of authentication:

- *SSL server authentication* requires the server to authenticate itself to the client.
- *SSL client authentication* is optional and requires the client to authenticate itself to the server after the server has authenticated itself to the client. Not all databases support SSL client authentication.

## Certificates

SSL requires the use of a digitally-signed document, an x.509 standard certificate, for authentication and the secure exchange of data. The purpose of this certificate is to tie the public key contained in the certificate securely to the person/company that holds the corresponding private key. The DataDirect Connect Series *for* ODBC drivers support many popular formats. Supported formats include:

- DER Encoded Binary X.509
- Base64 Encoded X.509
- PKCS #12 / Personal Information Exchange

## SSL Server Authentication

When the client makes a connection request, the server presents its public certificate for the client to accept or deny. The client checks the issuer of the certificate against a list of trusted Certificate Authorities (CAs) that resides in an encrypted file on the client known as a *truststore*. If the certificate matches a trusted CA in the truststore, an encrypted connection is established between the client and server. If the certificate does not match, the connection fails and the driver generates an error.

Most truststores are password-protected. The driver must be able to locate the truststore and unlock the truststore with the appropriate password. Two connection string attributes are available to the driver to provide this information: *TrustStore* and *TrustStorePassword*. The value of *TrustStore* is a pathname that specifies the location of the truststore file. The value of *TrustStorePassword* is the password required to access the contents of the truststore.

Alternatively, you can configure the driver to trust any certificate sent by the server, even if the issuer is not a trusted CA. Allowing a driver to trust any certificate sent from the server is useful in test environments because it eliminates the need to specify truststore information on each client in the test environment. *ValidateServerCertificate*, another connection string attribute, allows the driver to accept any certificate returned from the server regardless of whether the issuer of the certificate is a trusted CA.

Finally, the connection string attribute, *HostNameInCertificate*, allows an additional method of server verification. When a value is specified for *HostNameInCertificate*, it must match the host name of the server, which has been established by the SSL administrator. This prevents malicious intervention between the client and the server and ensures that the driver is connecting to the server that was requested.

## SSL Client Authentication

If the server is configured for SSL client authentication, the server asks the client to verify its identity after the server identity has been proven. Similar to server authentication, the client sends a public certificate to the server to accept or deny. The client stores its public certificate in an encrypted file known as a *keystore*. Public certificates are paired with a private key in the keystore. To send the public certificate, the driver must access the private key.

Like the truststore, most keystores are password-protected. The driver must be able to locate the keystore and unlock the keystore with the appropriate password. Two connection string attributes are available to the driver to provide this information: *KeyStore* and



KeyStorePassword. The value of KeyStore is a pathname that specifies the location of the keystore file. The value of KeystorePassword is the password required to access the keystore.

The private keys stored in a keystore can be individually password-protected. In many cases, the same password is used for access to both the keystore and to the individual keys in the keystore. It is possible, however, that the individual keys are protected by passwords different from the keystore password. The driver needs to know the password for an individual key to be able to retrieve it from the keystore. An additional connection string attribute, KeyPassword, allows you to specify a password for an individual key.

Not all databases support SSL client authentication. The individual driver chapters indicate whether client authentication is supported.

## Summary of Security-Related Options

[Table 3-4](#) summarizes how security-related connection options work with the drivers. See "Connection Option Descriptions" in each driver chapter for details about configuring the options.

---

**Table 3-4. Summary: Security Connection Options**

---

Option	Characteristic
Authentication Method	The method the driver uses to authenticate the user to the server when a connection is established.
Encryption Method	The method the driver uses to encrypt data sent between the driver and the database server.
GSS Client Library	The name of the Generic Security Service (GSS) client library that the driver uses to communicate with the Key Distribution Center (KDC).
Host Name In Certificate	The host name established by the SSL administrator for the driver to validate the host name contained in the certificate.
Key Password	The password required to access an individual key in the keystore.
Keystore	The path that specifies the location of the keystore file.
Keystore Password	The password required to access the keystore.
Truststore	The path that specifies the location of the truststore file.
Truststore Password	The password required to access the truststore.
User Name	The default user ID used to connect to your database.
Validate Server Certificate	Validates the security certificate of the server as part of the SSL authentication handshake.

---

---

## Using DataDirect Connection Pooling

DataDirect connection pooling is available in the following DataDirect Connect Series *for* ODBC drivers:

- DB2 Wire Protocol
- Greenplum Wire Protocol
- MySQL Wire Protocol
- Oracle Wire Protocol
- Oracle
- PostgreSQL Wire Protocol
- SQL Server Wire Protocol
- Sybase Wire Protocol
- Sybase IQ Wire Protocol

Connection pooling allows you to *reuse* connections rather than creating a new one every time the driver needs to establish a connection to the underlying database. The DataDirect Connect Series *for* ODBC drivers enable connection pooling without requiring changes to your client application.

NOTE: Connection pooling works only with connections that are established using SQLConnect or SQLDriverConnect with the SQL\_DRIVER\_NO\_PROMPT argument and only with applications that are thread-enabled.

DataDirect connection pooling that is implemented by the DataDirect driver is different than connection pooling implemented by the Windows Driver Manager. The Windows Driver Manager opens connections dynamically, up to the limits of memory and server resources. DataDirect connection pooling, however, allows you to control the number of connections in a pool through the Min Pool Size (minimum number of connections in a pool) and Max Pool Size (maximum number of connections in a pool) connection options. In addition, DataDirect connection pooling is cross-platform, allowing it to operate on UNIX and Linux. See the "Connection Option Descriptions" section in each driver's chapter for details about how the connection options manage DataDirect connection pooling.

IMPORTANT: On a Windows system, do not use both Windows Driver Manager connection pooling and DataDirect connection pooling at the same time.

### Creating a Connection Pool

Each connection pool is associated with a specific connection string. By default, the connection pool is created when the first connection with a unique connection string connects to the data source. The pool is populated with connections up to the minimum pool size before the first connection is returned. Additional connections can be added until the pool reaches the maximum pool size. If the Max Pool Size option is set to 10 and all connections are active, a request for an eleventh connection has to wait in queue for one of the 10 pool connections to become idle. The pool remains active until the process ends or the driver is unloaded.

If a new connection is opened and the connection string does not exactly match an existing pool, a new pool must be created. By using the same connection string, you can enhance the performance and scalability of your application.

## Adding Connections to a Pool

A connection pool is created in the process of creating each unique connection string that an application uses. When a pool is created, it is populated with enough connections to satisfy the minimum pool size requirement, set by the Min Pool Size connection option. The maximum pool size is set by the Max Pool Size connection option. The driver allocates additional connections to the pool until the number of connections reaches the value set by Max Pool Size.

Once the maximum pool size has been reached and no usable connection is available to satisfy a connection request, the request is queued in the driver. The driver waits for the length of time specified in the Login Timeout connection option for a usable connection to return to the application. If this time period expires and a connection has not become available, the driver returns an error to the application.

A connection is returned to the pool when the application calls SQLDisconnect. Your application is still responsible for freeing the handle, but this does not result in the database session ending.

## Removing Connections from a Pool

A connection is removed from a connection pool when it exceeds its lifetime as determined by the Load Balance Timeout connection option. In addition, DataDirect has created connection attributes described in [Table 3-5](#) to give your application the ability to reset connection pools. If connections are in use at the time of these calls, they are marked appropriately. When SQLDisconnect is called, the connections are discarded instead of being returned to the pool.

Table 3-5. Pool Reset Connection Attributes

Connection Attribute	Description
<b>SQL_ATTR_CLEAR_POOLS</b> Value: SQL_CLEAR_ALL_CONN_POOL	Calling SQLSetConnectAttr (SQL_ATTR_CLEAR_POOLS, SQL_CLEAR_ALL_CONN_POOL) clears all the connection pools associated with the driver that created the connection.  This is a write-only connection attribute. The driver returns an error if SQLGetConnectAttr (SQL_ATTR_CLEAR_POOLS) is called.
<b>SQL_ATTR_CLEAR_POOLS</b> Value: SQL_CLEAR_CURRENT_CONN_POOL	Calling SQLSetConnectAttr (SQL_ATTR_CLEAR_POOLS, SQL_CLEAR_CURRENT_CONN_POOL) clears the connection pool that is associated with the current connection.  This is a write-only connection attribute. The driver returns an error if SQLGetConnectAttr (SQL_ATTR_CLEAR_POOLS) is called.

NOTE: By default, if removing a connection causes the number of connections to drop below the number specified in the Min Pool Size option, a new connection is not created until an application needs one.

## Handling Dead Connections in a Pool

What happens when an idle connection loses its physical connection to the database? For example, suppose the database server is rebooted or the network experiences a temporary interruption. An application that attempts to connect could receive errors because the physical connection to the database has been lost.

DataDirect Connect Series *for* ODBC drivers handle this situation transparently to the user. The application does not receive any errors on the connection attempt because the driver simply returns a connection from a connection pool. The first time the connection handle is used to execute a SQL statement, the driver detects that the physical connection to the server has been lost and attempts to reconnect to the server *before* executing the SQL statement. If the driver can reconnect to the server, the result of the SQL execution is returned to the application; no errors are returned to the application.

The driver uses connection failover option values, if they are enabled, when attempting this seamless reconnection; however, it attempts to reconnect even if these options are not enabled. See ["Connection Failover" on page 66](#) for information about configuring the driver to connect to a backup server when the primary server is not available.

NOTE: If the driver cannot reconnect to the server (for example, because the server is still down), an error is returned indicating that the reconnect attempt failed, along with specifics about the reason the connection failed.

The technique that Progress DataDirect uses for handling dead connections in connection pools allows for maximum performance of the connection pooling mechanism. Some drivers periodically test the server with a dummy SQL statement while the connections sit idle. Other drivers test the server when the application requests the use of the connection from the connection pool. Both of these approaches add round trips to the database server and ultimately slow down the application during normal operation.

## Connection Pool Statistics

Progress DataDirect has created a connection attribute to monitor the status of the DataDirect Connect Series *for* ODBC connection pools. This attribute, which is described in [Table 3-6](#), allows your application to fetch statistics for the pool to which a connection belongs.

Table 3-6. Pool Statistics Connection Attribute	
Connection Attribute	Description
SQL_ATTR_POOL_INFO	Calling SQLGetConnectAttr (SQL_ATTR_POOL_INF, SQL_GET_POOL_INFO) returns a PoolInfoStruct that contains the statistics for the connection pool to which this connection belongs. This PoolInfoStruct is defined in qesqlxt.h. For example:
Value: SQL_GET_POOL_INFO	<pre>SQLGetConnectAttr(hdbc, SQL_ATTR_POOL_INFO, PoolInfoStruct *, SQL_LEN_BINARY_ATTR(PoolInfoStruct), &amp;len);</pre> <p>This is a read-only connection attribute. The driver returns an error if SQLSetConnectAttr (SQL_ATTR_POOL_INFO) is called.</p>

## Summary of Pooling-Related Options

[Table 3-7](#) summarizes how connection pooling-related connection options work with the drivers. See "Connection Option Descriptions" in each driver chapter for details about configuring the options.

**Table 3-7. Summary: Connection Pooling Connection Options**

Option	Characteristic
Connection Pooling	Enables connection pooling.
Connection Reset	Resets a connection that is removed from the connection pool to the initial configuration settings of the connection.
Load Balance Timeout	An integer value to specify the amount of time, in seconds, to keep connections open in a connection pool.
Max Pool Size	An integer value to specify the maximum number of connections within a single pool.
Min Pool Size	An integer value to specify the minimum number of connections that are opened and placed in a connection pool when it is created.

## Using DataDirect Bulk Load

The drivers support DataDirect Bulk Load, a feature that allows your application to send large numbers of rows of data to a database or Salesforce instance.

[Table 3-8](#) and [Table 3-9](#) describe the bulk load behavior for the drivers.

**Table 3-8. Bulk Load Behavior for DataDirect Connect for ODBC**

Driver	Bulk Load Behavior
Oracle <sup>1</sup> Microsoft SQL Server <sup>2</sup> Sybase	The driver sends the data to the database in a continuous stream instead of numerous smaller database packets. Similar to batch operations, using bulk load improves performance because far fewer network round trips are required. Bulk load bypasses the data parsing usually done by the database, providing an additional performance gain over batch operations.
DB2	Because DB2 does not have native bulk load support, the driver supports bulk through the native parameter array mechanism.

1. Supports bulk load for Oracle9i R2 and higher.

2. Supports bulk load for Microsoft SQL Server 2000 and higher.

**Table 3-9. Bulk Load Behavior for DataDirect Connect XE for ODBC**

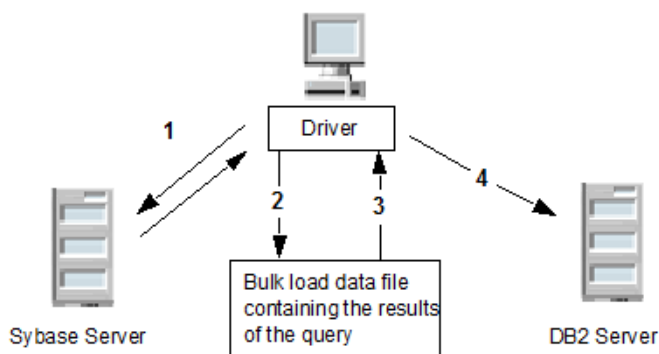
Driver	Bulk Load Behavior
Salesforce	The driver sends data to a Salesforce instance using the Salesforce Bulk API instead of the Web Service API. Using the Bulk API significantly reduces the number of Web service calls the driver uses to transfer data and may improve performance.

**IMPORTANT:**

- DataDirect Bulk Load requires a licensed installation of the drivers. If the drivers are installed with an evaluation license, the bulk load feature is available for prototyping with your applications, but with limited scope. Contact your sales representative or Progress DataDirect SupportLink for further information.
- Because a bulk load operation may bypass data integrity checks, your application must ensure that the data it is transferring does not violate integrity constraints in the database. For example, suppose you are bulk loading data into a database table and some of that data duplicates data stored as a primary key, which must be unique. The driver will not throw an exception to alert you to the error; your application must provide its own data integrity checks.

Bulk load operations are accomplished by exporting the results of a query from a database into a comma-separated value (CSV) file, a bulk load data file. The driver then loads the data from bulk load data file into a different database. The file can be used by any DataDirect Connect Series *for* ODBC drivers. In addition, the bulk load data file is supported by other DataDirect Connect product lines that feature bulk loading, for example, a DataDirect Connect *for* ADO.NET data provider that supports bulk load.

Suppose that you had customer data on a Sybase server and need to export it to a DB2 server. The driver would perform the following steps:



- 1 Application using Sybase Wire Protocol driver sends query to and receives results from Sybase server.
- 2 Driver exports results to bulk load data file.

- 3 Driver retrieves results from bulk load data file.
- 4 Driver bulk loads results on DB2 server.

## Bulk Export and Load Methods

You can take advantage of DataDirect Bulk Load either through the Driver setup dialog or programmatically.

Applications that are already coded to use parameter array batch functionality can leverage DataDirect Bulk Load features through the Enable Bulk Load connection option on the Bulk tab of the Driver setup dialog. Enabling this option automatically converts the parameter array batch operation to use the database bulk load protocol without any code changes to your application.

If you are not using parameter array batch functionality, the bulk operation buttons **Export Table** and **Load Table** on the Bulk tab of the driver Setup dialog also allow you to use bulk load functionality without any code changes. See the individual driver chapters for a description of the Bulk tab.

If you want to integrate bulk load functionality seamlessly into your application, you can include code to use the bulk load functions exposed by the driver.

NOTE: For your applications to use DataDirect Bulk Load functionality, they must obtain driver connection handles and function pointers, as follows:

- 1 Use SQLGetInfo with the parameter SQL\_DRIVER\_HDBC to obtain the driver's connection handle from the Driver Manager.
- 2 Use SQLGetInfo with the parameter SQL\_DRIVER\_HLIB to obtain the driver's shared library or DLL handle from the Driver Manager.
- 3 Obtain function pointers to the bulk load functions using the function name resolution method specific to your operating system. The bulk.c example program shipped with the drivers contains the function resolveName that illustrates how to obtain function pointers to the bulk load functions.

This is detailed in the code samples that follow and in ["DataDirect Bulk Load Functions"](#) in [Chapter 9](#) of the *DataDirect Connect Series for ODBC Reference*.

## Exporting Data from a Database

You can export data from a database in one of three ways:

- From a table by using the driver Setup dialog
- From a table by using DataDirect functions
- From a result set by using DataDirect statement attributes

From the DataDirect driver Setup dialog, select the Bulk tab and click **Export Table**. See the individual driver chapters for a description of this procedure.

Your application can export a table using the DataDirect functions `ExportTableToFile` (ANSI application) or `ExportTableToFileW` (Unicode application). The application must first obtain driver connection handles and function pointers, as shown in the following example:

```
HDBC      hdbc;
HENV      henv;
void      *driverHandle;
HMODULE    hmod;
PEXportTableToFile exportTableToFile;

char      tableName[128];
char      fileName[512];
char      logFile[512];
int        errorTolerance;
int        warningTolerance;
int        codePage;

/* Get the driver's connection handle from the DM.
   This handle must be used when calling directly into the driver. */

rc = SQLGetInfo (hdbc, SQL_DRIVER_HDBC, &driverHandle, 0, NULL);
if (rc != SQL_SUCCESS) {
    ODBC_error (henv, hdbc, SQL_NULL_HSTMT);
    EnvClose (henv, hdbc);
    exit (255);
}

/* Get the DM's shared library or DLL handle to the driver. */

rc = SQLGetInfo (hdbc, SQL_DRIVER_HLIB, &hmod, 0, NULL);
if (rc != SQL_SUCCESS) {
    ODBC_error (henv, hdbc, SQL_NULL_HSTMT);
    EnvClose (henv, hdbc);
    exit (255);
}

exportTableToFile = (PEXportTableToFile)
    resolveName (hmod, "ExportTableToFile");
if (! exportTableToFile) {
    printf ("Cannot find ExportTableToFile!\n");
    exit (255);
}

rc = (*exportTableToFile) (
    driverHandle,
    (const SQLCHAR *) tableName,
    (const SQLCHAR *) fileName,
    codePage,
    errorTolerance, warningTolerance,
    (const SQLCHAR *) logFile);
if (rc == SQL_SUCCESS) {
    printf ("Export succeeded.\n");
}
```



```
else {
    driverError (driverHandle, hmod);
}
```

Refer to ["DataDirect Bulk Load Functions"](#) in [Chapter 9](#) of the *DataDirect Connect Series for ODBC Reference* for a full description of these functions.

Your application can export a result set using the DataDirect statement attributes SQL\_BULK\_EXPORT and SQL\_BULK\_EXPORT\_PARAMS.

The export operation creates a bulk load data file with a .csv extension in which the exported data is stored. For example, assume that an Oracle source table named GBMAXTABLE contains four columns. The resulting bulk load data file GBMAXTABLE.csv containing the results of a query would be similar to the following:

```
1,0x6263,"bc","bc"
2,0x636465,"cde","cde"
3,0x64656667,"defg","defg"
4,0x6566676869,"efghi","efghi"
5,0x666768696a6b,"fghijk","fghijk"
6,0x6768696a6b6c6d,"ghijklm","ghijklm"
7,0x68696a6b6c6d6e6f,"hijklmno","hijklmno"
8,0x696a6b6c6d6e6f7071,"ijklmnopq","ijklmnopq"
9,0x6a6b6c6d6e6f70717273,"jklmnopqrs","jklmnopqrs"
10,0x6b,"k","k"
```

A bulk load configuration file with and .xml extension is also created when either a table or a result set is exported to a bulk load data file. See ["The Bulk Load Configuration File"](#) on [page 91](#) for an example of a bulk load configuration file.

In addition, a log file of events as well as external overflow files can be created during a bulk export operation. The log file is configured through either the driver Setup dialog Bulk tab, the ExportTableToFile function, or the SQL\_BULK\_EXPORT statement attribute. The external overflow files are configured through connection options; see ["External Overflow Files"](#) on [page 95](#) for details.

## Bulk Loading to a Database

The Enable Bulk Load connection option specifies the method by which bulk data is loaded to a database. When the option is enabled, the driver uses database bulk load protocols. When not enabled, the driver uses standard parameter arrays.

You can load data from the bulk load data file into the target database through the DataDirect driver Setup dialog by selecting the Bulk tab and clicking **Load Table**. See the individual driver chapters of the drivers that support bulk load for a description of this procedure.

Your application can also load data from the bulk load data file into the target database using the using the DataDirect functions LoadTableFromFile (ANSI application) or LoadTableFromFileW (Unicode application). The application must first obtain driver connection handles and function pointers, as shown in the following example:

```
HDBC      hdbc;
HENV      henv;
void      *driverHandle;
```

```

HMODULE    hmod;
PLoadTableFromFile loadTableFromFile;
char        tableName[128];
char        fileName[512];
char        configFile[512];
char        logFile[512];
char        discardFile[512];
int         errorTolerance;
int         warningTolerance;
int         loadStart;
int         loadCount;
int         readBufferSize;

/* Get the driver's connection handle from the DM.
   This handle must be used when calling directly into the driver. */

rc = SQLGetInfo (hdbc, SQL_DRIVER_HDBC, &driverHandle, 0, NULL);
if (rc != SQL_SUCCESS) {
    ODBC_error (henv, hdbc, SQL_NULL_HSTMT);
    EnvClose (henv, hdbc);
    exit (255);
}

/* Get the DM's shared library or DLL handle to the driver. */

rc = SQLGetInfo (hdbc, SQL_DRIVER_HLIB, &hmod, 0, NULL);
if (rc != SQL_SUCCESS) {
    ODBC_error (henv, hdbc, SQL_NULL_HSTMT);
    EnvClose (henv, hdbc);
    exit (255);
}

loadTableFromFile = (PLoadTableFromFile)
    resolveName (hmod, "LoadTableFromFile");
if (! loadTableFromFile) {
    printf ("Cannot find LoadTableFromFile!\n");
    exit (255);
}

rc = (*loadTableFromFile) (
    driverHandle,
    (const SQLCHAR *) tableName,
    (const SQLCHAR *) fileName,
    errorTolerance, warningTolerance,
    (const SQLCHAR *) configFile,
    (const SQLCHAR *) logFile,
    (const SQLCHAR *) discardFile,
    loadStart, loadCount,
    readBufferSize);
if (rc == SQL_SUCCESS) {
    printf ("Load succeeded.\n");
}

```

```
else {
    driverError (driverHandle, hmod);
}
```

Refer to ["DataDirect Bulk Load Functions"](#) in [Chapter 9](#) of the *DataDirect Connect Series for ODBC Reference* for a full description of these functions.

Use the BulkLoadBatchSize connection attribute to specify the number of rows the driver loads to the data source at a time when bulk loading data. Performance can be improved by increasing the number of rows the driver loads at a time because fewer network round trips are required. Be aware that increasing the number of rows that are loaded also causes the driver to consume more memory on the client.

A log file of events as well as a discard file that contains rows rejected during the load can be created during a bulk load operation. These files are configured through either the driver Setup dialog Bulk tab or the LoadTableFromFile function.

The discard file is in the same format as the bulk load data file. After fixing reported issues in the discard file, the bulk load can be reissued using the discard file as the bulk load data file.

**NOTE FOR SYBASE USERS:** Additional database configuration is required for destination tables that do not have an index. See the "Persisting a Result Set as an XML Data File" section in your driver chapter for more information.

**NOTE FOR SALESFORCE USERS:** In addition to bulk Insert, the Salesforce driver also supports bulk Delete, Update, and Upsert. This functionality is enabled with the SetBulkOperation function which is implemented in the driver. Refer to ["DataDirect Bulk Load Functions"](#) in [Chapter 9](#) of the *DataDirect Connect Series for ODBC Reference* for a full description of these functions.

## The Bulk Load Configuration File

A bulk load configuration file is created when either a table or a result set is exported to a bulk load data file. This file has the same name as the bulk load data file, but with an .xml extension.

The bulk load configuration file defines in its metadata the names and data types of the columns in the bulk load data file. The file defines these names and data types based on the table or result set created by the query that exported the data.

It also defines other data properties, such as length for character and binary data types, the character encoding code page for character types, precision and scale for numeric types, and nullability for all types.

When a bulk load data file cannot read its configuration file, the following defaults are assumed:

- All data is read in as character data. Each value between commas is read as character data.
- The default character set is defined, on Windows, by the current Windows code page. On UNIX/Linux, it is the IANAAppCodePage value, which defaults to 4.

For example, the format of the bulk load data file GBMAXTABLE.csv (discussed in ["Exporting Data from a Database" on page 87](#)) is defined by the bulk load configuration file, GBMAXTABLE.xml, as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<table codepage="UTF-16LE" xsi:noNamespaceSchemaLocation=
"http://www.datadirect.com/ns/bulk/BulkData.xsd" xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance">
  <row>
    <column datatype="DECIMAL" precision="38" scale="0" nullable=
      "false">INTEGERCOL</column>
    <column datatype="VARBINARY" length="10" nullable=
      "true">VARBINCOL</column>
    <column datatype="VARCHAR" length="10" sourcecodepage="Windows-1252"
      externalfilecodepage="Windows-1252" nullable="true">VCHARCOL</column>
    <column datatype="VARCHAR" length="10" sourcecodepage="Windows-1252"
      externalfilecodepage="Windows-1252" nullable="true">UNIVCHARCOL</column>
  </row>
</table>
```

## Bulk Load Configuration File Schema

The bulk load configuration file is supported by an underlying XML Schema defined at:

<http://www.datadirect.com/ns/bulk/BulkData.xsd>

The bulk load configuration file must conform to the bulk load configuration XML schema. Each bulk export operation generates a bulk load configuration file in UTF-8 format. If the bulk load data file cannot be created or does not comply with the XML Schema described in the bulk load configuration file, an error is generated.

## Verification of the Bulk Load Configuration File

You can verify the metadata in the configuration file against the data structure of the target database table. This insures that the data in the bulk load data file is compatible with the target database table structure.

The verification does not check the actual data in the bulk load data file, so it is possible that the load can fail even though the verification succeeds. For example, if you were to update the bulk load data file manually such that it has values that exceed the maximum column length of a character column in the target table, the load would fail.

Not all of the error messages or warnings that are generated by verification necessarily mean that the load will fail. Many of the messages simply notify you about possible incompatibilities between the source and target tables. For example, if the bulk load data file has a column that is defined as an integer and the column in the target table is defined as smallint, the load may still succeed if the values in the source column are small enough that they fit in a smallint column.

To verify the metadata in the bulk load configuration file through the DataDirect driver Setup dialog, select the Bulk tab and click **Verify**. See the individual driver chapters of the drivers that support bulk load for a description of this procedure.

Your application can also verify the metadata of the bulk load configuration file using the DataDirect functions `ValidateTableFromFile` (ANSI application) or `ValidateTableFromFileW` (Unicode application). The application must first obtain driver connection handles and function pointers, as shown in the following example:

```
HDBC      hdbc;
HENV      henv;
void      *driverHandle;
HMODULE    hmod;
PValidateTableFromFile validateTableFromFile;

char      tableName[128];
char      configFile[512];
char      messageList[10240];
SQLLEN    numMessages;

/* Get the driver's connection handle from the DM.
   This handle must be used when calling directly into the driver. */

rc = SQLGetInfo (hdbc, SQL_DRIVER_HDBC, &driverHandle, 0, NULL);
if (rc != SQL_SUCCESS) {
    ODBC_error (henv, hdbc, SQL_NULL_HSTMT);
    EnvClose (henv, hdbc);
    exit (255);
}

/* Get the DM's shared library or DLL handle to the driver. */

rc = SQLGetInfo (hdbc, SQL_DRIVER_HLIB, &hmod, 0, NULL);
if (rc != SQL_SUCCESS) {
    ODBC_error (henv, hdbc, SQL_NULL_HSTMT);
    EnvClose (henv, hdbc);
    exit (255);
}

validateTableFromFile = (PValidateTableFromFile)
    resolveName (hmod, "ValidateTableFromFile");
if (!validateTableFromFile) {
    printf ("Cannot find ValidateTableFromFile!\n");
    exit (255);
}

messageList[0] = 0;
numMessages = 0;

rc = (*validateTableFromFile) (
    driverHandle,
    (const SQLCHAR *) tableName,
    (const SQLCHAR *) configFile,
    (SQLCHAR *) messageList,
    sizeof (messageList),
    &numMessages);
```

```
printf ("%d message%s%s\n", numMessages,
        (numMessages == 0) ? "s" :
        ((numMessages == 1) ? " : " : "s : "),
        (numMessages > 0) ? messageList : "");
if (rc == SQL_SUCCESS) {
    printf ("Validate succeeded.\n");
}
else {
    driverError (driverHandle, hmod);
}
```

Refer to ["DataDirect Bulk Load Functions"](#) in [Chapter 9](#) of the *DataDirect Connect Series for ODBC Reference* for a complete description of these functions.

## Sample Applications

Progress DataDirect provides a sample application that demonstrates the bulk export, verification, and bulk load operations. This application is located in the \samples\bulk subdirectory of the product installation directory along with a text file named bulk.txt. Please consult bulk.txt for instructions on using the sample bulk load application.

A bulk streaming application is also provided in the \samples\bulkstrm subdirectory along with a text file named bulkstrm.txt. Please consult bulkstrm.txt for instructions on using the bulk streaming application.

## Determining the Bulk Load Protocol

Bulk operations can be performed using a dedicated bulk load protocol, that is, the protocol of the underlying database system, or by using parameter array batch operations. Dedicated protocols are generally more performance-efficient than parameter arrays. In some cases, however, you must use parameter arrays, for example, when the data to be loaded is in a data type not supported by the dedicated bulk protocol.

The Enable Bulk Load connection option determines bulk load behavior. When the option is enabled, the driver uses database bulk load protocols unless it encounters a problem, in which case it returns an error. In this situation, you must disable Enable Bulk Load so that the driver uses standard parameter arrays.

## Character Set Conversions

It is most performance-efficient to transfer data between databases that use the same character sets. At times, however, you might need to bulk load data between databases that use different character sets. You can do this by choosing a character set for the bulk load data file that will accommodate all data. If the source table contains character data that uses different character sets, then one of the Unicode character sets, UTF-8, UTF-16BE, or UTF-16LE must be specified for the bulk load data file. A Unicode character set should also be specified in the case of a target table uses a different character set than the source table to minimize conversion errors. If the source and target tables use the same character set, that set should be specified for the bulk load data file.

A character set is defined by a code page. The code page for the bulk load data file is defined in the configuration file and is specified through either the Code Page option of the Export Table driver Setup dialog or through the IANAAppCodePage parameter of the ExportTableToFile function. Any code page listed in [Chapter 1 "Code Page Values"](#) of the *DataDirect Connect Series for ODBC Reference* is supported for the bulk load data file.

Any character conversion errors are handled based on the value of the Report CodePage ConversionErrors connection option. See the individual driver chapters for a description of this option.

The configuration file may optionally define a second code page value for each character column (externalfilecodepage). If character data is stored in an external overflow file (see ["External Overflow Files" on page 95](#)), this second code page value is used for the external file.

## External Overflow Files

In addition to the bulk load data file, DataDirect Bulk Load can store bulk data in external overflow files. These overflow files are located in the same directory as the bulk load data file. Whether or not to use external overflow files is a performance consideration. For example, binary data is stored as hexadecimal-encoded character strings in the main bulk load data file, which increases the size of the file per unit of data stored. External files do not store binary data as hex character strings, and, therefore, require less space. On the other hand, more overhead is required to access external files than to access a single bulk load data file, so each bulk load situation must be considered individually.

The value of the Bulk Binary Threshold connection option determines the threshold, in KB, over which binary data is stored in external files instead of in the bulk load data file. Likewise, the Bulk Character Threshold connection option determines the threshold for character data.

In the case of an external character data file, the character set of the file is governed by the bulk load configuration file. If the bulk load data file is Unicode and the maximum character size of the source data is 1, then the data is stored in its source code page. See ["Character Set Conversions" on page 94](#).

The file name of the external file contains the bulk load data file name, a six-digit number, and a ".lob" extension in the following format: *CSVfilename\_nnnnnn.lob*. Increments start at 000001.lob.

## Using Bulk Load for Batch Inserts

For all drivers that support bulk operations, except the Salesforce driver, the driver uses the native bulk load protocol for database connections when the Enable Bulk Load connection option is set to `true`. For example, if you set the Enable Bulk Load connection option to `true`, the driver would use bulk load for the native parameter array insert request.

In some cases, the driver may not be able to use bulk load because of restrictions enforced by the bulk load protocol and will downgrade to a batch mechanism. For example, if the data being loaded has a data type that is not supported by the bulk load protocol, the driver cannot use bulk load, but will use the native parameter array insert mechanism instead.

For the Salesforce driver, when the Enable Bulk Load connection option is set to `true` and the number of rows to be inserted in the batch is larger than Bulk Load Threshold, the driver uses the Salesforce Bulk API instead of the Web service API.

For all drivers that support bulk operations, use the Bulk Load Batch Size connection option to specify the number of rows the driver loads at a time when bulk loading data.

Performance can be improved by increasing the number of rows the driver loads at a time because fewer network round trips are required. Be aware that increasing the number of rows that are loaded also causes the driver to consume more memory on the client.

## Using Bulk Load for Single Inserts/Updates/Deletes (Salesforce Driver)

When the Enable Bulk Load connection option is set to 1, the driver uses the Salesforce Bulk API for single Insert, Update, and Delete statements if the number of rows affected by the operation exceeds the threshold set by the Bulk Load Threshold connection property.

For example, if you set the Enable Bulk Load connection option to 1 and the Bulk Load Threshold connection option to 2000, executing the following statement would use the Bulk API if the number of rows returned by `SELECT rowid, sys_name FROM account` is more than 2000 rows.

```
INSERT INTO tmpAccounts(accountId, accountName)
SELECT rowid, sys_name FROM account
```

## Summary of Bulk Load Related Options

[Table 3-10](#) summarizes how DataDirect Bulk Load related connection options work with the drivers. See "Connection Option Descriptions" in each driver chapter for details about configuring the options.

---

**Table 3-10. Summary: Bulk Load Connection Options**

---

Option	Characteristic
Batch Size	An integer value that specifies the number of rows at a time that the driver sends to the database during bulk operations.
Bulk Binary Threshold	An integer value that specifies the maximum size, in KB, of binary data exported to the bulk data file. Any data exceeding this size is exported to an external file.
Bulk Character Threshold	An integer value that specifies the maximum size, in KB, of character data exported to the bulk data file. Any data exceeding this size is exported to an external file.
Enable Bulk Load	When enabled, the driver uses database bulk load protocols. When not enabled, the driver uses standard parameter arrays.

---



## 4 Configuring the Product on UNIX/Linux

This chapter contains specific information about using the DataDirect Connect Series *for* ODBC drivers in the UNIX and Linux environments. It discusses the following:

- ["Environment Variables" on page 97](#)
- ["The Test Loading Tool" on page 99](#)
- ["Data Source Configuration" on page 100](#)
- ["The demoodbc Application" on page 119](#)
- ["The example Application" on page 119](#)
- ["DSN-less Connections" on page 119](#)
- ["File Data Sources" on page 124](#)
- ["UTF-16 Applications on UNIX and Linux" on page 125](#)

See ["Environment-Specific Information" on page 45](#) for additional platform information.

---

### Environment Variables

The first step in setting up and configuring the drivers for use is to set several environment variables. The following procedures require that you have the appropriate permissions to modify your environment and to read, write, and execute various files. You must log in as a user with full r/w/x permissions recursively on the entire DataDirect Connect Series *for* ODBC installation directory.

#### Library Search Path

The library search path variable can be set by executing the appropriate shell script located in the ODBC home directory. From your login shell, determine which shell you are running by executing:

```
echo $SHELL
```

C shell login (and related shell) users must execute the following command before attempting to use ODBC-enabled applications:

```
source ./odbc.csh
```

Bourne shell login (and related shell) users must initialize their environment as follows:

```
. ./odbc.sh
```

Executing these scripts sets the appropriate library search path environment variable:

- LD\_LIBRARY\_PATH on HP-UX IPF, Linux, and Oracle Solaris
- LIBPATH on AIX
- SHLIB\_PATH on HP-UX PA-RISC

The library search path environment variable must be set so that the ODBC core components and drivers can be located at the time of execution. After running the setup script, execute:

```
env
```

to verify that the *installation\_directory/lib* directory has been added to your shared library path.

Some of the client-based drivers must have additional environment variables set. Consult the driver requirements in each of the individual driver chapters for additional environment variable information.

## ODBCINI

Setup installs in the product installation directory a default system information file, named *odbc.ini*, that contains data sources. See ["Data Source Configuration" on page 100](#) for an explanation of the *odbc.ini* file. The system administrator can choose to rename the file and/or move it to another location. In either case, the environment variable ODBCINI must be set to point to the fully qualified path name of the *odbc.ini* file.

For example, to point to the location of the file for an installation on */opt/odbc* in the C shell, you would set this variable as follows:

```
setenv ODBCINI /opt/odbc/odbc.ini
```

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

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

As an alternative, you can choose to make the *odbc.ini* file a hidden file and not set the ODBCINI variable. In this case, you would need to rename the file to *.odbc.ini* (to make it a hidden file) and move it to the user's \$HOME directory.

The driver searches for the location of the *odbc.ini* file as follows:

- 1 The driver checks the ODBCINI variable
- 2 The driver checks \$HOME for *.odbc.ini*

If the driver does not locate the system information file, it returns an error.

## ODBCINST

Setup installs in the product installation directory a default file, named *odbcinst.ini*, for use with DSN-less connections. See ["DSN-less Connections" on page 119](#) for an explanation of the *odbcinst.ini* file. The system administrator can choose to rename the file or move it to another location. In either case, the environment variable ODBCINST must be set to point to the fully qualified path name of the *odbcinst.ini* file.

For example, to point to the location of the file for an installation on */opt/odbc* in the C shell, you would set this variable as follows:

```
setenv ODBCINST /opt/odbc/odbcinst.ini
```

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

```
ODBCINST=/opt/odbc/odbcinst.ini;export ODBCINST
```

As an alternative, you can choose to make the odbcinst.ini file a hidden file and not set the ODBCINST variable. In this case, you would need to rename the file to .odbcinst.ini (to make it a hidden file) and move it to the user's \$HOME directory.

The driver searches for the location of the odbcinst.ini file as follows:

- 1 The driver checks the ODBCINST variable
- 2 The driver checks \$HOME for .odbcinst.ini

If the driver does not locate the odbcinst.ini file, it returns an error.

## DD\_INSTALLDIR

This variable provides the driver with the location of the product installation directory so that it can access support files. DD\_INSTALLDIR must be set to point to the fully qualified path name of the installation directory.

For example, to point to the location of the directory for an installation on /opt/odbc in the C shell, you would set this variable as follows:

```
setenv DD_INSTALLDIR /opt/odbc
```

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

```
DD_INSTALLDIR=/opt/odbc;export DD_INSTALLDIR
```

The driver searches for the location of the installation directory as follows:

- 1 The driver checks the DD\_INSTALLDIR variable
- 2 The driver checks the odbc.ini or the odbcinst.ini files for the InstallDir keyword (see ["Configuration Through the System Information \(odbc.ini\) File" on page 103](#) for a description of the InstallDir keyword)

If the driver does not locate the installation directory, it returns an error.

The next step is to test load the driver.

---

## The Test Loading Tool

The second step in preparing to use a driver is to test load it.

The ivtestlib (32-bit drivers) and ddtestlib (64-bit drivers) test loading tools are provided to test load drivers and help diagnose configuration problems in the UNIX and Linux environments, such as environment variables not correctly set or missing database client components. This tool is installed in the /bin subdirectory in the product installation directory. It attempts to load a specified ODBC driver and prints out all available error information if the load fails.

For example, if the drivers are installed in `/opt/odbc/lib`, the following command attempts to load the 32-bit Oracle Wire Protocol driver on Solaris, where `xx` represents the version number of the driver:

```
ivtestlib /opt/odbc/lib/ivoraxx.so
```

NOTE: On Solaris, AIX, and Linux, the full path to the driver does not have to be specified for the tool. The HP-UX version, however, requires the full path.

If the load is successful, the tool returns a success message along with the version string of the driver. If the driver cannot be loaded, the tool returns an error message explaining why.

See ["Version String Information" on page 57](#) for details about version strings.

The next step is to configure a data source through the system information file.

---

## Data Source Configuration

In the UNIX and Linux environments, a system information file is used to store data source information. Setup installs a default version of this file, called `odbc.ini`, in the product installation directory (see ["ODBCINI" on page 98](#) for details about relocating and renaming this file). This is a plain text file that contains data source definitions. If you have a Motif graphical user interface (GUI) in your Linux environment, you can use the DataDirect ODBC Data Source Administrator for Linux (the Linux ODBC Administrator) to create or modify data source definitions in this file (see ["Configuration Through the Administrator" on page 100](#) for details). If you do not, see ["Configuration Through the System Information \(odbc.ini\) File" on page 103](#) and follow the instructions for configuring the file.

NOTE: The Linux ODBC Administrator is currently supported only on Linux for x86 and x64 processors with Motif 2.0.3 and higher. It is not supported on Linux for Itanium II or other UNIX platforms.

On Linux, you can determine if you are using Motif through the following command:

```
rpm -qa |grep motif
```

The `rpm` command returns output similar to:

```
nc-linuxqa3[/home2/users/mike] rpm -qa |grep motif
openmotif-2.2.2-124
openmotif-devel-2.2.2-124
```

If you are not using a GUI, you can use any text editor to create or modify data source definitions directly in the `odbc.ini` file. See ["Configuration Through the System Information \(odbc.ini\) File" on page 103](#) for details.

### Configuration Through the Administrator

The Linux ODBC Administrator is located in the `/tools` directory of the product installation directory. For example,

```
/opt/odbc/tools/odbcadmin
```

The following drivers can be configured with the Linux ODBC Administrator:

- DB2 Wire Protocol
- dBASE
- Greenplum Wire Protocol
- Informix Wire Protocol
- Oracle
- Oracle Wire Protocol
- PostgreSQL Wire Protocol
- SQL Server Legacy Wire Protocol
- Sybase Wire Protocol
- The Driver for Teradata
- Text

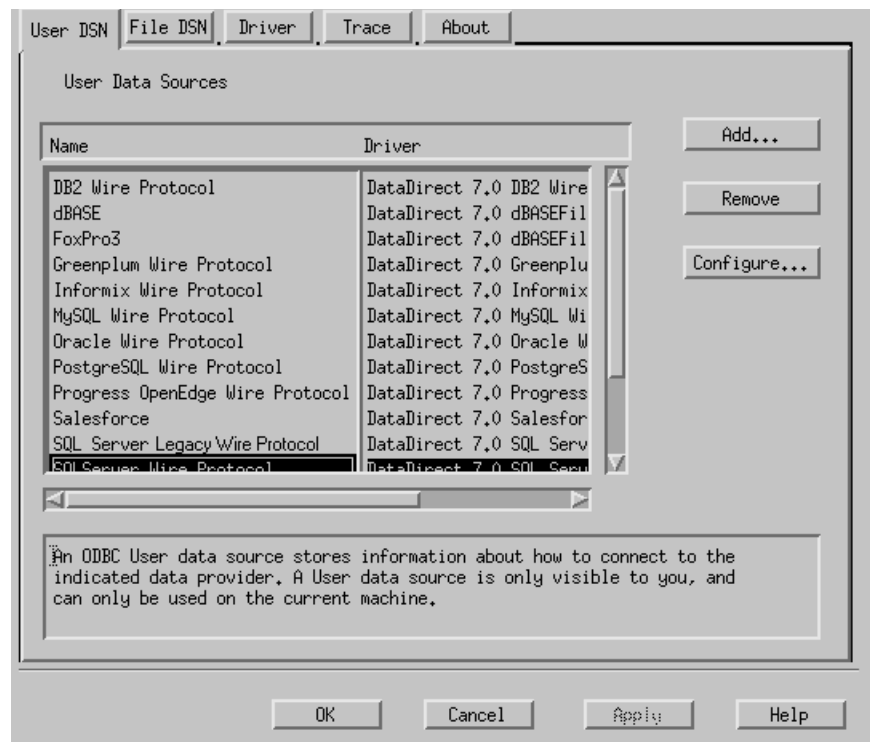
Note that all of the drivers included in your `odbc.ini` file are shown in the User Data Sources list box, even though some of them cannot be used with the Linux ODBC Administrator.

#### To configure a data source:

- 1 To start the Linux ODBC Administrator, change to the `install_dir/tools` directory, where `install_dir` is the path to the product installation directory. At a command prompt, enter:

```
./odbcadmin
```

The Administrator dialog box appears.



2 Click either the **User DSN** or **File DSN** tab to display a list of data sources.

- **User DSN:** If you are configuring an existing user data source, select the appropriate data source name and click **Configure** to display the driver Setup dialog box.

If you are configuring a new user data source, click **Add** to display a list of installed drivers. Select the appropriate driver and click **Finish** to display the driver Setup dialog box.

- **File DSN:** If you are configuring an existing file data source, select the appropriate data source file and click **Configure** to display the driver Setup dialog box.

To configure a new file data source, click **Add** to display a list of installed drivers. Select the appropriate driver and click **Advanced** to specify attributes; otherwise, click **Next** to proceed. Specify a name for the data source and click **Next**. Verify the data source information; then, click **Finish** to display the driver Setup dialog box.

NOTE: If you want to set a default directory for File DSNs, select the directory from the Directories list; then, click **Set Directory**. The next time that you open the Administrator, it displays data source files from this directory.

The General tab of the driver Setup dialog box appears by default.

See the individual driver chapters for specific information about the driver Setup dialog. See [Chapter 1 "Quick Start Connect" on page 35](#) for an explanation of different types of data sources.

## Drivers

The Drivers tab shows a list of all installed ODBC drivers.

## Tracing

The Tracing tab allows you to trace calls to ODBC drivers and create a log of the traces for troubleshooting purposes.

To specify the path and name of the trace log file, type the path and name in the Trace File field or click **Browse** to select a log file. If no location is specified, the trace log resides in the working directory of the application you are using.

DataDirect ships an enhanced library to perform tracing. This library appears by default in the Trace Library field. If you want to use a custom library instead, type the path and name of the library in the Trace Library field or click **Browse** to select a library.

To enable tracing, select the **Enable Tracing** check box on the Trace tab of the Administrator. Clear the check box to disable tracing. Tracing continues until you disable it. Be sure to turn off tracing when you are finished reproducing the issue because tracing decreases the performance of your ODBC application.

After making changes on the Trace tab, click **Apply** for them to take effect.

For a more complete discussion of tracing, refer to ["ODBC Trace" in Chapter 1](#) of the *DataDirect Connect Series for ODBC Troubleshooting Guide*.

When you are finished with the Linux ODBC Administrator, click **OK** or **Cancel**. If you click **OK**, any changes you have made to the Trace tab are accepted and the Administrator closes. If you click **Cancel**, the Administrator closes without saving any changes.

## Configuration Through the System Information (odbc.ini) File

To configure a data source manually, you edit the odbc.ini file with a text editor. The content of this file is divided into three sections.

At the beginning of the file is a section named [ODBC Data Sources] containing *data\_source\_name=installed-driver* pairs, for example:

```
Oracle Wire Protocol=DataDirect Oracle Wire Protocol.
```

The driver uses this section to match a data source to the appropriate installed driver.

The [ODBC Data Sources] section also includes data source definitions. The default odbc.ini contains a data source definition for each driver. Each data source definition begins with a data source name in square brackets, for example, [Oracle Wire Protocol 2]. The data source definitions contain connection string *attribute=value* pairs with default values. You can modify these values as appropriate for your system. Descriptions of these attributes are in each individual driver chapter. See ["Sample Default odbc.ini File" on page 104](#) for sample data sources.

The second section of the file is named [ODBC File DSN] and includes one keyword:

```
[ODBC File DSN]
DefaultDSNDir=
```

This keyword defines the path of the default location for file data sources (see ["File Data Sources" on page 124](#)).

NOTE: This section is not included in the default odbc.ini file that is installed by the product installer. You can add this section manually or, if you are using the Linux ODBC Administrator, it will be added automatically when you click **Set Directory** on the File DSN tab (see [Step 2](#) under ["Configuration Through the Administrator" on page 100](#)).

The third section of the file is named [ODBC] and includes several keywords, for example:

```
[ODBC]
IANAAppCodePage=4
InstallDir=/opt/odbc
Trace=0
TraceFile=odbctrace.out
TraceDll=/opt/odbc/lib/ivtrc27.so
ODBCTraceMaxFileSize=102400
ODBCTraceMaxNumFiles=10
```

The IANAAppCodePage keyword defines the default value that all UNIX/Linux drivers use if individual data sources have not specified a different value. See the individual driver chapters and refer to [Chapter 1 "Code Page Values"](#) in the *DataDirect Connect Series for ODBC Reference* for details. The default value is 4.

The InstallDir keyword must be included in this section. The value of this keyword is the path to the installation directory under which the /lib and /locale directories are contained. The installation process automatically writes your installation directory to the default odbc.ini file.

For example, if you choose an installation location of /opt/odbc, then the following line is written to the [ODBC] section of the default odbc.ini:

```
InstallDir=/opt/odbc
```

NOTE: If you are using only DSN-less connections through an odbcinst.ini file and do not have an odbc.ini file, then you must provide [ODBC] section information in the [ODBC] section of the odbcinst.ini file. The drivers and Driver Manager always check first in the [ODBC] section of an odbc.ini file. If no odbc.ini file exists or if the odbc.ini file does not contain an [ODBC] section, they check for an [ODBC] section in the odbcinst.ini file. See ["DSN-less Connections" on page 119](#) for details.

ODBC tracing allows you to trace calls to ODBC drivers and create a log of the traces for troubleshooting purposes. The following keywords all control tracing: Trace, TraceFile, TraceDLL, ODBCTraceMaxFileSize, and ODBCTraceMaxNumFiles.

For a complete description of these keywords and discussion of tracing, refer to ["ODBC Trace"](#) in [Chapter 1](#) of the *DataDirect Connect Series for ODBC Troubleshooting Guide*.

## Sample Default odbc.ini File

The following is a sample odbc.ini file that Setup installs in the installation directory. All occurrences of ODBCHOME are replaced with your installation directory path during installation of the file. Values that you must supply are enclosed by angle brackets (< >). If you are using the installed odbc.ini file, you must supply the values and remove the angle brackets before that data source section will operate properly. Commented lines are denoted by the # symbol. This sample shows 32-bit drivers with file names beginning with iv. A 64-bit driver file would be identical except that driver names would begin with dd and the list of data sources would include only the 64-bit drivers.

```
[ODBC Data Sources]
Apache Hive Wire Protocol=DataDirect 7.1 Apache Hive WP Driver
DB2 Wire Protocol=DataDirect 7.1 DB2 Wire Protocol
dBASE=DataDirect 7.1 dBASEFile (*.dbf)
FoxPro3=DataDirect 7.1 dBASEFile (*.dbf)
Greenplum Wire Protocol=DataDirect 7.1 Greenplum Wire Protocol
Informix Wire Protocol=DataDirect 7.1 Informix Wire Protocol
MySQL Wire Protocol=DataDirect 7.1 MySQL Wire Protocol
Oracle Wire Protocol=DataDirect 7.1 Oracle Wire Protocol
PostgreSQL Wire Protocol=DataDirect 7.1 PostgreSQL Wire Protocol
Progress OpenEdge Wire Protocol=DataDirect 7.1 Progress OpenEdge Wire Protocol
Salesforce=DataDirect 7.1 Salesforce
SQLServer Wire Protocol=DataDirect 7.1 SQL Server Wire Protocol
Sybase Wire Protocol=DataDirect 7.1 Sybase Wire Protocol
Sybase IQ Wire Protocol=DataDirect 7.1 Sybase IQ Wire Protocol
Teradata=DataDirect 7.1 Teradata
Text=DataDirect 7.1 TextFile (*.*)
Informix=DataDirect 7.1 Informix
Oracle=DataDirect 7.1 Oracle
SQLServ Legacy Wire Protocol=DataDirect 7.1 SQL Server Legacy Wire Protocol
```



```
[Apache Hive Wire Protocol]
Driver=ODBCHOME/lib/ivhive27.so
Description=DataDirect Apache Hive WP Driver
ArraySize=1024
Database=<database_name>
DataSourceName=default
DefaultLongDataBuffLen=1024
EnableDescribeParam=0
HostName=
LoginTimeout=30
LogonID=
MaxVarcharSize=
Password=
PortNumber=
RemoveColumnQualifiers=0
StringDescribeType=12
TransactionMode=0
UseCurrentSchema=0
```

```
[DB2 Wire Protocol]
Driver=ODBCHOME/lib/ivdb227.so
Description=DataDirect 7.1 DB2 Wire Protocol
AccountingInfo=
AddStringToCreateTable=
AlternateID=
AlternateServers=
ApplicationName=
ApplicationUsingThreads=1
AuthenticationMethod=0
BulkBinaryThreshold=32
BulkCharacterThreshold=-1
BulkLoadBatchSize=1024
BulkLoadFieldDelimiter=
BulkLoadRecordDelimiter=
CatalogSchema=
CharsetFor65535=0
ClientHostName=
ClientUser=
#Collection applies to z/OS and iSeries only
Collection=
ConcurrentAccessResolution=0
ConnectionReset=0
ConnectionRetryCount=0
ConnectionRetryDelay=3
CurrentFunctionPath=
#Database applies to DB2 UDB only
Database=<database_name>
DefaultIsolationLevel=1
DynamicSections=1000
EnableBulkLoad=0
EncryptionMethod=0
FailoverGranularity=0
FailoverMode=0
FailoverPreconnect=0
```

```

FetchTSWTZasTimestamp=0
GrantAuthid=PUBLIC
GrantExecute=1
GSSClient=native
HostNameInCertificate=
IpAddress=<DB2_server_host>
KeyPassword=
KeyStore=
KeyStorePassword=
LoadBalanceTimeout=0
LoadBalancing=0
#Location applies to z/OS and iSeries only
Location=<location_name>
LoginTimeout=15
LogonID=
MaxPoolSize=100
MinPoolSize=0
Password=
PackageCollection=NULLID
PackageNamePrefix=DD
PackageOwner=
Pooling=0
ProgramID=
QueryTimeout=0
ReportCodePageConversionErrors=0
TcpPort=50000
TrustStore=
TrustStorePassword=
UseCurrentSchema=1
ValidateServerCertificate=1
WithHold=1
XMLDescribeType=-10

```

```

[dBASE]
Driver=ODBCHOME/lib/ivdbf27.so
Description=DataDirect 7.1 dBASEFile (*.dbf)
ApplicationUsingThreads=1
CacheSize=4
CreateType=dBASE5
Database=ODBCHOME/demo
DataFileExtension=DBF
ExtensionCase=UPPER
FileOpenCache=0
IntlSort=0
LockCompatibility=dBASE
Locking=RECORD
UseLongNames=0
UseLongQualifiers=0

```

```

[FoxPro3]
Driver=ODBCHOME/lib/ivdbf27.so
Description=DataDirect 7.1 dBASEFile (*.dbf)
ApplicationUsingThreads=1
CacheSize=4
CreateType=FoxPro30

```

```

Database=ODBCHOME/demo
DataFileExtension=DBF
ExtensionCase=UPPER
FileOpenCache=0
IntlSort=0
LockCompatibility=Fox
Locking=RECORD
UseLongNames=0
UseLongQualifiers=0

[Greenplum Wire Protocol]
Driver=ODBCHOME/lib/ivgplm27.so
Description=DataDirect 7.1 Greenplum Wire Protocol
AlternateServers=
ApplicationUsingThreads=1
ConnectionReset=0
ConnectionRetryCount=0
ConnectionRetryDelay=3
Database=<database_name>
DefaultLongDataBuffLen=2048
EnableDescribeParam=1
EnableKeysetCursors=0
EncryptionMethod=0
ExtendedColumnMetaData=0
FailoverGranularity=0
FailoverMode=0
FailoverPreconnect=0
FetchRefCursors=1
FetchTSWTZasTimestamp=0
FetchTWFSasTime=0
HostName=<Greenplum_host>
HostNameInCertificate=
InitializationString=
KeyPassword=
KeysetCursorOptions=0
KeyStore=
KeyStorePassword=
LoadBalanceTimeout=0
LoadBalancing=0
LoginTimeout=15
LogonID=
MaxLongVarcharSize=
MaxPoolSize=100
MaxVarcharSize=
MinPoolSize=0
Password=
Pooling=0
PortNumber=<Greenplum_server_port>
QueryTimeout=0
ReportCodepageConversionErrors=0
TransactionErrorBehavior=1
TrustStore=
TrustStorePassword=
ValidateServerCertificate=1
XMLDescribeType=-10

```

```

[Informix Wire Protocol]
Driver=ODBCHOME/lib/ivifcl27.so
Description=DataDirect 7.1 Informix Wire Protocol
AlternateServers=
ApplicationUsingThreads=1
CancelDetectInterval=0
ConnectionRetryCount=0
ConnectionRetryDelay=3
Database=<database_name>
HostName=<Informix_host>
LoadBalancing=0
LogonID=
Password=
PortNumber=<Informix_server_port>
ServerName=<Informix_server>
TrimBlankFromIndexName=1
UseDelimitedIdentifiers=0

```

```

[MySQL Wire Protocol]
Driver=ODBCHOME/lib/ivmysql27.so
Description=DataDirect 7.1 MySQL Wire Protocol
AlternateServers=
ApplicationUsingThreads=1
ConnectionReset=0
ConnectionRetryCount=0
ConnectionRetryDelay=3
Database=<database_name>
DefaultLongDataBuffLen=1024
EnableDescribeParam=0
EncryptionMethod=0
FailoverGranularity=0
FailoverMode=0
FailoverPreconnect=0
HostName=<MySQL_host>
HostNameInCertificate=
InteractiveClient=0
KeyPassword=
Keystore=
KeystorePassword=
LoadBalanceTimeout=0
LoadBalancing=0
LoginTimeout=15
LogonID=
Password=
MaxPoolSize=100
MinPoolSize=0
Pooling=0
PortNumber=<MySQL_server_port>
QueryTimeout=0
ReportCodepageConversionErrors=0
TreatBinaryAsChar=0
TrustStore=
TrustStorePassword=
ValidateServerCertificate=1

```

```

[Oracle Wire Protocol]
Driver=ODBCHOME/lib/ivora27.so
Description=DataDirect 7.1 Oracle Wire Protocol
AccountingInfo=
Action=
AlternateServers=
ApplicationName=
ApplicationUsingThreads=1
ArraySize=60000
AuthenticationMethod=1
BulkLoadBatchSize=1024
BulkBinaryThreshold=32
BulkCharacterThreshold=-1
BulkLoadFieldDelimiter=
BulkLoadOptions=0
BulkLoadRecordDelimiter=
CachedCursorLimit=32
CachedDescLimit=0
CatalogIncludesSynonyms=1
CatalogOptions=0
ClientHostName=
ClientID=
ClientUser=
ConnectionReset=0
ConnectionRetryCount=0
ConnectionRetryDelay=3
DataIntegrityLevel=0
DataIntegrityTypes=
DefaultLongDataBuffLen=1024
DescribeAtPrepare=0
EditionName=
EnableBulkLoad=0
EnableDescribeParam=0
EnableNcharSupport=0
EnableScrollableCursors=1
EnableServerResultCache=0
EnableStaticCursorsForLongData=0
EnableTimestampWithTimeZone=0
EncryptionLevel=0
EncryptionMethod=0
EncryptionTypes=
FailoverGranularity=0
FailoverMode=0
FailoverPreconnect=0
FetchTSWTZasTimestamp=0
GSSClient=native
HostName=<Oracle_server>
HostNameInCertificate=
InitializationString=
KeyPassword=
KeyStore=
KeyStorePassword
LoadBalanceTimeout=0
LoadBalancing=0

```

```

LocalTimeZoneOffset=
LockTimeOut=-1
LoginTimeout=15
LogonID=
MaxPoolSize=100
MinPoolSize=0
Module=0
Password=
Pooling=0
PortNumber=<Oracle_server_port>
ProcedureRetResults=0
ProgramID=
QueryTimeout=0
ReportCodePageConversionErrors=0
ReportRecycleBin=0
ServerName=<server_name_in_tnsnames.ora>
ServerType=0
ServiceName=
SID=<Oracle_System_Identifier>
TimestampEscapeMapping=0
TNSNamesFile=<tnsnames.ora_filename>
TrustStore=
TrustStorePassword=
UseCurrentSchema=1
ValidateServerCertificate=1
WireProtocolMode=1

[PostgreSQL Wire Protocol]
Driver=ODBCHOME/lib/ivpsql27.so
Description=DataDirect 7.1 PostgreSQL Wire Protocol
AlternateServers=
ApplicationUsingThreads=1
ConnectionReset=0
ConnectionRetryCount=0
ConnectionRetryDelay=3
Database=<database_name>
EnableDescribeParam=1
EnableKeysetCursors=0
EncryptionMethod=0
ExtendedColumnMetaData=0
FailoverGranularity=0
FailoverMode=0
FailoverPreconnect=0
FetchTSWTZasTimestamp=0
FetchTWFSasTime=0
HostName=<PostgreSQL_host>
HostNameInCertificate=
InitializationString=
KeyPassword=
KeysetCursorOptions=0
Keysetstore=
KeystorePassword=
LoadBalanceTimeout=0
LoadBalancing=0
LoginTimeout=15

```

```

LogonID=
MaxLongVarcharSize=
MaxPoolSize=100
MaxVarcharSize=
MinPoolSize=0
Password=
Pooling=0
PortNumber=<PostgreSQL_server_port>
QueryTimeout=0
ReportCodepageConversionErrors=0
TransactionErrorBehavior=1
TrustStore=
TrustStorePassword=
ValidateServerCertificate=1
XMLDescribeType=-10

[Progress OpenEdge Wire Protocol]
Driver=ODBCHOME/lib/ivoe27.so
Description=DataDirect 7.1 Progress OpenEdge Wire Protocol
AlternateServers=
ArraySize=
ConnectionRetryCount=0
ConnectionRetryDelay=3
Database=<database_name>
DefaultIsolationLevel=1
EnableTimestampWithTimezone=1
Encryption Method=0
FailoverGranularity=0
FailoverMode=0
FailoverPreconnect=0
HostName=<Progress_server>
HostNameInCertificate=
LoadBalancing=0
LoginTimeout=15
PortNumber=<Progress_server_port>
QueryTimeout=0
TrustStore=
TrustStorePassword=
UseWideCharacterTypes=0
ValidateServerCertificate=1

[Salesforce]
Driver=ODBCHOME/lib/ivsfr27.so
Description=DataDirect 7.1 Salesforce
ApplicationUsingThreads=1
BulkLoadAsync=0
BulkLoadBatchSize=1024
BulkLoadConcurrencyMode=1
BulkLoadPollInterval=10
BulkLoadThreshold=4000
BulkLoadTimeout=0
ConnectionReset=0
ConfigOptions=
CreatedB=2
Database=

```

```

ExtendedOptions=
FetchSize=100
HostName=
InitializationString=
JVMArgs=-Xmx256m
JVMClasspath=
LoadBalanceTimeout=0
LogConfigFile=
LoginTimeout=15
LogonDomain=
LogonID=
MaxPoolSize=100
MinPoolSize=0
Password=
Pooling=0
ProxyHost=
ProxyPassword=
ProxyPort=
ProxyUser=
QueryTimeout=0
ReportCodepageConversionErrors=0
ReadOnly=0
RefreshCurrentSchema=0
RefreshDirtyCache=0
SecurityToken=
StmtCallLimit=20
StmtCallLimitBehavior=2
TransactionMode=0
WSFetchSize=0
WSRetryCount=0
WSTimeout=120

[SQLServer Wire Protocol]
Driver=ODBCHOME/lib/ivsqs27.so
Description=DataDirect 7.1 SQL Server Wire Protocol
AlternateServers=
AlwaysReportTriggerResults=0
AnsiNPW=1
ApplicationName=
ApplicationUsingThreads=1
AuthenticationMethod=1
BulkBinaryThreshold=32
BulkCharacterThreshold=-1
BulkLoadBatchSize=1024
BulkLoadFieldDelimiter=
BulkLoadOptions=2
BulkLoadRecordDelimiter=
ConnectionReset=0
ConnectionRetryCount=0
ConnectionRetryDelay=3
Database=<database_name>
DefaultLongDataBuffLen=1024
EnableBulkLoad=0
EnableQuotedIdentifiers=0
EncryptionMethod=0

```



```

FailoverGranularity=0
FailoverMode=0
FailoverPreconnect=0
FetchTSWTZasTimestamp=0
FetchTWFSasTime=1
GSSClient=native
HostName=<SQL_Server_host>
HostNameInCertificate=
InitializationString=
Language=
LoadBalanceTimeout=0
LoadBalancing=0
LoginTimeout=15
LogonID=
MaxPoolSize=100
MinPoolSize=0
PacketSize=-1
Password=
Pooling=0
PortNumber=<SQL_Server_server_port>
QueryTimeout=0
ReportCodePageConversionErrors=0
SnapshotSerializable=0
TrustStore=
TrustStorePassword=
ValidateServerCertificate=1
WorkStationID=
XMLDescribeType=-10

[Sybase Wire Protocol]
Driver=ODBCHOME/lib/ivase27.so
Description=DataDirect 7.1 Sybase Wire Protocol
AlternateServers=
ApplicationName=
ApplicationUsingThreads=1
ArraySize=50
AuthenticationMethod=0
BulkBinaryThreshold=32
BulkCharacterThreshold=-1
BulkLoadBatchSize=1024
BulkLoadFieldDelimiter=
BulkLoadRecordDelimiter=
Charset=
ConnectionReset=0
ConnectionRetryCount=0
ConnectionRetryDelay=3
CursorCacheSize=1
Database=<database_name>
DefaultLongDataBuffLen=1024
DistributedTransactionModel=0
EnableBulkLoad=0
EnableDescribeParam=0
EnableQuotedIdentifiers=0
EncryptionMethod=0
FailoverGranularity=0

```

```

FailoverMode=0
FailoverNetworkAddress=
FailoverPreconnect=0
FetchTWFSasTime=1
GSSClient=native
HostNameInCertificate=
InitializationString=
InterfacesFileServerName=
Language=
LoadBalanceTimeout=0
LoadBalancing=0
LoginTimeout=15
LogonID=
MaxPoolSize=100
MinPoolSize=0
NetworkAddress=<Sybase_host,Sybase_server_port>
OptimizePrepare=1
PacketSize=0
Password=
Pooling=0
QueryTimeout=0
RaiseErrorPositionBehavior=0
ReportCodePageConversionErrors=0
SelectMethod=0
ServicePrincipalName=
TightlyCoupledDistributedTransactions=
TruncateTimeTypeFractions=0
TrustStore=
TrustStorePassword=
ValidateServerCertificate=1
WorkStationID=
XAOpenStringParameters=

[Sybase IQ Wire Protocol]
Driver=ODBCHOME/lib/ivsyiq27.so
Description=DataDirect 7.1 Sybase IQ Wire Protocol
AlternateServers=
ApplicationName=
ApplicationUsingThreads=1
ArraySize=50
Charset=
ConnectionReset=0
ConnectionRetryCount=0
ConnectionRetryDelay=3
CursorCacheSize=1
Database=<database_name>
DefaultLongDataBuffLen=1024
DistributedTransactionModel=0
FailoverGranularity=0
FailoverMode=0
FailoverNetworkAddress=
FailoverPreconnect=0
FetchTWFSasTime=1
InitializationString=
InterfacesFile=

```

```

InterfacesFileServerName=
Language=
LoadBalanceTimeout=0
LoadBalancing=0
LoginTimeout=15
LogonID=
MaxPoolSize=100
MinPoolSize=0
NetworkAddress=<SybaseIQ_host,SybaseIQ_server_port>
OptimizePrepare=1
PacketSize=0
Password=
Pooling=0
QueryTimeout=0
RaiseErrorPositionBehavior=0
ReportCodePageConversionErrors=0
SelectMethod=0
TightlyCoupledDistributedTransactions=
TruncateTimeTypeFractions=0
WorkStationID=
XAOpenStringParameters=

```

```

[Teradata]
Driver=ODBCHOME/lib/ivtera27.so
Description=DataDirect 7.1 Teradata
AccountString=
AuthenticationDomain=
AuthenticationPassword=
AuthenticationUserid=
CharacterSet=ASCII
Database=
DBCName=<Teradata_server>
EnableDataEncryption=0
EnableExtendedStmtInfo=0
EnableLOBs=1
EnableReconnect=0
IntegratedSecurity=0
LoginTimeout=20
LogonID=
MapCallEscapeToExec=0
MaxRespSize=8192
Password=
PortNumber=1025
PrintOption=N
ProcedureWithSplSource=Y
ReportCodePageConversionErrors=0
SecurityMechanism=
SecurityParameter=
ShowSelectableTables=1
TDProfile=
TDRole=
TDUserName=

```

```

[Text]
Driver=ODBCHOME/lib/ivtxt27.so

```

```

Description=DataDirect 7.1 TextFile(*.*)
AllowUpdateAndDelete=0
ApplicationUsingThreads=1
CacheSize=4
CenturyBoundary=20
Database=ODBCHOME/demo
DataFileExtension=TXT
DecimalSymbol=.
Delimiter=,
FileOpenCache=0
FirstLineNames=0
IntlSort=0
ScanRows=25
TableType=Comma
UndefinedTable=GUESS

```

```

[Informix]
Driver=ODBCHOME/lib/ivinf27.so
Description=DataDirect 7.1 Informix
ApplicationUsingThreads=1
CancelDetectInterval=0
CursorBehavior=0
Database=<database_name>
EnableInsertCursors=0
GetDBListFromInformix=1
HostName=<Informix_host>
LogonID=
Password=
Protocol=onsoctcp
ServerName=<Informix_server>
Service=<Informix_service_name>
TrimBlankFromIndexName=1

```

```

[Oracle]
Driver=ODBCHOME/lib/ivor827.so
Description=DataDirect 7.1 Oracle
AlternateServers=
ApplicationUsingThreads=1
ArraySize=60000
CatalogIncludesSynonyms=1
CatalogOptions=0
ClientVersion=9iR2
ConnectionReset=0
ConnectionRetryCount=0
ConnectionRetryDelay=3
DefaultLongDataBuffLen=1024
DescribeAtPrepare=0
EnableDescribeParam=0
EnableNcharSupport=0
EnableScrollableCursors=1
EnableStaticCursorsForLongData=0
EnableTimestampWithTimeZone=0
LoadBalanceTimeout=0
LoadBalancing=0
LocalTimeZoneOffset=

```

```

LockTimeOut=-1
LogonID=
MaxPoolSize=100
MinPoolSize=0
OptimizeLongPerformance=0
Password=
Pooling=0
ProcedureRetResults=0
ReportCodePageConversionErrors=0
ReportRecycleBin=0
ServerName=<Oracle_server>
TimestampEscapeMapping=0
UseCurrentSchema=1

[SQLServ Legacy Wire Protocol]
Driver=ODBCHOME/lib/ivmsss27.so
Description=DataDirect 7.1 SQL Server Legacy Wire Protocol
Address=<SQLServer_host, SQLServer_server_port>
AlternateServers=
AnsiNPW=Yes
ConnectionRetryCount=0
ConnectionRetryDelay=3
Database=<database_name>
FetchTSWTZasTimestamp=0
FetchTWFSasTime=0
LoadBalancing=0
LogonID=
Password=
QuotedId=No
ReportCodepageConversionErrors=0
SnapshotSerializable=0

[ODBC]
IANAAppCodePage=4
InstallDir=ODBCHOME
Trace=0
TraceFile=odbctrace.out
TraceDll=ODBCHOME/lib/ivtrc27.so
ODBCTraceMaxFileSize=102400
ODBCTraceMaxNumFiles=10

```

To modify or create data sources in the `odbc.ini` file, use the following procedures.

**To modify a data source:**

- 1 Using a text editor, open the `odbc.ini` file.
- 2 Modify the default attributes in the data source definitions as necessary based on your system specifics, for example, enter the host name and port number of your system in the appropriate location.  
  
Consult the "Connection String Attributes" table of each driver chapter for other specific attribute values.
- 3 After making all modifications, save the `odbc.ini` file and close the text editor.

IMPORTANT: The "Connection Option Descriptions" section of each driver chapter lists both the long and short names of the attribute. When entering attribute names into `odbc.ini`, you must use the long name of the attribute. The short name is not valid in the `odbc.ini` file.

**To create a new data source:**

- 1 Using a text editor, open the `odbc.ini` file.
- 2 Copy an appropriate existing default data source definition and paste it to another location in the file.
- 3 Change the data source name in the copied data source definition to a new name. The data source name is between square brackets at the beginning of the definition, for example, `[Oracle Wire Protocol]`.
- 4 Modify the attributes in the new definition as necessary based on your system specifics, for example, enter the host name and port number of your system in the appropriate location.

Consult the "Connection String Attributes" table of each driver chapter for other specific attribute values.

- 5 In the `[ODBC]` section at the beginning of the file, add a new `data_source_name=installed-driver` pair containing the new data source name and the appropriate installed driver name.
- 6 After making all modifications, save the `odbc.ini` file and close the text editor.

IMPORTANT: The "Connection String Attributes" table of each driver chapter lists both the long and short name of the attribute. When entering attribute names into `odbc.ini`, you must use the long name of the attribute. The short name is not valid in the `odbc.ini` file.

## Translators

Progress DataDirect provides a sample translator named "OEM to ANSI" that provides a framework for coding a translation library. Refer to the `readme.trn` file in the `/samples/src/trn` subdirectory in the product installation directory for details.

To perform a translation with a particular driver, you must include the `TranslationSharedLibrary` keyword in that driver's data source definition in the `odbc.ini` file. The `TranslationSharedLibrary` keyword represents the full path to the translation library.

For example, the 32-bit DB2 driver would be:

```
[DB2]
Driver=ODBCHOME/lib/ivdb227.so
Description=DataDirect 7.1 DB2 Wire Protocol
TranslationSharedLibrary=ODBCHOME/lib/ivtrn27.so
```

The `TranslationOption` keyword is the ASCII representation of the 32-bit integer translation option. Use of the `TranslationOption` keyword is optional.

---

## The demoodbc Application

Progress DataDirect ships an application, named demoodbc, that is installed in the /samples/demo subdirectory of the product installation directory. Once you have set up your environment and data source, use the demoodbc application to test your connection. The syntax to run the application is:

```
demoodbc -uid user_name -pwd password data_source_name
```

For example:

```
demoodbc -uid johndoe -pwd secret DataSource3
```

The demoodbc application is coded to execute a Select statement from a table named emp. If you have an emp table in your database, the results are returned. If you do not have an emp table, you receive the message: Invalid object name 'EMP'. This message confirms a successful connection to the database.

Refer to the demoodbc.txt file in the demo subdirectory for an explanation of how to build and use this application.

---

## The example Application

Progress DataDirect ships an application, named example, that is installed in the /samples/example subdirectory of the product installation directory. Once you have configured your environment and data source, use the example application to test passing SQL statements. To run the application, enter example and follow the prompts to enter your data source name, user name, and password.

If successful, a SQL> prompt appears and you can type in SQL statements, such as `SELECT * FROM table_name`. If example is unable to connect to the database, an appropriate error message appears.

Refer to the example.txt file in the example subdirectory for an explanation of how to build and use this application.

---

## DSN-less Connections

Connections to a data source can be made via a connection string without referring to a data source name (DSN-less connections). This is done by specifying the "DRIVER=" keyword instead of the "DSN=" keyword in a connection string, as outlined in the ODBC specification. A file named odbcinstr.ini must exist when the driver encounters DRIVER= in a connection string.

Setup installs a default version of this file in the product installation directory (see ["ODBCINST" on page 98](#) for details about relocating and renaming this file). This is a plain text file that contains default DSN-less connection information. You should not normally need to edit this file. The content of this file is divided into several sections.

At the beginning of the file is a section named [ODBC Drivers] that lists installed drivers, for example,

```
DataDirect Oracle Wire Protocol=Installed.
```

This section also includes additional information for each driver.

The next section of the file is named [Administrator]. The keyword in this section, AdminHelpRootDirectory, is required for the Linux ODBC Administrator to locate its help system. The installation process automatically provides the correct value for this keyword.

The final section of the file is named [ODBC]. The [ODBC] section in the odbcinist.ini file fulfills the same purpose in DSN-less connections as the [ODBC] section in the odbc.ini file does for data source connections. See ["Configuration Through the System Information \(odbc.ini\) File" on page 103](#) for a description of the other keywords this section.

NOTE: The odbcinist.ini file and the odbc.ini file include an [ODBC] section. If the information in these two sections is not the same, the values in the odbc.ini [ODBC] section override those of the odbcinist.ini [ODBC] section.

## Sample odbcinist.ini File

The following is a sample odbcinist.ini. All occurrences of ODBCHOME are replaced with your installation directory path during installation of the file. Commented lines are denoted by the # symbol. This sample shows 32-bit drivers with file names beginning with iv; a 64-bit driver file would be identical except that driver names would begin with dd.

```
[ODBC Drivers]
DataDirect 7.1 Apache Hive WP Driver
DataDirect 7.1 DB2 Wire Protocol=Installed
DataDirect 7.1 dBASEFile (*.dbf)=Installed
DataDirect 7.1 Greenplum Wire Protocol=Installed
DataDirect 7.1 Informix Wire Protocol=Installed
DataDirect 7.1 MySQL Wire Protocol=Installed
DataDirect 7.1 Oracle Wire Protocol=Installed
DataDirect 7.1 PostgreSQL Wire Protocol=Installed
DataDirect 7.1 Progress OpenEdge Wire Protocol=Installed
DataDirect 7.1 SQL Server Wire Protocol=Installed
DataDirect 7.1 Sybase Wire Protocol=Installed
DataDirect 7.1 Salesforce=Installed
DataDirect 7.1 Teradata=Installed
DataDirect 7.1 TextFile (*.*)=Installed
DataDirect 7.1 Informix=Installed
DataDirect 7.1 Oracle=Installed
DataDirect 7.1 SQL Server Legacy Wire Protocol=Installed

[Apache Hive Wire Protocol]
Driver=ODBCHOME/lib/ivhive27.so
Description=DataDirect 7.1 Apache Hive WP Driver
ArraySize=1024
Database=<database_name>
DataSourceName=default
DefaultLongDataBuffLen=1024
EnableDescribeParam=0
HostName=
```



```

LoginTimeout=30
LogonID=
MaxVarcharSize=
Password=
PortNumber=
RemoveColumnQualifiers=0
StringDescribeType=12
TransactionMode=0
UseCurrentSchema=0

[DataDirect 7.1 DB2 Wire Protocol]
Driver=ODBCHOME/lib/ivdb227.so
APILevel=0
ConnectFunctions=YYY
DriverODBCVer=3.52
FileUsage=0
HelpRootDirectory=ODBCHOME/help
Setup=ODBCHOME/lib/ivdb227.so
SQLLevel=0

[DataDirect 7.1 dBASEFile (*.dbf)]
Driver=ODBCHOME/lib/ivdbf27.so
APILevel=0
ConnectFunctions=YYY
DriverODBCVer=3.52
FileExtns=*.dbf
FileUsage=1
HelpRootDirectory=ODBCHOME/help
Setup=ODBCHOME/lib/ivdbf27.so
SQLLevel=0

[DataDirect 7.1 Greenplum Wire Protocol]
Driver=ODBCHOME/lib/ivgplm27.so
APILevel=0
ConnectFunctions=YYY
DriverODBCVer=3.52
FileUsage=0
HelpRootDirectory=ODBCHOME/help
Setup=ODBCHOME/lib/ivgplm27.so
SQLLevel=0

[DataDirect 7.1 Informix Wire Protocol]
Driver=ODBCHOME/lib/ivifcl27.so
APILevel=0
ConnectFunctions=YYY
DriverODBCVer=3.52
FileUsage=0
HelpRootDirectory=ODBCHOME/help
Setup=ODBCHOME/lib/ivifcl27.so
SQLLevel=0

[DataDirect 7.1 MySQL Wire Protocol]
Driver=ODBCHOME/lib/ivmysql27.so
APILevel=0
ConnectFunctions=YYY

```

```

DriverODBCVer=3.52
FileUsage=0
HelpRootDirectory=ODBCHOME/help
Setup=ODBCHOME/lib/ivmysql27.so
SQLLevel=0

[DataDirect 7.1 Oracle Wire Protocol]
Driver=ODBCHOME/lib/ivora27.so
APILevel=0
ConnectFunctions=YYY
DriverODBCVer=3.52
FileUsage=0
HelpRootDirectory=ODBCHOME/help
Setup=ODBCHOME/lib/ivora27.so
SQLLevel=0

[DataDirect 7.1 PostgreSQL Wire Protocol]
Driver=ODBCHOME/lib/ivpsql27.so
APILevel=0
ConnectFunctions=YYY
DriverODBCVer=3.52
FileUsage=0
HelpRootDirectory=ODBCHOME/help
Setup=ODBCHOME/lib/ivpsql27.so
SQLLevel=0

[DataDirect 7.1 Progress OpenEdge Wire Protocol]
Driver=ODBCHOME/lib/ivoe27.so
APILevel=1
ConnectFunctions=YYN
DriverODBCVer=3.52
SQLLevel=0

[DataDirect 7.1 SQL Server Wire Protocol]
Driver=ODBCHOME/lib/ivsqs27.so
APILevel=1
ConnectFunctions=YYY
DriverODBCVer=3.52
FileUsage=0
HelpRootDirectory=ODBCHOME/help
SQLLevel=0

[DataDirect 7.1 Sybase Wire Protocol]
Driver=ODBCHOME/lib/ivase27.so
APILevel=0
ConnectFunctions=YYY
DriverODBCVer=3.52
FileUsage=0
HelpRootDirectory=ODBCHOME/help
Setup=ODBCHOME/lib/ivase27.so
SQLLevel=0

[DataDirect 7.1 Salesforce]
Driver=ODBCHOME/lib/ivsfr27.so
APILevel=0

```

```

ConnectFunctions=YYY
DriverODBCVer=3.52
FileUsage=0
HelpRootDirectory=ODBCHOME/help
SQLLevel=0

[DataDirect 7.1 Sybase IQ Wire Protocol]
Driver=ODBCHOME/lib/ivsyiq27.so
APILevel=0
ConnectFunctions=YYY
DriverODBCVer=3.52
FileUsage=0
HelpRootDirectory=ODBCHOME/help
SQLLevel=0

[DataDirect 7.1 Teradata]
Driver=ODBCHOME/lib/ivtera27.so
APILevel=0
ConnectFunctions=YYY
DriverODBCVer=3.52
FileUsage=0
HelpRootDirectory=ODBCHOME/help
Setup=ODBCHOME/lib/ivtera27s.so
SQLLevel=0

[DataDirect 7.1 TextFile (*.*)]
Driver=ODBCHOME/lib/ivtxt27.so
APILevel=0
ConnectFunctions=YYY
DriverODBCVer=3.52
FileExtns=*. *
FileUsage=1
HelpRootDirectory=ODBCHOME/help
Setup=ODBCHOME/lib/ivtxt27.so
SQLLevel=0

[DataDirect 7.1 Informix]
Driver=ODBCHOME/lib/ivinf27.so
APILevel=0
ConnectFunctions=YYY
DriverODBCVer=3.52
FileUsage=0
HelpRootDirectory=ODBCHOME/help
SQLLevel=0

[DataDirect 7.1 Oracle]
Driver=ODBCHOME/lib/ivor827.so
APILevel=0
ConnectFunctions=YYY
DriverODBCVer=3.52
FileUsage=0
HelpRootDirectory=ODBCHOME/help
Setup=ODBCHOME/lib/ivor827s.so
SQLLevel=0

```

```

[DataDirect 7.1 SQL Server Legacy Wire Protocol]
Driver=ODBCHOME/lib/ivmsss27.so
APILevel=2
ConnectFunctions=YYY
DriverODBCVer=3.52
FileUsage=0
HelpRootDirectory=ODBCHOME/help
Setup=ODBCHOME/lib/ivmsss27s.so
SQLLevel=0

[Administrator]
HelpRootDirectory=ODBCHOME/adminhelp

[ODBC]
#This section must contain values for DSN-less connections
#if no odbc.ini file exists. If an odbc.ini file exists,
#the values from that [ODBC] section are used.

IANAAppCodePage=4
InstallDir=ODBCHOME
Trace=0
TraceFile=odbctrace.out
TraceDll=ODBCHOME/lib/odbcsrc27.so
ODBCTraceMaxFileSize=102400
ODBCTraceMaxNumFiles=10

```

---

## File Data Sources

The Driver Manager on UNIX and Linux supports file data sources. The advantage of a file data source is that it can be stored on a server and accessed by other machines, either Windows, UNIX, or Linux. See ["Quick Start Connect" on page 35](#) for a general description of ODBC data sources on both Windows and UNIX.

A file data source is simply a text file that contains connection information. It can be created through the Linux ODBC Administrator (see ["Configuration Through the Administrator" on page 100](#)) or it can be created with a text editor. The file normally has an extension of .dsn.

For example, a file data source for the Oracle Wire Protocol driver would be similar to the following:

```

[ODBC]
Driver=DataDirect Oracle Wire Protocol
Port=1522
HostName=ORA2
LogonID=JOHN
Servicename=SALES.US.ACME.COM
CatalogOptions=1

```

It must contain all basic connection information plus any optional attributes. Because it uses the "DRIVER=" keyword, an odbcinst.ini file containing the driver location must exist (see ["DSN-less Connections" on page 119](#)).

The file data source is accessed by specifying the "FILEDSN=" instead of the "DSN=" keyword in a connection string, as outlined in the ODBC specification. The complete path to the file data source can be specified in the syntax that is normal for the machine on which the file is located. For example, on Windows:

```
FILEDSN=C:\Program Files\Common Files\ODBC\DataSources\Oraclewp.dsn
```

or, on UNIX and Linux:

```
FILEDSN=/home/users/john/filedsn/Oraclewp2.dsn
```

If no path is specified for the file data source, the Driver Manager uses the DefaultDSNDir property, which is defined in the [ODBC File DSN] setting in the odbc.ini file to locate file data sources (see ["Configuration Through the System Information \(odbc.ini\) File" on page 103](#) for details). If the [ODBC File DSN] setting is not defined, the Driver Manager uses the InstallDir setting in the [ODBC] section of the odbc.ini file. The Driver Manager does not support the SQLReadFileDSN and SQLWriteFileDSN functions.

As with any connection string, you can specify attributes to override the default values in the data source:

```
FILEDSN=/home/users/john/filedsn/Oraclewp2.dsn;UID=james;PWD=test01
```

---

## UTF-16 Applications on UNIX and Linux

Because the DataDirect Driver Manager allows applications to use either UTF-8 or UTF-16 Unicode encoding, applications written in UTF-16 for Windows platforms can also be used on UNIX and Linux platforms.

The Driver Manager assumes a default of UTF-8 applications; therefore, two things must occur for it to determine that the application is UTF-16:

- The definition of SQLWCHAR in the ODBC header files must be switched from "char \*" to "short \*." To do this, the application uses #define SQLWCHARSHORT.
- The application must set the ODBC environment attribute SQL\_ATTR\_APP\_UNICODE\_TYPE to a value of SQL\_DD\_CP\_UTF16, for example:

```
rc = SQLSetEnvAttr(*henv, SQL_ATTR_APP_UNICODE_TYPE,
(SQLPOINTER)SQL_DD_CP_UTF16, SQL_IS_INTEGER);
```



## Part 2: The 32-Bit/64-Bit Drivers

This part describes the drivers that are available in both 32- and 64-bit versions. See ["Part 3: The 32-Bit Drivers" on page 541](#) for the drivers that are available only in 32-bit versions. See ["Part 4: The Connect XE Drivers" on page 667](#) for information about the Salesforce driver and the Driver for Teradata.

This part contains the following chapters:

- [Chapter 5 "The DB2 Wire Protocol Driver" on page 129](#)
- [Chapter 6 "The Informix Wire Protocol Driver" on page 191](#)
- [Chapter 7 "The MySQL Wire Protocol Driver" on page 209](#)
- [Chapter 8 "The Oracle Wire Protocol Driver" on page 241](#)
- [Chapter 9 "The PostgreSQL Wire Protocol Driver" on page 315](#)
- [Chapter 10 "The Progress OpenEdge® Wire Protocol Driver" on page 351](#)
- [Chapter 11 "The SQL Server Wire Protocol Driver" on page 375](#)
- [Chapter 12 "The Sybase Wire Protocol Driver" on page 421](#)
- [Chapter 13 "The Oracle Driver" on page 477](#)
- [Chapter 14 "The SQL Server Legacy Wire Protocol Driver" on page 515](#)





## 5 The DB2 Wire Protocol Driver

The DataDirect Connect *for* ODBC and DataDirect Connect64 *for* ODBC DB2 Wire Protocol driver (the DB2 Wire Protocol driver) each support the following DB2 database servers:

- DB2 V10.1 for Linux, UNIX, Windows
- DB2 V9.1, V9.5, V9.7 for Linux, UNIX, Windows
- DB2 V8.x for Linux, UNIX, Windows
- DB2 10 for z/OS
- DB2 V9.1 for z/OS
- DB2 UDB V8.1 for z/OS
- DB2 UDB V7R1 for iSeries
- DB2 UDB V6R1 for iSeries
- DB2 UDB V5R3, V5R4 for iSeries

NOTE: This documentation uses the following terms to describe the different DB2 versions:

- "DB2 for Linux/UNIX/Windows" refers to all versions of DB2 for Linux, UNIX, and Windows
- "DB2 for z/OS" refers to all versions of DB2 for z/OS (formerly OS/390)
- "DB2 for XXX" refers to all versions of DB2 for iSeries (formerly iSeries)

The DB2 Wire Protocol driver is supported in the Windows, UNIX, and Linux environments. See ["Environment-Specific Information" on page 45](#) for detailed information about the environments supported by this driver.

Refer to the readme file shipped with your DataDirect Connect product for the file name of the DB2 Wire Protocol driver.

---

### Driver Requirements

The server requirement for all platforms is the same. The DB2 database can be installed as the Server Version or the Personal Edition.

The driver has no client requirements.

## Binding

The driver does not work properly unless bind packages exist on every server to which you intend to connect.

**IMPORTANT:** You must have the appropriate privileges for the driver to create and bind packages with your user ID. These privileges are:

- BINDADD for binding packages
- CREATEIN on the collection specified by the Package Collection option
- GRANT EXECUTE on the PUBLIC group for executing the packages

These are typically the permissions of a Database Administrator (DBA). If you do not have these privileges, someone who has a user ID with DBA privileges needs to create packages by connecting with the driver.

When connecting for the first time, the driver determines whether bind packages exist on the server. If packages do not exist, the driver creates them automatically using driver data source default values.

**NOTE:** The initial driver connection to a particular server may take a few minutes because of the number and size of the packages that must be created on the server. Subsequent connections do not incur this delay.

If you change default values in a data source before connecting with the driver for the first time, the new defaults are used when creating the packages. If you want to change these values after the packages have been created, you can create or modify packages from the Modify Bindings tab of the Setup dialog. See [Step 6](#) under "[Configuring and Connecting to Data Sources](#)" on page 132 for details.



On UNIX and Linux, you can also create or modify packages through a special bind utility. Depending on the platform of the DB2 server, the attribute values that must be set in the data source to bind packages are:

**Linux/UNIX/Windows DB2 Servers:** IpAddress, Database, TcpPort

**z/OS and iSeries DB2 Servers:** IpAddress, Location, TcpPort

Other attribute values also affect binding. See the note for [Step 6](#) under "[Configuring and Connecting to Data Sources](#)" on page 132 for details. See "[Connection Option Descriptions](#)" on page 150 for a description of these connection string attributes and their values. You must use the default values or specify new ones for these attributes in the DB2 data source section of the `odbc.ini` file before binding. See [Chapter 4 "Configuring the Product on UNIX/Linux"](#) on page 97 for details on creating the DB2 data source.

The bind utility is located in *installation\_directory/bin*. After specifying the appropriate connection string attribute values in the `odbc.ini` file, create or modify packages by entering the command:

```
bindxx dsn
```

where *xx* is the driver level number in the driver file name and *dsn* is the ODBC data source name in the `odbc.ini` file. For example:

```
bind27 DB2 Wire Protocol
```

You are prompted for a user ID and password if they are not stored in the data source. If packages are created and bound successfully, a message indicating success appears. If there are problems connecting or creating the packages, an appropriate error message appears.

## Creating DB2 Packages Using List Files

You can bind the following list files on your database server to create DB2 packages:

- DDODBC\_LUW.lst (DB2 for Linux/UNIX/Windows)
- DDODBC\_MVS.lst (DB2 for z/OS)
- DDODBC\_400.lst (DB2 for iSeries)

The list files are located in the \bind\LUW, \bind\zOS, and \bind\iSeries directories, respectively, in your DataDirect Connect Series *for* ODBC installation directory. When you bind the list files, if any DataDirect DB2 packages exist, they will be replaced by the new packages. The list files create DB2 packages that, by default, contain 200 dynamic sections and are created in the NULLID collection.

### To create DB2 packages by binding list files:

- 1 Copy the appropriate list (\*.lst) file and bind (\*.bnd) files located in the /bind directory to a directory on the database server.
- 2 From the database server directory where you placed the list and bind files, start the DB2 command-line utility. Use the utility to connect to the database where you want to bind the packages. Connect using the following command:

```
connect to database_name user authorization_name using password
```

where:

*database\_name* is the name of the database to which you are connecting.

*authorization\_name* is the name of the user you are authenticating to the server.

*password* is the user's password.

- 3 Execute the DB2 bind command:

```
bind @list_file grant public
```

where *list\_file* is the name of the list file you want to bind.

## Creating DB2 Packages Manually

On z/OS and iSeries servers, you can bind files manually to create DB2 packages. Refer to one of the following instruction files, as appropriate:

- CFODBC ZOS MANUAL PACKAGE CREATION INSTRUCTIONS.TXT (DB2 for z/OS)
- CFODBC AS400 MANUAL PACKAGE CREATION INSTRUCTIONS.TXT (DB2 for iSeries)

These instruction files are located in the bind\ZOS and \bind\ISERIES directories, respectively, in your DataDirect Connect Series *for* ODBC installation directory.

## Configuring and Connecting to Data Sources

After you install the driver, you configure data sources to connect to the database. See [Chapter 1 “Quick Start Connect” on page 35](#) for an explanation of different types of data sources. The data source contains connection options that allow you to tune the driver for specific performance. If you want to use a data source but need to change some of its values, you can either modify the data source or override its values at connection time through a connection string.

If you choose to use a connection string, you must use specific connection string attributes. See [“Using a Connection String” on page 148](#) and Table 5-1 on page 151 for an alphabetical list of driver connection string attributes and their initial default values.



### Data Source Configuration in the UNIX/Linux odbc.ini File

On UNIX and Linux, you must set up the proper ODBC environment before configuring data sources. See [“Environment Configuration” on page 37](#) for basic setup information and [“Environment Variables” on page 97](#) for more detail about this procedure.

Data sources for UNIX and Linux are stored in the system information file (by default, odbc.ini). If you have a Motif GUI environment on Linux, you can configure and modify data sources through the DataDirect ODBC Data Source Administrator for Linux (the Linux ODBC Administrator) using a driver Setup dialog box. (See [“Configuration Through the Administrator” on page 100](#) for a detailed explanation of the Administrator.)

If you do not have a GUI environment, you can configure and modify data sources directly by editing the odbc.ini file and storing default connection values there. See [“Configuration Through the System Information \(odbc.ini\) File” on page 103](#) for detailed information about the specific steps necessary to configure a data source.

Table 5-1 on page 151 lists driver connection string attributes that must be used in the odbc.ini file to set the value of the attributes. Note that only the long name of the attribute can be used in the file. The default listed in the table is the initial default value when the driver is installed.

### Data Source Configuration through a GUI



On Windows, data sources are stored in the Windows Registry. You can configure and modify data sources through the ODBC Administrator using a driver Setup dialog box, as described in this section.



On UNIX and Linux, data sources are stored in the odbc.ini file. On Linux, you can configure and modify data sources through the Linux ODBC Administrator using a driver Setup dialog box, as described in this section.

**NOTE:** This book shows dialog box images that are specific to Windows. If you are using the drivers in the Linux environment, the dialog box that you see may differ slightly from the Windows version. Windows-only and UNIX-only connection options are specifically noted by icons in the Setup dialog box descriptions.

When the driver is first installed, the values of its connection options are set by default. These values appear on the driver Setup dialog box tabs when you create a new data source. You can change these default values by modifying the data source. In the following procedure, the description of each tab is followed by a table that lists the connection options for that tab and their initial default values. This table links you to a complete description of the options and their connection string attribute equivalents. The connection string attributes are used to override the default values of the data source if you want to change these values at connection time.

### To configure a DB2 data source:

#### 1 Start the ODBC Administrator:



- On Windows, start the ODBC Administrator by selecting its icon from the DataDirect Connect program group.



- On Linux, change to the *install\_dir/tools* directory and, at a command prompt, enter:

```
odbcadmin
```

where *install\_dir* is the path to the product installation directory.

#### 2 Select a tab:

- **User DSN:** If you are configuring an existing user data source, select the data source name and click **Configure** to display the driver Setup dialog box.

If you are configuring a new user data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.



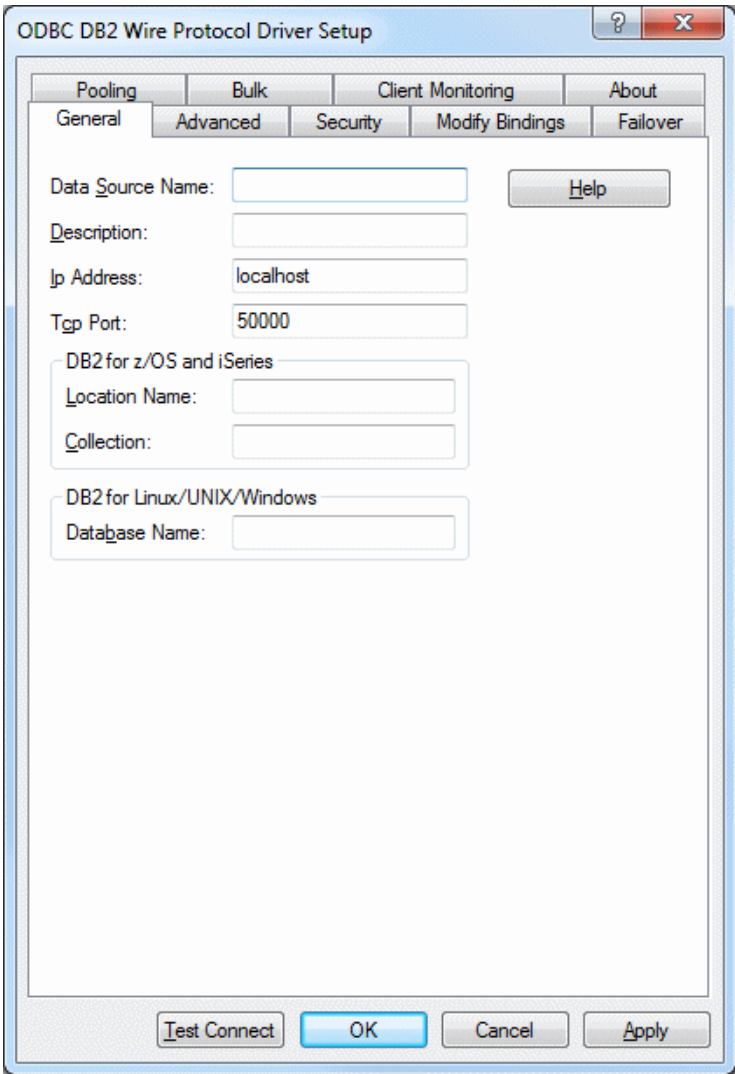
- **System DSN:** If you are configuring an existing system data source, select the data source name and click **Configure** to display the driver Setup dialog box.

If you are configuring a new system data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.

- **File DSN:** If you are configuring an existing file data source, select the data source file and click **Configure** to display the driver Setup dialog box.

If you are configuring a new file data source, click **Add** to display a list of installed drivers; then, select a driver. Click **Advanced** if you want to specify attributes; otherwise, click **Next** to proceed. Specify a name for the data source and click **Next**. Verify the data source information; then, click **Finish** to display the driver Setup dialog box.

The General tab of the Setup dialog box appears by default.

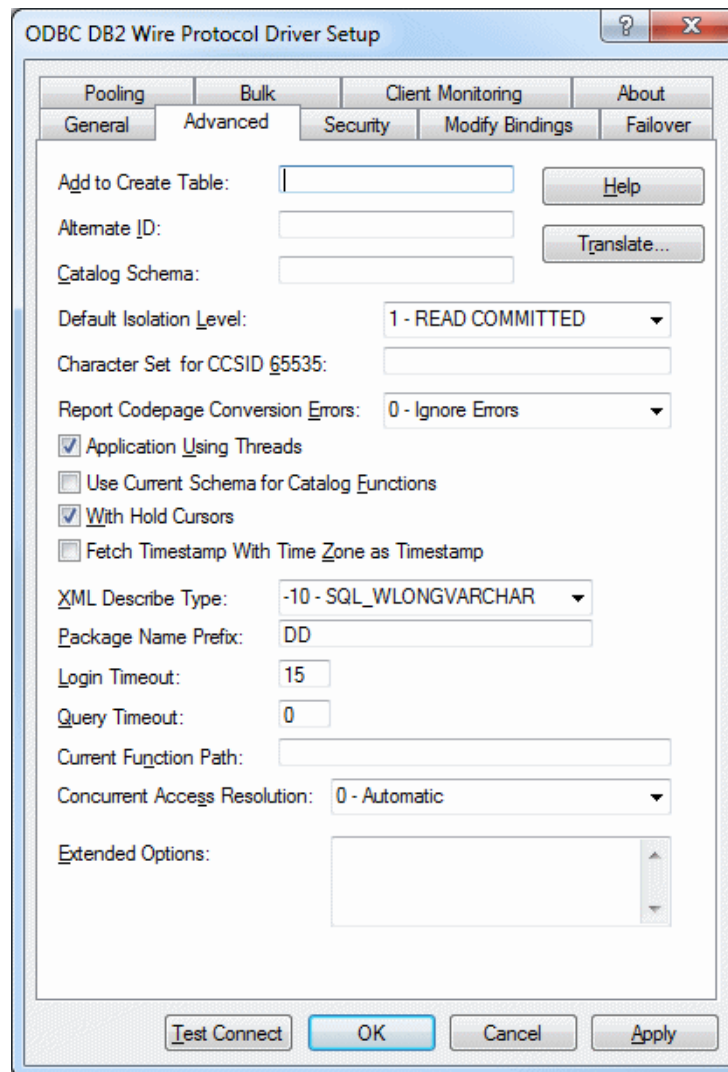


NOTE: The General tab displays only fields that are required for creating a data source. The fields on all other tabs are optional, unless noted otherwise.

- 3 On this tab, provide values for the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: General	Default
<a href="#">Data Source Name (see page 161)</a>	None
<a href="#">Description (see page 163)</a>	None
<a href="#">Ip Address (see page 169)</a>	localhost
<a href="#">Tcp Port (see page 176)</a>	50000
<a href="#">Location Name (see page 171)</a>	None
<a href="#">Collection (see page 158)</a>	None
<a href="#">Database Name (see page 161)</a>	None

- 4 Optionally, click the **Advanced** tab to specify additional data source settings.



On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Advanced	Default
<a href="#">Add to Create Table (see page 153)</a>	None
<a href="#">Alternate ID (see page 153)</a>	None
<a href="#">Catalog Schema (see page 157)</a>	None
<a href="#">Default Isolation Level (see page 162)</a>	1 - READ COMMITTED
<a href="#">Character Set for CCSID 65535 (see page 157)</a>	None
<a href="#">Report Codepage Conversion Errors (see page 176)</a>	0 - Ignore Errors
<a href="#">Application Using Threads (see page 154)</a>	Enabled
<a href="#">Use Current Schema for Catalog Functions (see page 178)</a>	Disabled
<a href="#">With Hold Cursors (see page 179)</a>	Enabled

Connection Options: Advanced	Default
<a href="#">Fetch Time Stamp With Time Zone as Timestamp (see page 166)</a>	Disabled
<a href="#">XML Describe Type (see page 179)</a>	-10 - SQL_WLONGVARCHAR
<a href="#">Package Name Prefix (see page 173)</a>	DD
<a href="#">Login Timeout (see page 172)</a>	15
<a href="#">Query Timeout (see page 175)</a>	0
<a href="#">Current Function Path (see page 161)</a>	None
<a href="#">Concurrent Access Resolution (see page 159)</a>	0 (Automatic)
<a href="#">IANAAppCodePage (see page 169)</a>	4 (ISO 8559-1 Latin 1)
UNIX ONLY	



**Extended Options:** Type a semi-colon separated list of connection options and their values. Use this configuration option to set the value of undocumented connection options that are provided by Progress DataDirect customer support. You can include any valid connection option in the Extended Options string, for example:

```
Database=Server1;UndocumentedOption1=value[;UndocumentedOption2=value;]
```

If the Extended Options string contains option values that are also set in the setup dialog or data source, the values of the options specified in the Extended Options string take precedence. However, connection options that are specified on a connection string override any option value specified in the Extended Options string.

NOTE: Do not specify the Extended Options configuration option in a connection string, or the driver will return an error. Instead, applications should specify the individual undocumented connection options in the connection string.



**Translate:** Click **Translate** to display the Select Translator dialog box, which lists the translators specified in the ODBC Translators section of the Registry. Progress 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.



- 5 Optionally, click the **Security** tab to specify security data source settings.

The screenshot shows the 'ODBC DB2 Wire Protocol Driver Setup' dialog box with the 'Security' tab selected. The dialog has several tabs: Pooling, Bulk, Client Monitoring, About, General, Advanced, Security, Modify Bindings, and Failover. The 'Security' tab contains two main sections: 'Authentication' and 'Encryption'. In the 'Authentication' section, there is a 'User Name' text box, an 'Authentication Method' dropdown menu set to '0 - No Encryption', and a 'GSS Client Library' text box set to 'native'. In the 'Encryption' section, there is an 'Encryption Method' dropdown menu set to '0 - No Encryption', a checked checkbox for 'Validate Server Certificate', and several text boxes for 'Trust Store', 'Trust Store Password', 'Key Store', 'Key Store Password', 'Key Password', and 'Host Name In Certificate'. At the bottom of the dialog are buttons for 'Test Connect', 'OK', 'Cancel', and 'Apply'.

See ["Using Security" on page 77](#) for a general description of authentication and encryption and their configuration requirements.

On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Security	Default
<a href="#">User Name (see page 178)</a>	None
<a href="#">Authentication Method (see page 155)</a>	0 (No Encryption)
<a href="#">GSS Client Library (see page 168)</a>	native
<a href="#">Encryption Method (see page 164)</a>	0 (No Encryption)
<a href="#">Validate Server Certificate (see page 178)</a>	Enabled
<a href="#">Truststore (see page 177)</a>	None
<a href="#">Truststore Password (see page 177)</a>	None
<a href="#">Keystore (see page 170)</a>	None

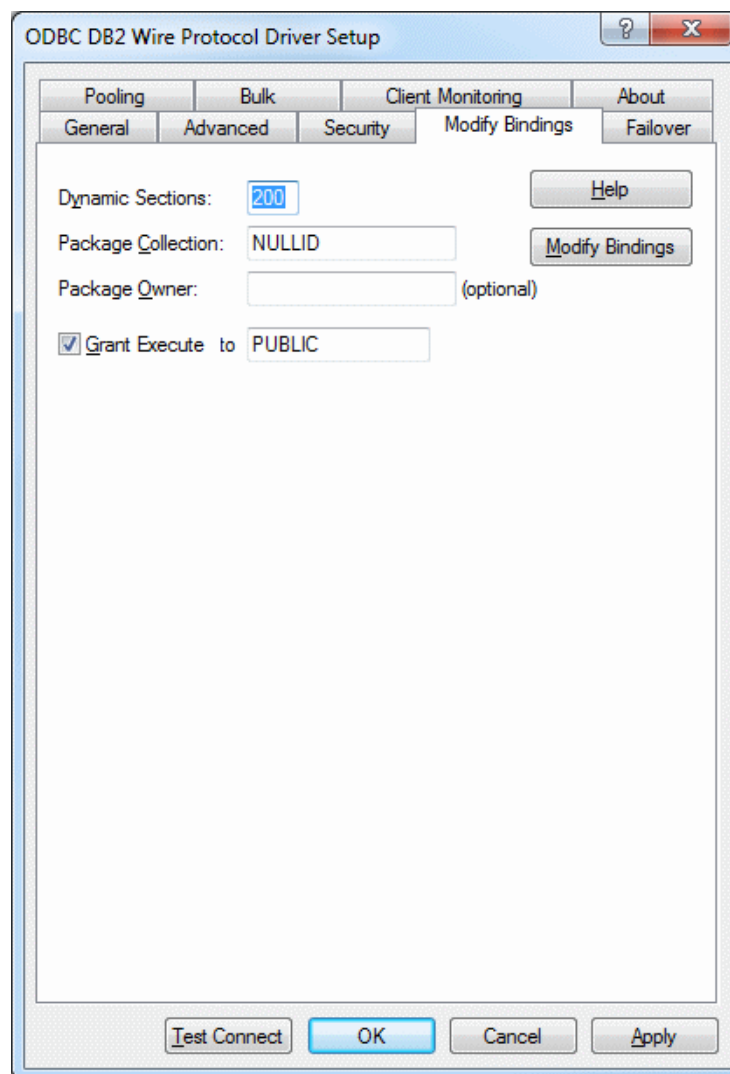
**Connection Options: Security**

Keystore Password (see page 170)  
 Key Password (see page 170)  
 Host Name In Certificate (see page 168)

**Default**

None  
 None  
 None

- 6 Optionally, click the **Modify Bindings** tab to configure options for creating or modifying bind packages.



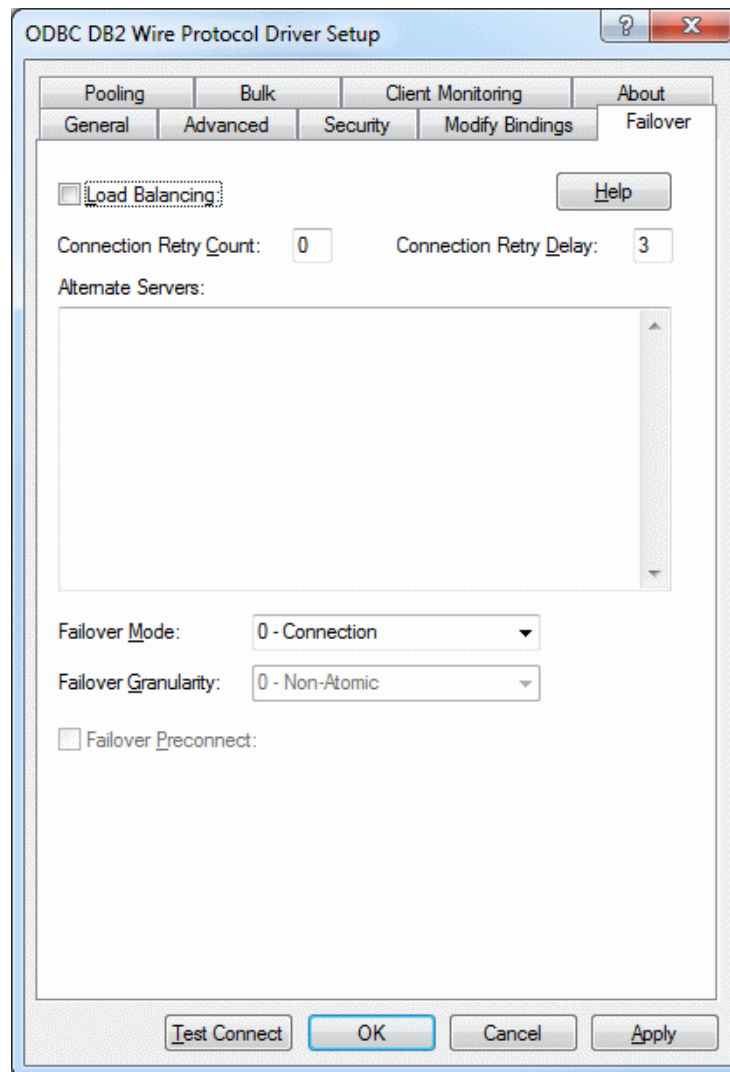
The Modify Bindings tab allows you to create or modify bind packages on the server accessed by the driver. If you connect with the driver before explicitly creating bind packages, the driver creates packages on the server automatically using the current values from the Setup dialog box. Alternatively, you can create a bind package before testing the connection. You can also modify bind packages after their creation from the Modify Bindings tab. You must also provide appropriate values for the options on the General tab.

**NOTE:** If you change any of the values on this tab after having initially created bind packages, you must rebind the packages. The changes are reflected only on new connections after rebinding.

On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Modify Bindings	Default
<a href="#">Dynamic Sections (see page 163)</a>	1000
<a href="#">Package Collection (see page 173)</a>	NULLID
<a href="#">Package Owner (see page 174)</a>	None
<a href="#">Grant Execute to [check box] (see page 167)</a>	Enabled
<a href="#">Grant Execute to [field] (see page 167)</a>	PUBLIC

- 7 Optionally, click the **Failover** tab to specify failover data source settings.

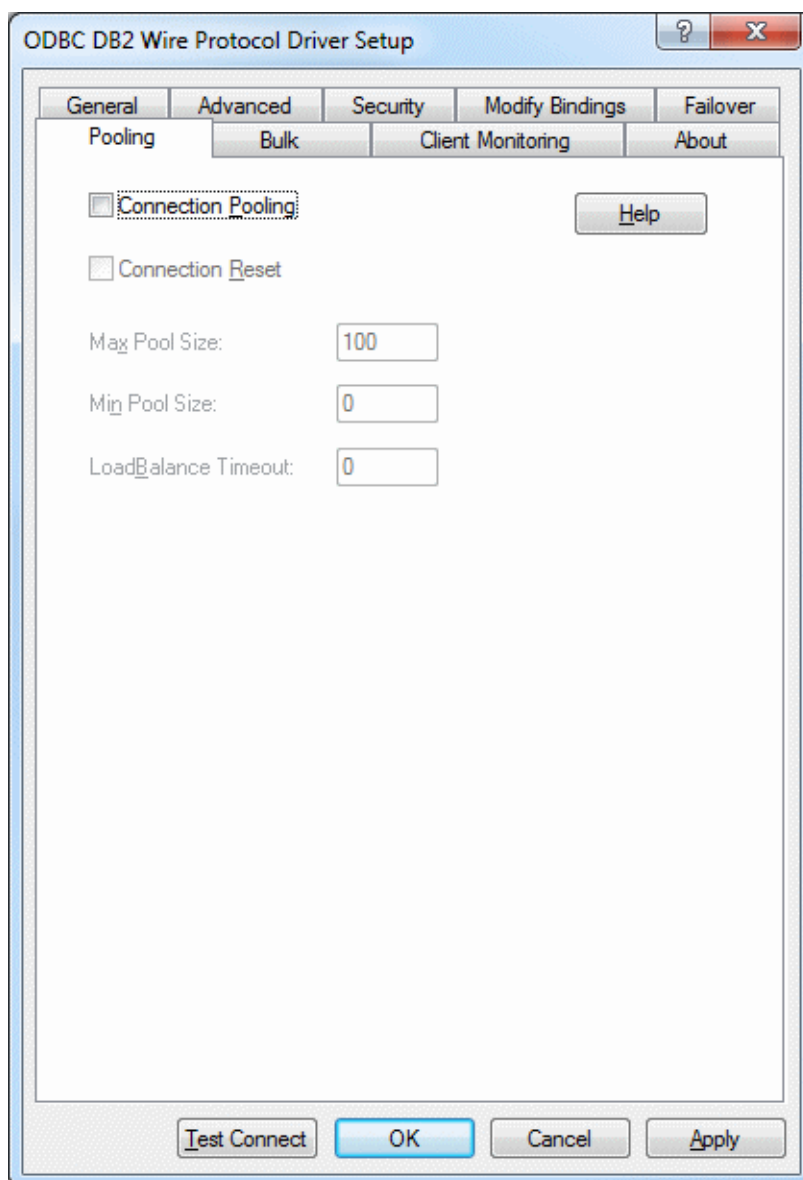


See ["Using Failover" on page 65](#) for a general description of failover and its related connection options.

On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Failover	Default
<a href="#">Load Balancing (see page 171)</a>	Disabled
<a href="#">Connection Retry Count (see page 160)</a>	0
<a href="#">Connection Retry Delay (see page 161)</a>	3
<a href="#">Alternate Servers (see page 153)</a>	None
<a href="#">Failover Mode (see page 165)</a>	0 (Connection)
<a href="#">Failover Granularity (see page 165)</a>	0 (Non-Atomic)
<a href="#">Failover Preconnect (see page 166)</a>	Disabled

- 8 Optionally, click the **Pooling** tab to specify connection pooling data source settings.

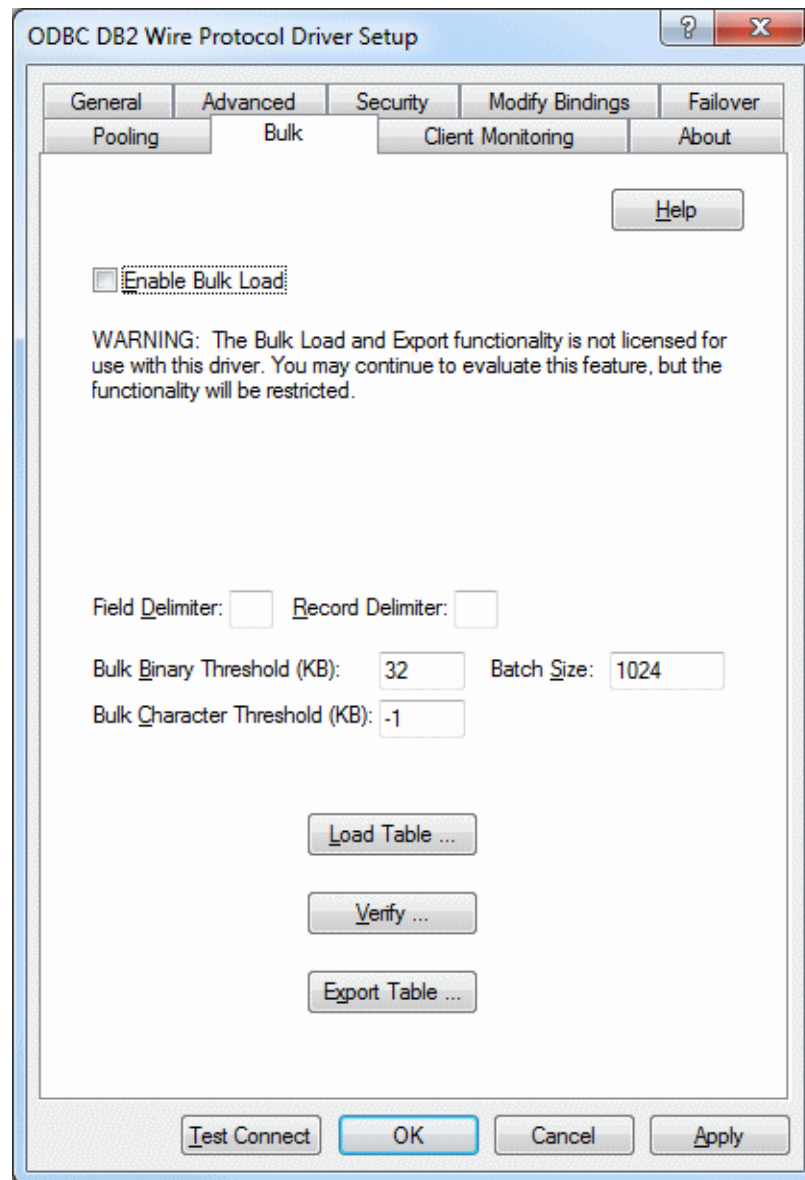


See ["Using DataDirect Connection Pooling" on page 82](#) for a general description of connection pooling.

On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Pooling	Default
<a href="#">Connection Pooling (see page 159)</a>	Disabled
<a href="#">Connection Reset (see page 160)</a>	Disabled
<a href="#">Max Pool Size (see page 172)</a>	100
<a href="#">Min Pool Size (see page 173)</a>	0
<a href="#">Load Balance Timeout (see page 171)</a>	0

- 9 Optionally, click the **Bulk** tab to specify DataDirect Bulk Load data source settings.



See ["Using DataDirect Bulk Load" on page 85](#) for a general description of DataDirect Bulk Load.

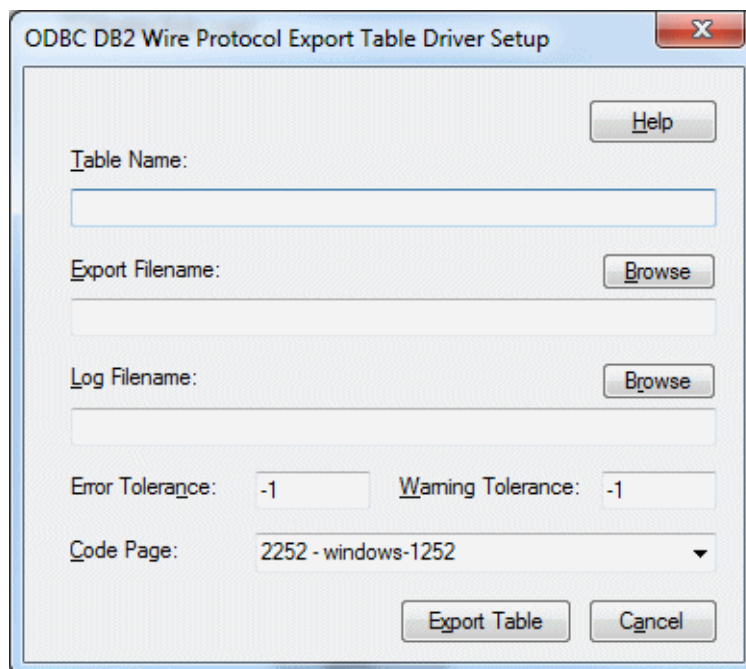
On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Bulk	Default
<a href="#">Enable Bulk Load (see page 164)</a>	Disabled
<a href="#">Field Delimiter (see page 167)</a>	None
<a href="#">Record Delimiter (see page 176)</a>	None
<a href="#">Bulk Binary Threshold (see page 156)</a>	32
<a href="#">Bulk Character Threshold (see page 156)</a>	-1
<a href="#">Batch Size (see page 155)</a>	1024

If your application is already coded to use parameter array batch functionality, you can leverage DataDirect Bulk Load features through the Enable Bulk Load connection option. Enabling this option automatically converts the parameter array batch operation to use the database bulk load protocol.

If you are not using parameter array batch functionality, you can export data to a bulk load data file, verify the metadata of the bulk load configuration file against the structure of the target table, and bulk load data to a table. Use the following steps to accomplish these tasks.

- a To export data from a table to a bulk load data file, click **Export Table** from the Bulk tab. The Export Table dialog box appears.



Both a bulk data file and a bulk configuration file are produced by exporting a table. The configuration file has the same name as the data file, but with an XML extension. See ["Using DataDirect Bulk Load" on page 85](#) for details about these files.

The bulk export operation can create a log file and can also export to external files. See ["External Overflow Files" on page 95](#) for more information. The export operation can be configured such that if any errors or warnings occur:

- The operation always completes.
- The operation always terminates.
- The operation terminates after a certain threshold of warnings or errors is exceeded.

**Table Name:** A string that specifies the name of the source database table containing the data to be exported.

**Export Filename:** A string that specifies the path (relative or absolute) and file of the bulk load data file to which the data is to be exported. It also specifies the file name of the bulk configuration file. These files must not already exist; if one of both of them already exists, an error is returned.

**Log Filename:** A string that specifies the path (relative or absolute) and file name of the bulk log file. The log file is created if it does not exist. Events logged to this file are:

- Total number of rows fetched
- A message for each row that failed to export
- Total number of rows that failed to export
- Total number of rows successfully exported

Information about the load is written to this file, preceded by a header. Information about the next load is appended to the end of the file.

If you do not supply a value for Log Filename, no log file is created.

**Error Tolerance:** A value that specifies the number of errors to tolerate before an operation terminates. A value of 0 indicates that no errors are tolerated; the operation fails when the first error is encountered.

The default of -1 means that an infinite number of errors is tolerated.

**Warning Tolerance:** A value that specifies the number of warnings to tolerate before an operation terminates. A value of 0 indicates that no warnings are tolerated; the operation fails when the first warning is encountered.

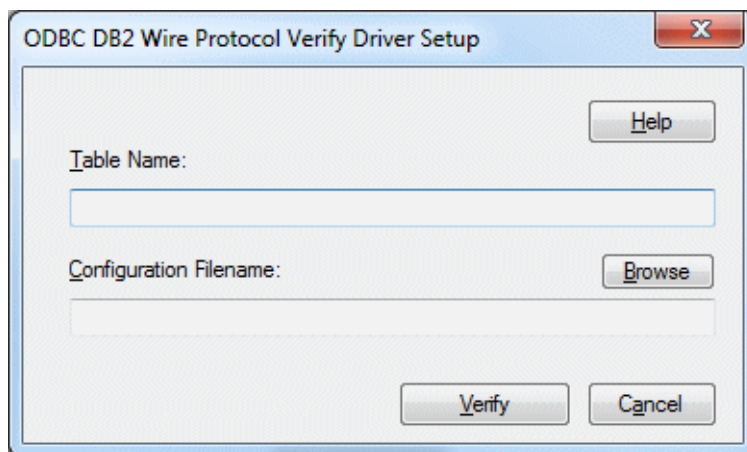
The default of -1 means that an infinite number of warnings is tolerated.

**Code Page:** A value that specifies the code page value to which the driver must convert all data for storage in the bulk data file. See ["Character Set Conversions" on page 94](#) for more information.

The default value on Windows is the current code page of the machine. On UNIX/Linux, the default value is 4 (ISO 8559-1 Latin-1).

Click **Export Table** to connect to the database and export data to the bulk data file or click **Cancel**.

To verify the metadata of the bulk load configuration file against the structure of the target database table, click **Verify** from the Bulk tab. See ["Verification of the Bulk Load Configuration File" on page 92](#) for details. The Verify dialog box appears.



**Table Name:** A string that specifies the name of the target database table into which the data is to be loaded.

**Configuration Filename:** A string that specifies the path (relative or absolute) and file name of the bulk configuration file.

Click **Verify** to verify table structure or click **Cancel**.



- b To bulk load data from the bulk data file to a database table, click **Load Table** from the Bulk tab. The Load File dialog box appears.

The load operation can create a log file and can also create a discard file that contains rows rejected during the load. The discard file is in the same format as the bulk load data file. After fixing reported issues in the discard file, the bulk load can be reissued using the discard file as the bulk load data file.

The export operation can be configured such that if any errors or warnings occur:

- The operation always completes.
- The operation always terminates.
- The operation terminates after a certain threshold of warnings or errors is exceeded.

If a load fails, the Load Start and Load Count options can be used to control which rows are loaded when a load is restarted after a failure.

**Table Name:** A string that specifies the name of the target database table into which the data is loaded.

**Load Data Filename:** A string that specifies the path (relative or absolute) and file name of the bulk data file from which the data is loaded.

**Configuration Filename:** A string that specifies the path (relative or absolute) and file name of the bulk configuration file.

**Log Filename:** A string that specifies the path (relative or absolute) and file name of the bulk log file. Specifying a value for Log Filename creates the file if it does not already exist. Events logged to this file are:

- Total number of rows read
- Message for each row that failed to load
- Total number of rows that failed to load
- Total number of rows successfully loaded

Information about the load is written to this file, preceded by a header. Information about the next load is appended to the end of the file.

If you do not specify a value for Log Filename, no log file is created.

**Discard Filename:** A string that specifies the path (relative or absolute) and file name of the bulk discard file. Any row that cannot be inserted into database as result of bulk load is added to this file, with the last row rejected added to the end of the file.

Information about the load is written to this file, preceded by a header. Information about the next load is appended to the end of the file.

If you do not specify a value for Discard Filename, a discard file is not created.

**Error Tolerance:** A value that specifies the number of errors to tolerate before an operation terminates. A value of 0 indicates that no errors are tolerated; the operation fails when the first error is encountered.

The default of -1 means that an infinite number of errors is tolerated.

**Load Start:** A value that specifies the first row to be loaded from the data file. Rows are numbered starting with 1. For example, when Load Start is 10, the first 9 rows of the file are skipped and the first row loaded is row 10. This option can be used to restart a load after a failure.

The default value is 1.

**Read Buffer Size (KB):** A value that specifies the size, in KB, of the buffer that is used to read the bulk data file for a bulk load operation.

The default value is 2048.

**Warning Tolerance:** A value that specifies the number of warnings to tolerate before an operation terminates. A value of 0 indicates that no warnings are tolerated; the operation fails when the first warning is encountered.

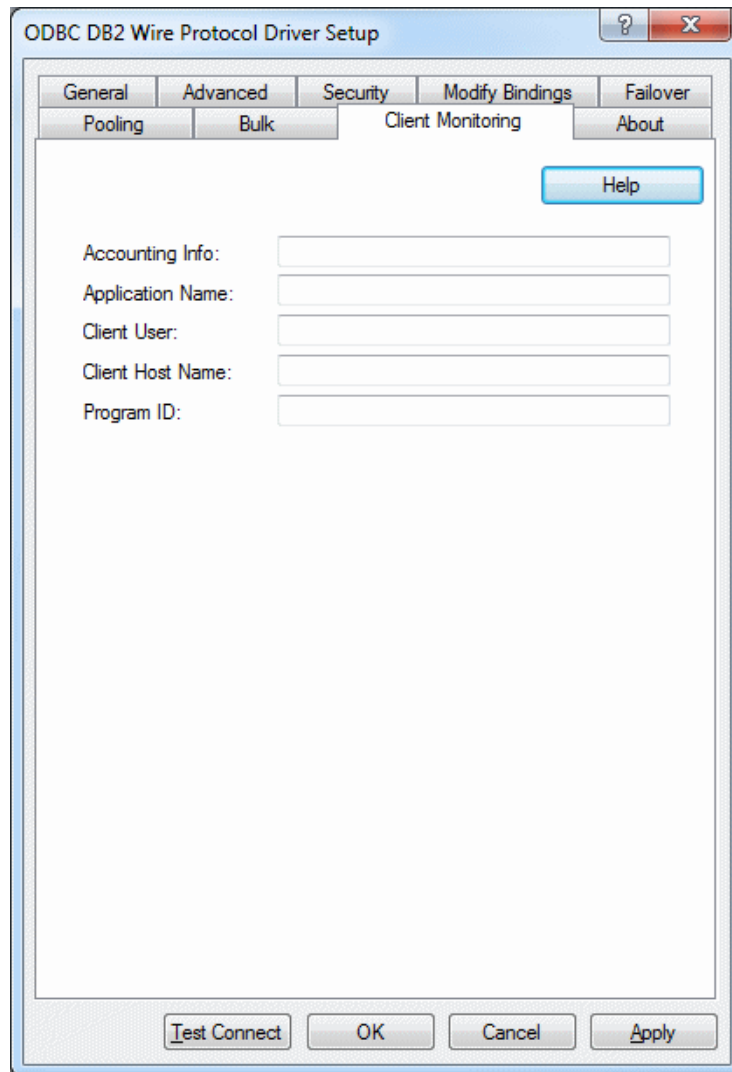
The default of -1 means that an infinite number of warnings is tolerated.

**Load Count:** A value that specifies the number of rows to be loaded from the data file. The bulk load operation loads rows up to the value of Load Count from the file to the database. It is valid for Load Count to specify more rows than exist in the data file. The bulk load operation completes successfully when either the number of rows specified by the Load Count value has been loaded or the end of the data file is reached. This option can be used in conjunction with Load Start to restart a load after a failure.

The default value is the maximum value for SQLULEN. If set to 0, no rows are loaded.

Click **Load Table** to connect to the database and load the table or click **Cancel**.

Optionally, click the **Client Monitoring** tab to specify additional data source settings.



See ["Using Client Information" on page 75](#) for additional information about client monitoring.

On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Client Monitoring	Default
<a href="#">Accounting Info (see page 152)</a>	None
<a href="#">Application Name (see page 154)</a>	None
<a href="#">Client User (see page 158)</a>	None

Connection Options: Client Monitoring	Default
<a href="#">Client Host Name (see page 158)</a>	None
<a href="#">Program ID (see page 175)</a>	None

- 10** At any point during the configuration process, you can click **Test Connect** to attempt to connect to the data source using the connection options specified in the driver Setup dialog box. A logon dialog box appears (see ["Using a Logon Dialog Box" on page 149](#) for details). 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 incorrect environment or connection value, it displays an appropriate error message. Click **OK**.

**IMPORTANT:** If you have not already created bind packages by clicking the Modify Bindings button on the Modify Bindings tab, the initial connection through the Test Connect button may take a few minutes because of the number and size of the packages that must be created on the server. Subsequent connections occur without this delay.

**NOTE:** If you are configuring alternate servers for use with the connection failover feature, be aware that the Test Connect button tests only the primary server, not the alternate servers.

- 11** 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.

## Using a Connection String

If you want to use a connection string for connecting to a database, or if your application requires it, you must specify either a DSN (data source name), a File DSN, or a DSN-less connection in the string. The difference is whether you use the DSN=, FILEDSN=, or the DRIVER= keyword in the connection string, as described in the ODBC specification. A DSN or FILEDSN connection string tells the driver where to find the default connection information. Optionally, you may specify *attribute=value* pairs in the connection string to override the default values stored in the data source.

The DSN connection string has the form:

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

The FILEDSN connection string has the form:

```
FILEDSN=filename.dsn[;attribute=value[;attribute=value]...]
```

The DSN-less connection string specifies a driver instead of a data source. All connection information must be entered in the connection string because the information is not stored in a data source.

The DSN-less connection string has the form:

```
DRIVER={[driver_name]}[;attribute=value[;attribute=value]...]
```

[Table 5-1](#) lists the long and short names for each attribute, as well as the initial default value when the driver is first installed. You can specify either long or short names in the connection string.

An example of a DSN connection string with overriding attribute values for DB2 for Linux/UNIX/Windows is:

```
DSN=DB2ACCOUNT;DB=DB2DATA;UID=JOHN;PWD=XYZZY
```

A FILEDSN connection string is similar except for the initial keyword:

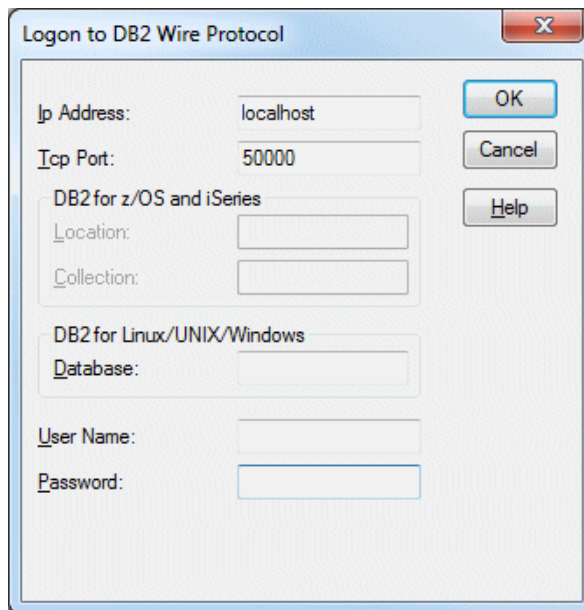
```
FILEDSN=DB2.dsn;DB=DB2DATA;UID=JOHN;PWD=XYZZY
```

A DSN-less connection string must provide all necessary connection information:

```
DRIVER=DataDirect 7.1 DB2 Wire Protocol;IpAddress=123.456.78.90;  
PORT=5179;DB=DB2DATA;UID=JOHN;PWD=XYZZY
```

## 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, provide the following information:

- 1 In the Ip Address field, type the IP (Internet Protocol) 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. The default is localhost.

The IP address can be specified in either IPv4 or IPv6 format, or a combination of the two. See ["Using IP Addresses" on page 54](#) for details concerning these formats.

- 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 (50000 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 for z/OS or iSeries, perform Steps 3a and 3b. Otherwise, skip to Step 4.
  - a In the Location field, type the DB2 location name. Use the name defined during the local DB2 installation.
  - b By default, the User ID is used for the value of Collection. The User ID must always be used on DB2 for z/OS.  
  
 On iSeries, you can type the name of the schema that is to be the default qualifier for unqualified object names. If you want to access a table outside of this schema, you need to specify the appropriate two-part name, for example, `SELECT * FROM Schema.Tablename`. On iSeries only, Collection is also the current library.  
  
 Skip to Step 5.
- 4 If you are running DB2 for Linux/UNIX/Windows, type the name of the database to which you want to connect in the Database field.
- 5 If required, type your logon ID in the User Name field.
- 6 If required, type your password in the Password field.
- 7 Click **OK** to complete the logon and to update the values in the Registry.

---

## Connection Option Descriptions

The following connection option descriptions are listed alphabetically by the GUI name that appears on the driver Setup dialog box. The connection string attribute name, along with its short name, is listed immediately underneath the GUI name. For example:

### Application Using Threads

Attribute    ApplicationUsingThreads (AUT)

In most cases, the GUI name and the attribute name are the same; however, some exceptions exist. If you need to look up an option by its connection string attribute name, please refer to the alphabetical table of connection string attribute names.

Also, a few connection string attributes, for example, Password, do not have equivalent options that appear on the GUI. They are in the list of descriptions alphabetically by their attribute names.

[Table 5-1](#) lists the connection string attributes supported by the DB2 Wire Protocol driver.

**Table 5-1. DB2 Wire Protocol Attribute Names**

<b>Attribute (Short Name)</b>	<b>Default</b>
AccountingInfo (AI)	None
AddStringToCreateTable (ASCT)	None
AlternateID (AID)	None
AlternateServers (ASRV)	None
ApplicationName (AN)	None
ApplicationUsingThreads (AUT)	1 (Enabled)
AuthenticationMethod (AM)	0 (No Encryption)
BulkBinaryThreshold (BBT)	32
BulkCharacterThreshold (BCT)	-1
BulkLoadBatchSize (BLBS)	1024
BulkLoadFieldDelimiter (BLFD)	None
BulkLoadRecordDelimiter (BLRD)	None
CatalogSchema (CS)	None
CharsetFor65535 (CF6)	0
ClientHostName (CHN)	None
ClientUser (CU)	None
Collection (COL)	None
ConcurrentAccessResolution (CAR)	0 (Automatic)
ConnectionReset (CR)	0 (Disabled)
ConnectionRetryCount (CRC)	0
ConnectionRetryDelay (CRD)	3
CurrentFunctionPath (CFP)	None
Database (DB)	None
DataSourceName (DSN)	None
DefaultIsolationLevel (DIL)	1 (READ_COMMITTED)
Description (n/a)	None
DynamicSections (DS)	1000
EnableBulkLoad (EBL)	0 (Disabled)
EncryptionMethod (EM)	0 (No Encryption)
FailoverGranularity (FG)	0 (Non-Atomic)
FailoverMode (FM)	0 (Connection)
FailoverPreconnect (FP)	0 (Disabled)
FetchTSWTZasTimestamp	0 (Disabled)
GrantAuthid (GA)	PUBLIC
GrantExecute (GE)	1 (Enabled)
GSSClient (GSSC)	native
HostNameInCertificate (HNIC)	None

**Table 5-1. DB2 Wire Protocol Attribute Names** (cont.)

Attribute (Short Name)	Default
IANAAppCodePage (IACP) UNIX ONLY	4 (ISO 8559-1 Latin-1)
IpAddress (IP)	localhost
KeyPassword (KP)	None
Keystore (KS)	None
KeystorePassword (KSP)	None
LoadBalanceTimeout (LBT)	0
LoadBalancing (LB)	0 (Disabled)
Location (LOC)	None
LoginTimeout (LT)	15
LogonID (UID)	None
MaxPoolSize (MXPS)	100
MinPoolSize (MNPS)	0
PackageCollection (PC)	NULLID
PackageNamePrefix (PNP)	DD
PackageOwner (PO)	None
Password (PWD)	None
Pooling (POOL)	0 (disabled)
ProgramID (PID)	None
QueryTimeout (QT)	0
ReportCodepageConversionErrors (RCCE)	0 (Ignore Errors)
TcpPort (PORT)	50000
Truststore (TS)	None
TruststorePassword (TSP)	None
UseCurrentSchema (UCS)	0 (disabled)
ValidateServerCertificate (VSC)	1 (enabled)
WithHold (WH)	1 (enabled)
XMLDescribeType (XDT)	-10

## Accounting Info

Attribute	AccountingInfo (AI)
Description	Specifies accounting information to be stored in the database. This value sets the CURRENT CLIENT_ACCTNG register (DB2 for Linux/UNIX/Windows) or the CLIENT ACCTNG register (DB2 for z/OS) in the database. This value is used by the DB2 Workload Manager.

This connection option can affect performance. See ["Performance Considerations" on page 179](#) for details.



Valid Values	<i>string</i> where <i>string</i> is the accounting information.
Default	None
GUI Tab	<a href="#">Client Monitoring tab</a> on page 147

## Add to Create Table

Attribute	AddStringToCreateTable (ASCT)
Description	Specifies a string that is automatically added to all Create Table statements. This option is for users who need to add an In Database clause to Create Table statements.
Valid Values	<i>string</i> where <i>string</i> is valid syntax for the In Database clause of a Create Table statement.
Default	None
GUI Tab	<a href="#">Advanced tab</a> on page 135

## Alternate ID

Attribute	AlternateID (AID)
Description	Specifies the name of the default schema that is used to qualify unqualified database objects in dynamically prepared SQL statements. If the attempt to change the current schema fails, the connection fails and you receive the message <i>Invalid value for Alternate ID</i> . DB2 permissions must be set to SYSADM.
Valid Values	<i>string</i> where <i>string</i> is a valid DB2 schema name.
Default	None
GUI Tab	<a href="#">Advanced tab</a> on page 135

## Alternate Servers

Attribute	AlternateServers (ASRV)
Description	A list of alternate database servers to which the driver tries to connect if the primary database server is unavailable. Specifying a value for this option enables connection failover for the driver. The value you specify must be in the form of a string that defines the physical location of each alternate server. All of the other required connection information for each alternate server is the same as what is defined for the primary server connection.
Valid Values	<p>(IPAddress=<i>ipvalue</i>:TcpPort=<i>portvalue</i>:{Database   Location}=<i>databasevalue</i>[, . . .])</p> <p>You must specify the IP address, port number, and database name (Linux/UNIX/Windows) or location (z/OS and iSeries) of each alternate server.</p> <p>NOTE: An alternate server address in IPv6 format must be enclosed in double quotation marks.</p>

**Example** The following Alternate Servers values define two alternate database servers for connection failover:

```
AlternateServers=(IpAddress=123.456.78.90:TcpPort=5177:Database=DB2DAT,  
IpAddress=223.456.78.90:TcpPort=5178:Database=DB2DAT3)
```

or

```
AlternateServers=(IpAddress=123.456.78.90:  
TcpPort=5177:Location=DB2DAT, IpAddress=223.456.78.90:TcpPort=5178:  
Location=DB2DAT3)
```

**Default** None

**GUI Tab** [Failover tab](#) on page 139

## Application Name

**Attribute** ApplicationName (AN)

**Description** Specifies the name of the application to be stored in the database. This value sets the CURRENT\_CLIENT\_APPLNAME register (DB2 for Linux/UNIX/Windows) or CLIENT\_APPLNAME register (DB2 for z/OS) in the database. For DB2 V9.1 and higher for Linux/UNIX/Windows, this value also sets the APPL\_NAME value of the SYSIBMADM.APPLICATIONS table. This value is used by the DB2 Workload Manager.

This connection option can affect performance. See ["Performance Considerations" on page 179](#) for details.

**Valid Values** *string*

where *string* is the name of the application.

**Default** None

**GUI Tab** [Client Monitoring tab](#) on page 147

## Application Using Threads

**Attribute** ApplicationUsingThreads (AUT)

**Description** Determines whether the driver works with applications using multiple ODBC threads.

This connection option can affect performance. See ["Performance Considerations" on page 179](#) for details.

**Valid Values** 0 | 1

If set to 1 (Enabled), the driver works with single-threaded and multi-threaded applications.

If set to 0 (Disabled), the driver does not work with multi-threaded applications. If using the driver with single-threaded applications, this value avoids additional processing required for ODBC thread-safety standards.

**Default** 1 (Enabled)

**GUI Tab** [Advanced tab](#) on page 135

## Authentication Method

Attribute	AuthenticationMethod (AM)
Description	Specifies the method the driver uses to authenticate the user to the server when a connection is established. If the specified authentication method is not supported by the database server, the connection fails and the driver generates an error.
Valid Values	0   1   2   3   4   7   8  <p>If set to 0 (No Encryption), the driver sends the user ID and password in clear text to the server for authentication.</p> <p>If set to 1 (Encrypt Password), the driver sends the user ID in clear text and an encrypted password to the server for authentication.</p> <p>If set to 2 (Encrypt UID and Password), the driver sends an encrypted user ID and password to the server for authentication.</p> <p>If set to 3 (Client Authentication), the driver uses client authentication when establishing a connection. The database server relies on the client to authenticate the user and does not provide additional authentication.</p> <p>If set to 4 (Kerberos Authentication), the driver uses Kerberos authentication. This method supports both Windows Active Directory Kerberos and MIT Kerberos environments.</p> <p>If set to 7 (Encrypted Password AES), the driver encrypts the password with 256-bit AES encryption in the connection request. (DB2 V9.7 and higher only.)</p> <p>If set to 8 (Encrypted UID and Password AES), the driver encrypts the user id and password with 256-bit AES encryption in the connection request. (DB2 V9.7 and higher only.)</p> <p>NOTE: The use of AES encryption (values 7 and 8) requires that the DataDirect OpenSSL library be installed.</p>
Default	0 (No Encryption)
GUI Tab	<a href="#">Security tab</a> on page 137

## Batch Size

Attribute	BulkLoadBatchSize (BLBS)
Description	The number of rows that the driver sends to the database at a time during bulk operations. This value applies to all methods of bulk loading.
Valid Values	$x$  <p>where <math>x</math> is a positive integer that specifies the number of rows to be sent.</p>
Default	1024
GUI Tab	<a href="#">Bulk tab</a> on page 141

## Bulk Binary Threshold

Attribute	BulkBinaryThreshold (BBT)
Description	The maximum size, in KB, of binary data that is exported to the bulk data file.
Valid Values	-1   0   $x$  where $x$ is an integer that specifies the number of KB.  If set to -1, all binary data, regardless of size, is written to the bulk data file, not to an external file.  If set to 0, all binary data, regardless of size, is written to an external file, not the bulk data file. A reference to the external file is written to the bulk data file.  If set to $x$ , any binary data exceeding this specified number of KB is written to an external file, not the bulk data file. A reference to the external file is written to the bulk data file.
Default	32
GUI Tab	<a href="#">Bulk tab</a> on page 141

## Bulk Character Threshold

Attribute	BulkCharacterThreshold (BCT)
Description	The maximum size, in KB, of character data that is exported to the bulk data file.
Valid Values	-1   0   $x$  where $x$ is an integer that specifies the number of KB.  If set to -1, all character data, regardless of size, is written to the bulk data file, not to an external file.  If set to 0, all character data regardless of size, is written to an external file, not the bulk data file. A reference to the external file is written to the bulk data file.  If set to $x$ , any character data exceeding this specified number of KB is written to an external file, not the bulk data file. A reference to the external file is written to the bulk data file.
Default	-1
GUI Tab	<a href="#">Bulk tab</a> on page 141

## Catalog Schema

Attribute	CatalogSchema (CS)
Description	Specifies the DB2 schema to use for Catalog functions. Specifying a schema allows you to use copies or views of the system catalog tables for catalog functions.
Valid Values	<p><i>schema_name</i></p> <p>where <i>schema_name</i> is the name of a valid DB2 schema. If you do not specify a value for this attribute, the driver uses SYSIBM when connected to z/OS, QSYS2 when connected to iSeries, and SYSCAT when connected to Linux/UNIX/Windows.</p> <p><b>Example</b> Create a view DB2ADMIN.TABLES from SYSCAT.TABLES.</p> <pre>CREATE VIEW DB2ADMIN.TABLES AS SELECT * FROM SYSCAT.TABLES WHERE OWNER LIKE 'ODBC%'</pre> <p>Set CatalogSchema=DB2ADMIN, and do the SQLTables thing.</p> <pre>"TABLE_CAT", "TABLE_SCHEM", "TABLE_NAME", "TABLE_TYPE", "REMARKS"</pre> <p>The results come from the DB2ADMIN.TABLES view. Three rows are fetched from five columns.</p> <pre>&lt;Null&gt;, "DB2ADMIN", "BUG", "TABLE", &lt;Null&gt; &lt;Null&gt;, "DB2ADMIN", "DATETEST", "TABLE", &lt;Null&gt; &lt;Null&gt;, "DB2ADMIN", "TESTCP", "TABLE", &lt;Null&gt;</pre>
Default	None
GUI Tab	<a href="#">Advanced tab</a> on page 135

## Character Set for CCSID 65535

Attribute	CharsetFor65535 (CF6)
Description	Specifies the IANA code page to be used by the driver to convert character data stored as bit data in character columns (Char, Varchar, Longvarchar, Clob, Char for Bit Data, Varchar for Bit Data, Longvarchar for Bit Data) defined with CCSID 65535.
Valid Values	<p>0   <i>IANA_code_page</i></p> <p>where <i>IANA_code_page</i> is a valid IANA code page. Refer to <a href="#">"IBM to IANA Code Page Values"</a> in <a href="#">Chapter 1</a> of the <i>DataDirect Connect Series for ODBC Reference</i> for a list of the most commonly used IBM code pages and their IANA code page equivalents.</p> <p>If unspecified or set to 0, the driver returns these columns as binary columns (SQL_BINARY, SQL_VARBINARY, SQL_LONGVARBINARY) and does no conversion of the data.</p> <p>If an IANA code page is specified, the driver returns these columns as character columns in the character set specified. The driver does no conversion of data supplied in bound parameters.</p>
Default	0
GUI Tab	<a href="#">Advanced tab</a> on page 135

## Client Host Name

Attribute	ClientHostName (CHN)
Description	<p>Specifies the host name of the client machine to be stored in the database. This value sets the CURRENT CLIENT_WRKSTNNAME register (DB2 for Linux/UNIX/Windows) or CLIENT WRKSTNNAME register (DB2 for z/OS) in the database. This value is used by the DB2 Workload Manager.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 179</a> for details.</p>
Valid Values	<p><i>string</i></p> <p>where <i>string</i> is the host name of the client machine.</p>
Default	None
GUI Tab	<a href="#">Client Monitoring tab</a> on page 147

## Client User

Attribute	ClientUser (CU)
Description	<p>The user ID to be stored in the database. This option sets the CURRENT CLIENT_USERID register (DB2 for Linux/UNIX/Windows) and CLIENT USERID register (DB2 for z/OS) in the database. This value is used by the DB2 Workload Manager.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 179</a> for details.</p>
Valid Values	<p>-1   <i>client_userid</i></p> <p>where <i>client_userid</i> is a valid user ID.</p> <p>The value -1 means that the driver uses the userid of the user that is currently logged onto the client.</p>
Default	None
GUI Tab	<a href="#">Client Monitoring tab</a> on page 147

## Collection

Attribute	Collection (COL)
Description	<p>The current collection or library. Valid only on DB2 for z/OS and iSeries.</p> <p>For DB2 for z/OS, this value is the user ID. If an attempt to change the current schema fails, the connection fails and you receive the message <code>Invalid value for Alternate ID</code>.</p> <p>For DB2 for iSeries, this value is the name of the schema to be used as the default qualifier for unqualified object names. If you want to access a table outside of this schema, you must specify the schema name and the table name. For example:</p> <pre>SELECT * FROM Schema.Tablename</pre> <p>Also, if the Alternate ID option is set, it overrides this option.</p>

NOTE: This option is mutually exclusive with the Database Name option.

Valid Values *user\_ID* (DB2 for z/OS) | *schema\_name* (DB2 for iSeries)

where

*user\_ID* is a valid user ID. DB2 permissions on the user ID must be set to SYSADM.

*schema\_name* is the default schema to use for unqualified object names.

Default None

GUI Tab [General tab](#) on page 134

## Concurrent Access Resolution

Attribute ConcurrentAccessResolution (CAR)

Description Specifies whether a read-only query can access the currently committed value of rows that are locked by a transaction that is updating the rows. The driver must be connected to DB2 V9.7 for LUW or higher and the application isolation level must be either read committed or repeatable read.

Valid Values 0 | 1 | 2

If set to 0 (Automatic), the driver persists the server behavior, as specified by the *cur\_commit* server parameter. If *cur\_commit* is set to "Available" or "Disable," then the current behavior, pending until the row lock is released, is used. When *cur\_commit* is set to "On," the driver returns the last committed value of the row, regardless of whether the row is locked.

If set to 1 (Wait For Outcome), the driver always waits for the transaction to be completed before returning a row of data that has been locked by another transaction, regardless of how the *cur\_commit* parameter is configured on the server.

If set to 2 (Use Currently Committed), the driver returns the value that was committed during the last transaction if the *cur\_commit* parameter is configured to "On" or "Available," even though the row is locked.

NOTE: This option is ignored when connecting to a DB2 server version earlier than DB2 V9.7 for LUW.

Default 0 (Automatic)

GUI Tab [Advanced tab](#) on page 135

## Connection Pooling

Attribute Pooling (POOL)

Description Specifies whether to use the driver's connection pooling.

NOTE: The application must be thread-enabled to use connection pooling.

This connection option can affect performance. See ["Performance Considerations" on page 179](#) for details.

Valid Values 0 | 1

If set to 1 (Enabled), the driver uses connection pooling.

If set to 0 (Disabled), the driver does not use connection pooling.

Default 0 (Disabled)

GUI Tab [Pooling tab](#) on page 140

## Connection Reset

Attribute ConnectionReset (CR)

Description Determines whether the state of connections that are removed from the connection pool for reuse by the application is reset to the initial configuration of the connection.

This connection option can affect performance. See ["Performance Considerations" on page 179](#) for details.

Valid Values 0 | 1

If set to 1 (Enabled), the state of connections removed from the connection pool for reuse by an application is reset to the initial configuration of the connection. Resetting the state can negatively impact performance because additional commands must be sent over the network to the server to reset the state of the connection.

If set to 0 (Disabled), the state of connections is not reset.

Default 0 (Disabled)

GUI Tab [Pooling tab](#) on page 140

## Connection Retry Count

Attribute ConnectionRetryCount (CRC)

Description The number of times the driver retries connection attempts to the primary database server, and if specified, alternate servers until a successful connection is established.

This option and the Connection Retry Delay connection option, which specifies the wait interval between attempts, can be used in conjunction with failover.

Valid Values 0 |  $x$

where  $x$  is a positive integer from 1 to 65535.

If set to 0, the driver does not try to connect after the initial unsuccessful attempt.

If set to  $x$ , the driver retries connection attempts the specified number of times. If a connection is not established during the retry attempts, the driver returns an error that is generated by the last server to which it tried to connect.

Default 0

GUI Tab [Failover tab](#) on page 139



## Connection Retry Delay

Attribute	ConnectionRetryDelay (CRD)
Description	<p>The number of seconds the driver waits between connection retry attempts when Connection Retry Count is set to a positive integer.</p> <p>This option and the Connection Retry Count connection option can be used in conjunction with failover.</p>
Valid Values	<p>0   <i>x</i></p> <p>where <i>x</i> is a positive integer from 1 to 65535.</p> <p>If set to 0, there is no delay between retries.</p> <p>If set to <i>x</i>, the driver waits the specified number of seconds between connection retry attempts.</p>
Default	3
GUI Tab	<a href="#">Failover tab</a> on page 139

## Current Function Path

Attribute	CurrentFunctionPath (CFP)
Description	Specifies a comma-separated list of DB2 schema names used to resolve unqualified function names and data type references in dynamically prepared SQL statements. This value also is used to resolve unqualified stored procedure names specified in CALL statements.
Valid Values	<p><i>schema_name</i> [, ...]</p> <p>where <i>schema_name</i> is the name of a valid DB2 schema.</p>
Default	None
GUI Tab	<a href="#">Advanced tab</a> on page 135

## Data Source Name

Attribute	DataSourceName (DSN)
Description	The name of a data source in your Windows Registry or odbc.ini file.
Valid Values	<p><i>string</i></p> <p>where <i>string</i> is the name of a data source.</p>
Default	None
GUI Tab	<a href="#">General tab</a> on page 134

## Database Name

Attribute	Database (DB)
Description	The name of the database to which you want to connect.

Valid only for DB2 for Linux/UNIX/Windows.

This option is mutually exclusive with the Location Name and Collection options.

Valid Values

*ext*

where *ext* is the name of the one- to three-character file name extension.

This value is used for all Create Table statements. Sending a Create Table using an extension other than the value specified for this option causes an error.

In other SQL statements, such as Select or Insert, users can specify an extension other than the one specified for this connection option. The Data File Extension value is used when no extension is specified.

Default      None

GUI Tab      [General tab](#) on page 134

Default Isolation Level

Attribute      DefaultIsolationLevel (DIL)

Description      Specifies the method by which locks on data in the database are acquired and released.

The following table shows how ODBC isolation levels map to DB2 isolation levels.

ODBC	DB2
Read Uncommitted	Uncommitted Read
Read Committed	Cursor Stability
Repeatable Read	Read Stability
Serializable	Repeatable Read

Refer to [Chapter 7 “Locking and Isolation Levels”](#) in the *DataDirect Connect Series for ODBC Reference* for details about ODBC isolation levels.

Valid Values

0 | 1 | 2 | 3 | 4

If set to 0 (READ\_UNCOMMITTED), other processes can be read from the database. Only modified data is locked and is not released until the transaction ends.

If set to 1 (READ\_COMMITTED) other processes can change a row that your application has read if the cursor is not on the row you want to change. This level prevents other processes from changing records that your application has changed until your application commits them or ends the transaction.

It also prevents your application from reading a modified record that has not been committed by another process, unless the Concurrent Access Resolution connection option is set to:

- Automatic (0) and the cur\_commit server parameter is set to On
- Use Currently Committed (2) and the cur\_commit server parameter is set to On or Available

In either of these cases, the application can read the last committed value. See the connection option ["Concurrent Access Resolution" on page 159](#) for further details.

See ["Cursor Stability Isolation Level" on page 186](#) for information about enhancements to the Read Committed (Cursor Stability) isolation level.

If set to 2 (REPEATABLE\_READ), other processes are prevented from accessing data that your application has read or modified. All read or modified data is locked until transaction ends.

If set to 3 (SERIALIZABLE), other processes are prevented from changing records that are read or changed by your application (including phantom records) until your program commits them or ends the transaction. This level prevents the application from reading modified records that have not been committed by another process. If your application 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 application.

If set to 4 (NONE), your application can read modified records even if they have not been committed by another application. This level can only be set in the data source, not from the application. (This level is valid only on DB2 for iSeries, and is the only isolation level that works for collections that have journaling disabled.)

Default 1 (READ\_COMMITTED)

GUI Tab [Advanced tab](#) on page 135

## Description

Attribute Description (n/a)

Description An optional long description of a data source. This description is not used as a runtime connection attribute, but does appear in the ODBC.INI section of the Registry and in the `odbc.ini` file.

Valid Values *string*

where *string* is a description of a data source.

Default None

GUI Tab [General tab](#) on page 134

## Dynamic Sections

Attribute DynamicSections (DS)

Description Specifies the maximum number of prepared statements that the driver can have open at any time.

A dynamic section is associated with a prepared statement. The driver only keeps open the number of prepared statements specified by the dynamic sections value. If the driver detects that the number of dynamic sections available in the bound DB2 packages is less

than the number of dynamic sections requested in the connection string or data source, it generates the following message:

The current number of dynamic sections available for use is different than the number of dynamic sections currently specified in the connection string or data source.

Valid Values  $x$

where  $x$  is a positive integer that specifies the number of prepared statements.

Default 1000

GUI Tab [Modify Bindings tab](#) on page 138

## Enable Bulk Load

Attribute EnableBulkLoad (EBL)

Description Specifies the bulk load method.

Valid Values 0 | 1

If set to 1 (Enabled), the driver uses the database bulk load protocol when an application executes an INSERT with multiple rows of parameter data. If the protocol cannot be used, the driver returns a warning.

If set to 0 (Disabled), the driver uses standard parameter arrays.

Default 0 (Disabled)

GUI Tab [Bulk tab](#) on page 141

## Encryption Method

Attribute EncryptionMethod (EM)

Description The method the driver uses to encrypt data sent between the driver and the database server. If the specified encryption method is not supported by the database server, the connection fails and the driver returns an error.

This connection option can affect performance. See ["Performance Considerations" on page 179](#) for details.

Valid Values 0 | 1 | 2

If set to 0 (No Encryption), data is not encrypted.

If set to 1 (SSL), data is encrypted using SSL. If the server supports protocol negotiation, the driver and server negotiate the use of TLS v1, SSL v3, or SSL v2 in that order.

If set to 2 (Database Encryption), data is encrypted using the DB2 encryption protocol (supported only on DB2 for Linux/UNIX/Windows and DB2 for z/OS).

This option can only be set to 1 or 2 when Authentication Method is set to 0, 1, or 2.

Default 0 (No Encryption)

GUI Tab [Security tab](#) on page 137



## Failover Granularity

Attribute	FailoverGranularity (FG)
Description	<p>Determines whether the driver fails the entire failover process or continues with the process if errors occur while trying to reestablish a lost connection.</p> <p>This option applies only when Failover Mode is set to 1 (Extended Connection) or 2 (Select).</p> <p>The Alternate Servers option specifies one or multiple alternate servers for failover and is required for all failover methods.</p>
Valid Values	<p>0   1   2   3</p> <p>If set to 0 (Non-Atomic), the driver continues with the failover process and posts any errors on the statement on which they occur.</p> <p>If set to 1 (Atomic) the driver fails the entire failover process if an error is generated as the result of anything other than executing and repositioning a Select statement. If an error is generated as a result of repositioning a result set to the last row position, the driver continues with the failover process, but generates a warning that the Select statement must be reissued.</p> <p>If set to 2 (Atomic Including Repositioning), the driver fails the entire failover process if any error is generated as the result of restoring the state of the connection or the state of work in progress.</p> <p>If set to 3 (Disable Integrity Check), the driver does not verify that the rows that were restored during the failover process match the original rows. This value applies only when Failover Mode is set to 2 (Select).</p>
Default	0 (Non-Atomic)
GUI Tab	<a href="#">Failover tab</a> on page 139

## Failover Mode

Attribute	FailoverMode (FM)
Description	<p>Specifies the type of failover method the driver uses.</p> <p>The Alternate Servers option specifies one or multiple alternate servers for failover and is required for all failover methods.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 179</a> for details.</p>
Valid Values	<p>0   1   2</p> <p>If set to 0 (Connection), the driver provides failover protection for new connections only.</p> <p>If set to 1 (Extended Connection), the driver provides failover protection for new and lost connections, but not any work in progress.</p> <p>If set to 2 (Select), the driver provides failover protection for new and lost connections. In addition, it preserves the state of work performed by the last Select statement executed.</p>

Default 0 (Connection)  
 GUI Tab [Failover tab](#) on page 139

## Failover Preconnect

Attribute FailoverPreconnect (FP)

Description Specifies whether the driver tries to connect to the primary and an alternate server at the same time.

This attribute applies only when Failover Mode is set to 1 (Extended Connection) or 2 (Select) and at least one alternate server is specified.

The Alternate Servers option specifies one or multiple alternate servers for failover and is required for all failover methods.

Valid Values 0 | 1

If set to 0 (Disabled), the driver tries to connect to an alternate server only when failover is caused by an unsuccessful connection attempt or a lost connection. This value provides the best performance, but your application typically experiences a short wait while the failover connection is attempted.

If set to 1 (Enabled), the driver tries to connect to the primary and an alternate server at the same time. This can be useful if your application is time-sensitive and cannot absorb the wait for the failover connection to succeed.

Default 0 (Disabled)

GUI Tab [Failover tab](#) on page 139

## Fetch Time Stamp With Time Zone as Timestamp

Attribute FetchTSWTZasTimestamp

Description Determines whether the driver returns Timestamp with Time Zone columns as an ODBC SQL\_TYPE\_TIMESTAMP or as a SQL\_VARCHAR data type.

Valid only for DB2 for z/OS, version 10 or higher.

Valid Values 0 | 1

If set to 0 (Disabled), Timestamp with Time Zone columns are mapped to SQL\_VARCHAR. Use this setting if your application needs to retrieve the information as a string.

If set to 1 (Enabled), the driver maps Timestamp with Time Zone columns to the ODBC SQL\_TYPE\_TIMESTAMP data type. The time zone information is truncated from the results.

Default 0 (Disabled)

GUI Tab [Advanced tab](#) on page 135

## Field Delimiter

Attribute	BulkLoadFieldDelimiter (BLFD)
Description	Specifies the character that the driver will use to delimit the field entries in a bulk load data file.
Valid Values	<i>x</i>  where <i>x</i> is any printable character.  For simplicity, avoid using a value that can be in the data, including all alphanumeric characters, the dash(-), the colon(:), the period (.), the forward slash (/), the space character, the single quote (') and the double quote ("). You can use some of these characters as delimiters if all of the data in the file is contained within double quotes.  NOTE: The Bulk Load Field Delimiter character must be different from the Bulk Load Record Delimiter.
Default	None
GUI Tab	<a href="#">Bulk tab</a> on page 141

## Grant Execute to [check box]

Attribute	GrantExecute (GE)
Description	Determines how EXECUTE privileges are granted on DB2 packages.
Valid Values	0   1  If set to 1 (Enabled), EXECUTE privileges are granted on DB2 packages that you are creating. By default, the schema to which privileges are granted is PUBLIC.  If set to 0 (Disabled), EXECUTE privileges are granted to the schema that created the DB2 packages.
Default	1 (Enabled)
GUI Tab	<a href="#">Modify Bindings tab</a> on page 138

## Grant Execute to [field]

Attribute	GrantAuthid (GA)
Description	Determines which DB2 schema is granted EXECUTE privileges for DB2 packages.
Valid Values	<i>schema_name</i>  where <i>schema_name</i> is the name of a valid DB2 schema.
Default	PUBLIC
GUI Tab	<a href="#">Modify Bindings tab</a> on page 138

## GSS Client Library

Attribute	GSSClient (GSSC)
Description	<p>The name of the GSS client library that the driver uses to communicate with the Key Distribution Center (KDC).</p> <p>The driver uses the path defined by the PATH environment variable for loading the specified client library.</p>
Valid Values	<p>native   <i>client_library</i></p> <p>where <i>client_library</i> is a GSS client library installed on the client.</p> <p>If set to <i>client_library</i>, the driver uses the specified GSS client library.</p> <p>If set to native, the driver uses the GSS client shipped with the operating system.</p>
Default	native
GUI Tab	<a href="#">Security tab</a> on page 137

## Host Name In Certificate

Attribute	HostNameInCertificate (HNIC)
Description	<p>A host name for certificate validation when SSL encryption is enabled (Encryption Method=1) and validation is enabled (Validate Server Certificate=1). This option provides additional security against man-in-the-middle (MITM) attacks by ensuring that the server the driver is connecting to is the server that was requested.</p>
Valid Values	<p><i>host_name</i>   #SERVERNAME#</p> <p>where the <i>host_name</i> is the host name specified in the certificate. Consult your SSL administrator for the correct value.</p> <p>If set to a host name, the driver examines the subjectAltName values included in the certificate. If a dnsName value is present in the subjectAltName values, then the driver compares the value specified for Host Name In Certificate with the dnsName value. The connection succeeds if the values match. The connection fails if the Host Name In Certificate value does not match the dnsName value.</p> <p>If no subjectAltName values exist or a dnsName value is not in the list of subjectAltName values, then the driver compares the value specified for Host Name In Certificate with the commonName part of the Subject name in the certificate. The commonName typically contains the host name of the machine for which the certificate was created. The connection succeeds if the values match. The connection fails if the Host Name In Certificate value does not match the commonName. If multiple commonName parts exist in the Subject name of the certificate, the connection succeeds if the Host Name In Certificate value matches any of the commonName parts.</p> <p>If set to #SERVERNAME#, the driver compares the host server name specified as part of a data source or connection string to the dnsName or the commonName value.</p>
Default	None
GUI Tab	<a href="#">Security tab</a> on page 137





## IANAAppCodePage

IANAAppCodePage (IACP)

**Description** An Internet Assigned Numbers Authority (IANA) value. You must specify a value for this option if your application is not Unicode-enabled or if your database character set is not Unicode. Refer to [Chapter 4 “Internationalization, Localization, and Unicode”](#) in the *DataDirect Connect Series for ODBC Reference* for details.

The driver uses the specified IANA code page to convert "W" (wide) functions to ANSI.

The driver and Driver Manager both check for the value of IANAAppCodePage in the following order:

- In the connection string
- In the Data Source section of the system information file (odbc.ini)
- In the ODBC section of the system information file (odbc.ini)

If the driver does not find an IANAAppCodePage value, the driver uses the default value of 4 (ISO 8859-1 Latin-1).

**Valid Values** *IANA\_code\_page*

where *IANA\_code\_page* is one of the valid values listed in [Chapter 1 “Values for the Attribute IANAAppCodePage”](#) in the *DataDirect Connect Series for ODBC Reference*. The value must match the database character encoding and the system locale.

**Default** 4 (ISO 8559-1 Latin-1)

**GUI Tab** [Advanced tab](#) on page 135

## Ip Address

**Attribute** IpAddress (IP)

**Description** Identifies the machine where catalog tables are stored.

**Valid Values** *host\_name* | *IP\_address*

where

*host\_name* is the host name of the machine where catalog tables are stored. The driver must be able to find this name (with the correct address assignment) in the HOSTS file on the workstation or in a DNS server.

*IP\_address* is the IP address of the machine where catalog tables are stored. The IP address can be specified in either IPv4 or IPv6 format, or a combination of the two. See ["Using IP Addresses" on page 54](#) for details about these formats.

**Default** localhost

**GUI Tab** [General tab](#) on page 134

## Key Password

Attribute	KeyPassword (KP)
Description	The password used to access the individual keys in the keystore file when SSL is enabled (Encryption Method=1) and SSL client authentication is enabled on the database server. Keys stored in a keystore can be individually password-protected. To extract the key from the keystore, the driver must have the password of the key.
Valid Values	<i>key_password</i>  where <i>key_password</i> is the password of a key in the keystore.
Default	None
GUI Tab	<a href="#">Security tab</a> on page 137

## Keystore

Attribute	Keystore (KS)
Description	The name of the directory containing the keystore file to be used when SSL is enabled (Encryption Method=1) and SSL client authentication is enabled on the database server. The keystore file contains the certificates that the client sends to the server in response to the server's certificate request. If you do not specify a directory, the current directory is used.  NOTE: The keystore and truststore files may be the same file.
Valid Values	<i>keystore_directory</i>  where <i>keystore_directory</i> is the location of the keystore file.
Default	None
GUI Tab	<a href="#">Security tab</a> on page 137

## Keystore Password

Attribute	KeystorePassword (KSP)
Description	The password used to access the keystore file when SSL is enabled (Encryption Method=1) and SSL client authentication is enabled on the database server. The keystore file contains the certificates that the client sends to the server in response to the server's certificate request.  NOTE: The keystore and truststore files may be the same file; therefore, they may have the same password.
Valid Values	<i>keystore_password</i>  where <i>keystore_password</i> is the password of the keystore file.
Default	None
GUI Tab	<a href="#">Security tab</a> on page 137

## Load Balance Timeout

Attribute	LoadBalanceTimeout (LBT)
Description	<p>The number of seconds to keep inactive connections open in a connection pool. An inactive connection is a database session that is not associated with an ODBC connection handle, that is, a connection in the pool that is not in use by an application.</p> <p>NOTE: The Min Pool Size option may cause some connections to ignore this value.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 179</a> for details.</p>
Valid Values	<p>0   <math>x</math></p> <p>where <math>x</math> is a positive integer that specifies a number of seconds.</p> <p>If set to 0, inactive connections are kept open.</p> <p>If set to <math>x</math>, inactive connections are closed after the specified number of seconds passes.</p>
Default	0
GUI Tab	<a href="#">Pooling tab</a> on page 140

## Load Balancing

Attribute	LoadBalancing (LB)
Description	Determines whether the driver uses client load balancing in its attempts to connect to the database servers (primary and alternate). You can specify one or multiple alternate servers by setting the Alternate Servers option.
Valid Values	<p>0   1</p> <p>If set to 1 (Enabled), the driver uses client load balancing and attempts to connect to the database servers (primary and alternate servers) in random order.</p> <p>If set to 0 (Disabled), the driver does not use client load balancing and connects to each server based on their sequential order (primary server first, then, alternate servers in the order they are specified).</p> <p>NOTE: This option has no effect unless alternate servers are defined for the Alternate Servers connection option.</p>
Default	0 (Disabled)
GUI Tab	<a href="#">Failover tab</a> on page 140

## Location Name

Attribute	Location (LOC)
Description	<p>Valid only for DB2 for z/OS and iSeries. The name of the DB2 location that you want to access.</p> <p>On DB2 for z/OS, your system administrator can determine the name of your DB2 location using the <code>DISPLAY DDF</code> command.</p>

On DB2 for iSeries, your system administrator can determine the name of your DB2 location using the `WRKRDBDIRE` command. The name of the database that is listed as \*LOCAL" is the value to use.

This option is mutually exclusive with the Database Name option.

Valid Values *location\_name*

where *location\_name* is the name of a valid DB2 location.

Default None

GUI Tab [General tab](#) on page 134

## Login Timeout

Attribute LoginTimeout (LT)

Description The number of seconds the driver waits for a connection to be established before returning control to the application and generating a timeout error. To override the value that is set by this connection option for an individual connection, set a different value in the `SQL_ATTR_LOGIN_TIMEOUT` connection attribute using the `SQLSetConnectAttr()` function.

Valid Values -1 | 0 | *x*

where *x* is a positive integer that specifies a number of seconds.

If set to -1, the connection request does not time out. The driver silently ignores the `SQL_ATTR_LOGIN_TIMEOUT` attribute.

If set to 0, the connection request does not time out, but the driver responds to the `SQL_ATTR_LOGIN_TIMEOUT` attribute.

If set to *x*, the connection request times out after the specified number of seconds unless the application overrides this setting with the `SQL_ATTR_LOGIN_TIMEOUT` attribute.

Default 15

GUI Tab [Advanced tab](#) on page 135

## Max Pool Size

Attribute MaxPoolSize (MXPS)

Description The maximum number of connections allowed within a single connection pool. When the maximum number of connections is reached, no additional connections can be created in the connection pool.

This connection option can affect performance. See ["Performance Considerations" on page 179](#) for details.

Valid Values An integer from 1 to 65535

For example, if set to 20, the maximum number of connections allowed in the pool is 20.

Default 100

GUI Tab [Pooling tab](#) on page 140

## Min Pool Size

Attribute	MinPoolSize (MNPS)
Description	<p>The minimum number of connections that are opened and placed in a connection pool, in addition to the active connection, when the pool is created. The connection pool retains this number of connections, even when some connections exceed their Load Balance Timeout value.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 179</a> for details.</p>
Valid Values	<p>0   <i>x</i></p> <p>where <i>x</i> is an integer from 1 to 65535.</p> <p>For example, if set to 5, the start-up number of connections in the pool is 5 in addition to the current existing connection.</p> <p>If set to 0, no connections are opened in addition to the current existing connection.</p>
Default	0
GUI Tab	<a href="#">Pooling tab</a> on page 140

## Package Collection

Attribute	PackageCollection (PC)
Description	Specifies the name of the DB2 collection or location where the driver creates bind packages and, when required, searches for them.
Valid Values	<p><i>collection_name</i></p> <p>where <i>collection_name</i> is a valid DB2 collection or location name.</p>
Default	NULLID
GUI Tab	<a href="#">Modify Bindings tab</a> on page 138

## Package Name Prefix

Attribute	PackageNamePrefix (PNP)
Description	<p>Specifies a two-character prefix used for package names when the driver executes dynamic SQL. The default package name uses the following syntax:</p> <p><i>DD<i>i</i>VRM<i>x</i></i></p> <p>where:</p> <p><i>DD</i> is the two-character prefix.</p> <p><i>i</i> is one of the following characters:</p> <ul style="list-style-type: none"> <li>■ S—Serializable : DB2 RR</li> <li>■ R—Repeatable Read : DB2 RS</li> <li>■ C—Committed Read : DB2 CS</li> </ul>

- U—Uncommitted Read : DB2 UC
- N—Not committed: DB2 NC

*VRM* is the Version Release Modification, for example, you can specify 520 to represent version 5.2.0.

*x* is a one-character suffix that specifies:

- A—Cursor queries/updates
- B—Cursor queries/updates with hold
- C—Stored procedures (section 1 is for stored procedures that do not have parameters; section 2 is for procedures that do have parameters)

For example, the package name DDOC520A would represent a package using the Committed Read isolation level, at version 5.20, and using cursor queries/updates.

Valid Values *xx*

where *xx* is a two-character prefix.

Default DD

GUI Tab [Advanced tab](#) on page 135

## Package Owner

Attribute PackageOwner (PO)

Description Specifies the AuthID assigned to the package.

Valid Values *authid*

where *authid* is a valid DB2 AuthID that has permissions to execute all the SQL in the package.

Default None

GUI Tab [Modify Bindings tab](#) on page 138

## Password

Attribute Password (PWD)

Description The password that the application uses to connect to your database. The Password option cannot be specified through the driver Setup dialog box and should not be stored in a data source. It is specified through the Logon dialog box or a connection string.

Valid Values *pwd*

where *pwd* is a valid password.

Default None

GUI Tab n/a

## Program ID

Attribute	ProgramID (PID)
Description	<p>Specifies the product and version information of the driver on the client to be stored in the database. This value sets the CLIENT_PRDID value in the database. For DB2 V9.1 and higher for Linux/UNIX/Windows, this value is located in the SYSIBMADM.APPLICATIONS table. This value is used by the DB2 Workload Manager.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 179</a> for details.</p>
Valid Values	<p><i>DDTVVRRM</i></p> <p>where <i>DDTVVRRM</i> is a value that identifies the product and version of the driver on the client.</p> <ul style="list-style-type: none"> <li>■ <i>DDT</i> identifies a DataDirect Connect driver.</li> <li>■ <i>VV</i> identifies a 2-digit version number (with high-order 0 in the case of a 1-digit version).</li> <li>■ <i>RR</i> identifies a 2-digit release number (with high-order 0 in the case of a 1-digit release).</li> <li>■ <i>M</i> identifies a 1-character modification level (0-9 or A-Z).</li> </ul>
Example	DDT06010
Default	None
GUI Tab	<a href="#">Client Monitoring tab</a> on page 147

## Query Timeout

Attribute	QueryTimeout (QT)
Description	<p>The number of seconds for the default query timeout for all statements that are created by a connection. To override the value set by this connection option for an individual statement, set a different value in the SQL_ATTR_QUERY_TIMEOUT statement attribute on the SQLSetStmtAttr() function.</p> <p>Query timeout is supported on DB2 for Linux/UNIX/Windows 8.1 and higher and on DB2 for z/OS 8.1 and higher</p>
Valid Values	<p>-1   0   <i>x</i></p> <p>where <i>x</i> is a positive integer that specifies a number of seconds.</p> <p>If set to -1, the query does not time out. The driver silently ignores the SQL_ATTR_QUERY_TIMEOUT attribute.</p> <p>If set to 0, the query does not time out, but the driver responds to the SQL_ATTR_QUERY_TIMEOUT attribute.</p> <p>If set to <i>x</i>, all queries time out after the specified number of seconds unless the application overrides this value by setting the SQL_ATTR_QUERY_TIMEOUT attribute.</p>
Default	0
GUI Tab	<a href="#">Advanced tab</a> on page 135

## Record Delimiter

Attribute	BulkLoadRecordDelimiter (BLRD)
Description	Specifies the character that the driver will use to delimit the record entries in a bulk load data file.
Valid Values	<i>x</i>  where <i>x</i> is any printable character.  For simplicity, avoid using a value that can be in the data, including all alphanumeric characters, the dash(-), the colon(:), the period (.), the forward slash (/), the space character, the single quote (') and the double quote ("). You can use some of these characters as delimiters if all of the data in the file is contained within double quotes.  NOTE: The Bulk Load Record Delimiter character must be different from the Bulk Load Field Delimiter.
Default	None
GUI Tab	<a href="#">Bulk tab</a> on page 141

## Report Codepage Conversion Errors

Attribute	ReportCodepageConversionErrors (RCCE)
Description	Specifies how the driver handles code page conversion errors that occur when a character cannot be converted from one character set to another.  An error message or warning can occur if an ODBC call causes a conversion error, or if an error occurs during code page conversions to and from the database or to and from the application. The error or warning generated is <code>Code page conversion error encountered</code> . In the case of parameter data conversion errors, the driver adds the following sentence: <code>Error in parameter <i>x</i></code> , where <i>x</i> is the parameter number. The standard rules for returning specific row and column errors for bulk operations apply.
Valid Values	0   1   2  If set to 0 (Ignore Errors), the driver substitutes 0x1A for each character that cannot be converted and does not return a warning or error.  If set to 1 (Return Error), the driver returns an error instead of substituting 0x1A for unconverted characters.  If set to 2 (Return Warning), the driver substitutes 0x1A for each character that cannot be converted and returns a warning.
Default	0 (Ignore Errors)
GUI Tab	<a href="#">Advanced tab</a> on page 135

## Tcp Port

Attribute	TcpPort (PORT)
Description	Specifies the port number that is assigned to the DB2 DRDA listener process on the server host machine.



On DB2 for iSeries only, execute `NETSTAT` from a command line to determine the correct port number. Select option 3 to display a list of active ports on the iSeries machine. Find the entry for DRDA, and press F14 to toggle and display the port number. If DRDA is not currently listening, the command `CHGDDMTCPA AUTOSTART(*YES) PWDRQD(*YES)` starts the listener and ensures that it is active at IPL.

Valid Values *IP\_address* | *service\_name*

where:

*IP\_address* is the port's IP address.

*service\_name* is the port's service name. The driver must be able to find this name (with the correct port assignment) in the SERVICES file on the workstation.

Default 50000

GUI Tab [General tab](#) on page 134

## Truststore

Attribute Truststore (TS)

Description The directory that contains the truststore file and the truststore file name to be used when SSL is enabled (Encryption Method=1) and server authentication is used. The truststore file contains a list of the valid Certificate Authorities (CAs) that are trusted by the client machine for SSL server authentication. If you do not specify a directory, the current directory is used.

NOTE: The truststore and keystore files may be the same file.

Valid Values *truststore\_directory\filename*

where *truststore\_directory* is the directory where the truststore file is located and *filename* is the file name of the truststore file.

Default None

GUI Tab [Security tab](#) on page 137

## Truststore Password

Attribute TruststorePassword (TSP)

Description The password that is used to access the truststore file when SSL is enabled (Encryption Method=1) and server authentication is used. The truststore file contains a list of the Certificate Authorities (CAs) that the client trusts.

NOTE: The truststore and keystore files may be the same file; therefore, they may have the same password.

Valid Values *truststore\_password*

where *truststore\_password* is a valid password for the truststore file.

Default None

GUI Tab [Security tab](#) on page 137

## Use Current Schema for Catalog Functions

Attribute	UseCurrentSchema (UCS)
Description	Specifies whether results are restricted to the tables and views in the current schema if a catalog function call is made without specifying a schema or if the schema is specified as the wildcard character %. Restricting results to the tables and views in the current schema improves performance of catalog calls that do not specify a schema.
Valid Values	0   1  If set to 1 (Enabled), results of catalog function calls are restricted to the tables and views in the current schema.  If set to 0 (Disabled), results of catalog function calls are not restricted.
Default	0 (Disabled)
GUI Tab	<a href="#">Advanced tab</a> on page 135

## User Name

Attribute	LogonID (UID)
Description	The default user ID that is used to connect to your database. Your ODBC application may override this value or you may override it in the logon dialog box or connection string.
Valid Values	<i>userid</i>  where <i>userid</i> is a valid user ID with permissions to access the database.
Default	None
GUI Tab	<a href="#">Security tab</a> on page 137

## Validate Server Certificate

Attribute	ValidateServerCertificate (VSC)
Description	Determines whether the driver validates the certificate that is sent by the database server when SSL encryption is enabled (Encryption Method=1). When using SSL server authentication, any certificate sent by the server must be issued by a trusted Certificate Authority (CA). Allowing the driver to trust any certificate returned from the server even if the issuer is not a trusted CA is useful in test environments because it eliminates the need to specify truststore information on each client in the test environment.  Truststore information is specified using the Trust Store and Trust Store Password options.
Valid Values	0   1  If set to 1 (Enabled), the driver validates the certificate that is sent by the database server. Any certificate from the server must be issued by a trusted CA in the truststore file. If the Host Name In Certificate option is specified, the driver also validates the certificate using a host name. The Host Name In Certificate option provides additional security against man-in-the-middle (MITM) attacks by ensuring that the server the driver is connecting to is the server that was requested.

If set to 0 (Disabled), the driver does not validate the certificate that is sent by the database server. The driver ignores any truststore information specified by the Trust Store and Trust Store Password options.

Default 1 (Enabled)  
 GUI Tab [Security tab](#) on page 137

## With Hold Cursors

Attribute WithHold (WH)  
 Description Determines whether the cursor stays open on a commit.  
 Valid Values 0 | 1

If set to 1 (Enabled), cursor behavior is Preserve, which keeps cursors open after a commit or rollback (SQLGetInfo( ) returns SQL\_CB\_PRESERVE for SQL\_COMMIT\_CURSOR\_BEHAVIOR).

If set to 0 (Disabled), cursor behavior is Delete, which closes all cursors open after a commit or rollback (SQLGetInfo( ) returns SQL\_CB\_DELETE).

Default 1 (Enabled)  
 GUI Tab [Advanced tab](#) on page 135

## XML Describe Type

Attribute XMLDescribeType (XDT)  
 Description The SQL data type that is returned by SQLGetTypeInfo for the XML data type.

See ["Using the XML Data Type" on page 182](#) for further information about the XML data type.

Valid Values -4 | -10

If set to -4 (SQL\_LONGVARIABLE), the driver uses the description SQL\_LONGVARIABLE for columns that are defined as the XML data type.

If set to -10 (SQL\_WLONGVARIABLE), the driver uses the description SQL\_WLONGVARIABLE for columns that are defined as the XML data type.

Default -10  
 GUI Tab [Advanced tab](#) on page 135

---

## Performance Considerations

The following connection options can enhance driver performance. You can also enhance performance through efficient application design. Refer to [Chapter 5 "Designing ODBC Applications for Performance Optimization"](#) in the *DataDirect Connect Series for ODBC Reference* for details.

**Application Using Threads (ApplicationUsingThreads):** The driver coordinates concurrent database operations (operations from different threads) by acquiring locks. Although locking prevents errors in the driver, it also decreases performance. If your application does not make ODBC calls from different threads, the driver has no reason to coordinate operations. In this case, the ApplicationUsingThreads attribute should be disabled (set to 0).

NOTE: If you are using a multi-threaded application, you must enable the Application Using Threads option.

**Connection Pooling (ConnectionPooling):** If you enable the driver to use connection pooling, you can set additional options that affect performance:

- **Load Balance Timeout:** You can define how long to keep connections in the pool. The time that a connection was last used is compared to the current time and, if the timespan exceeds the value of the Load Balance Timeout option, the connection is destroyed. The Min Pool Size option can cause some connections to ignore this value.
- **Connection Reset:** Resetting a re-used connection to the initial configuration settings impacts performance negatively because the connection must issue additional commands to the server.
- **Max Pool Size:** Setting the maximum number of connections that the pool can contain too low might cause delays while waiting for a connection to become available. Setting the number too high wastes resources.
- **Min Pool Size:** A connection pool is created when the first connection with a unique connection string connects to the database. The pool is populated with connections up to the minimum pool size, if one has been specified. The connection pool retains this number of connections, even when some connections exceed their Load Balance Timeout value.

**Encryption Method (EncryptionMethod):** Data encryption may adversely affect performance because of the additional overhead (mainly CPU usage) that is required to encrypt and decrypt data.

**Failover Mode (FailoverMode):** Although high availability that replays queries after a failure provides increased levels of protection, it can adversely affect performance because of increased overhead.

**Use Current Schema for Catalog Functions (UseCurrentSchema):** If your application needs to access database objects owned only by the current user, then performance can be improved. In this case, the Use Current Schema for Catalog Functions option must be enabled. When this option is enabled, the driver returns only database objects owned by the current user when executing catalog functions. Calls to catalog functions are optimized by grouping queries. Enabling this option is equivalent to passing the Logon ID used on the connection as the SchemaName argument to the catalog functions.

**Workload Manager:** The Workload Manager (WLM) automatically adjusts server resources, such as CPU and memory, based on the service class associated with a DB2 workload. Therefore, an application's performance is tied to the DB2 workload to which it is assigned and, ultimately, to the service class associated with that workload. The DB2 Wire Protocol driver allows your application to set client information in the DB2 database that can be used by the WLM to classify work. If you know that your database environment uses

WLM, coordinate with your database administrator to determine how setting the following options affects performance.

- **Accounting Info:** Sets the CURRENT CLIENT\_ACCTNG register (DB2 for Linux/UNIX/Windows) or the CLIENT ACCTNG register (DB2 for z/OS) on the server.
- **Application Name:** Sets the CURRENT CLIENT\_APPLNAME register (DB2 for Linux/UNIX/Windows) or CLIENT APPLNAME register (DB2 for z/OS) on the server.
- **Client Host Name:** Sets the CURRENT CLIENT\_WRKSTNNAME register (DB2 for Linux/UNIX/Windows) or CLIENT WRKSTNNAME register (DB2 for z/OS) on the server.
- **Client User:** Sets the CURRENT CLIENT\_USERID register (DB2 for Linux/UNIX/Windows) and CLIENT USERID register (DB2 for z/OS) on the server.
- **Program ID:** Sets the CLIENT\_PRDID value on the server.

---

## IBM to IANA Code Page Values

Refer to ["IBM to IANA Code Page Values" on page 18](#) in [Chapter 1](#) of the *DataDirect Connect Series for ODBC Reference* for a list of the most commonly used IBM code pages and their IANA code page equivalents. The IANA values are valid for the CharsetFor65535 connection string attribute and the Character Set for the CCSID 65535 option.

---

## Data Types

[Table 5-2](#) shows how the DB2 data types map to the standard ODBC data types. ["Unicode Support" on page 183](#) lists DB2 to Unicode data type mappings.

---

**Table 5-2. DB2 Data Types**

---

DB2	ODBC
Bigint <sup>1</sup>	SQL_BIGINT
Blob <sup>2</sup>	SQL_LONGVARBINARY
Char	SQL_CHAR
Char() for Bit Data	SQL_BINARY
Clob <sup>3</sup>	SQL_LONGVARCHAR
Date	SQL_TYPE_DATE
Decfloat <sup>4</sup>	SQL_DOUBLE
Decimal	SQL_DECIMAL
Double	SQL_DOUBLE
Float	SQL_DOUBLE
Integer	SQL_INTEGER

**Table 5-2. DB2 Data Types** (cont.)

DB2	ODBC
Long Varchar	SQL_LONGVARCHAR
Long Varchar for Bit Data	SQL_LONGVARBINARY
Numeric	SQL_NUMERIC
Real	SQL_REAL
Smallint	SQL_SMALLINT
Time	SQL_TYPE_TIME
Timestamp <sup>5</sup>	SQL_TYPE_TIMESTAMP
Timestamp With Time Zone <sup>6,7</sup>	SQL_VARCHAR
Varchar	SQL_VARCHAR
Varchar() for Bit Data	SQL_VARBINARY
XML <sup>7</sup>	SQL_LONGVARCHAR

---

1. Supported on DB2 V8.x and higher for Linux/UNIX/Windows, DB2 V9 and higher for z/OS, and DB2 V5R3 and higher for iSeries.
2. Supported on DB2 V8.x and higher for Linux/UNIX/Windows; DB2 for z/OS; and DB2 V5R3 and higher for iSeries.
3. On DB2 for Linux/UNIX/Windows versions previous to V8.1 and DB2 V5R2 for iSeries, only the first 32 KB of the Clob data are returned when fetching, and only 32 KB can be inserted and updated.
4. Supported only on DB2 V9 and higher for Linux/UNIX/Windows, DB2 V9 and higher for z/OS, and DB2 V6R1 for iSeries.
5. Timestamp values with a fractional seconds precision greater than 9 are described as the ODBC SQL\_VARCHAR data type.
6. Timestamp with Time Zone mapping changes based on the setting of the FetchTSWTZasTimestamp option only on DB2 V10 and higher for z/OS.
7. Supported only on DB2 V10 for z/OS.

---

See ["Retrieving Data Type Information" on page 59](#) for information about retrieving data types.

## Using the XML Data Type

By default, DB2 returns XML data to the driver encoded as UTF-8. To avoid data loss, an application must bind XML data as SQL\_C\_WCHAR. The driver then returns the data as either UTF-8 or UTF-16, depending on platform and application settings. If the application binds XML data as SQL\_C\_CHAR, the driver converts it to the client character encoding, possibly causing data loss or corruption. To prevent any conversion of XML data, the application must set the attribute [XML Describe Type](#) to SQL\_LONGVARBINARY (-10) and bind the data as SQL\_C\_BINARY.

## Unicode Support

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

The DB2 Wire Protocol driver maps the DB2 data types to Unicode data types as shown in the following table:

DB2 Data Type	Mapped to. . .
Dbclob <sup>1</sup>	SQL_WLONGVARCHAR
Graphic	SQL_WCHAR
Long Vargraphic	SQL_WLONGVARCHAR
Vargraphic	SQL_WVARCHAR

1. Supported on DB2 V8.x and higher for Linux/UNIX/Windows; DB2 for z/OS; and DB2 V5R3 and higher for iSeries.

## Advanced Features

The driver supports the following advanced features:

- Failover
- Client Information
- Security
- Connection Pooling
- DataDirect Bulk Load

### Failover

The driver supports failover and its related connection options. Failover connection options are located on the [Failover tab](#) of the driver Setup dialog box. See ["Using Failover" on page 65](#) for a general description of failover and its implementation.

### Client Information

DB2 provides a Workload Manager (WLM) that allows an administrator to define different work load classifications and store client information associated with a connection. The WLM is a priority and resource manager within DB2 for Linux/UNIX/Windows. On z/OS, the WLM is part of the operating system. WLM prioritizes and matches DB2 workloads with available resources. DB2 workloads allow you to categorize similar types of work. For example, a database administrator can create a DB2 workload named Sales to service all connections that come from Sales applications.

The WLM automatically adjusts server resources, such as CPU and memory, based on the service class associated with a DB2 workload. Therefore, an application's performance is tied to the DB2 workload to which it is assigned and, ultimately, to the service class

associated with that workload. For example, an application that performs batch work nightly when resource usage is low can use the default workload. In contrast, sales updates that need to be processed quickly twice a day need to use a workload that is governed by a high priority service class.

It is important to understand that, unless specified otherwise, all work runs in the default workload that is governed by the default service class. To ensure the best performance, consult with your database administrator to verify that your application is associated with the appropriate DB2 workload and service class.

In addition to workload management, WLM also provides monitoring functionality that is useful for troubleshooting. For example, the database administrator can set threshold limits to detect long-running queries and gather information about those queries.

The DB2 Wire Protocol driver allows your application to set client information in the DB2 database that can be used by the WLM to classify work. If you know that your database environment uses WLM, coordinate with your database administrator to determine which connection options you need to set. These options are located on the [Client Monitoring](#) tab of the driver Setup dialog box. See ["Using Client Information" on page 75](#) for a general description of client information and its implementation.

## DB2 V9.5 and V9.7 for Linux/UNIX/Windows

[Table 5-3](#) lists the WLM attributes for DB2 V9.5 and V9.7 for Linux/UNIX/Windows that map to information set by driver connection options. Refer to your DB2 documentation for information about using these WLM attributes.

**Table 5-3. WLM Attributes for DB2 V9.5 and V9.7 for Linux/UNIX/Windows**

WLM Attribute	Driver Option	Description
APPLNAME	Application Name	Name of the application currently using the connection
CURRENT CLIENT_ACCTNG	Accounting Info	Additional information that may be used for accounting or troubleshooting purposes, such as an accounting ID
CLIENT_PRDID	Program ID	Product name and version of the driver on the client
CURRENT CLIENT_USERID	Client User	User ID for whom the application using the connection is performing work.
CURRENT CLIENT_WRKSTNNAME	Client Host Name	Host name of the client on which the application using the connection is running

## DB2 for z/OS

[Table 5-4](#) lists the WLM attributes for DB2 for z/OS that map to information set by driver connection options. Refer to your DB2 documentation for information about using these WLM attributes.



**Table 5-4. WLM Attributes for DB2 z/OS**

WLM Attribute	Driver Option	Description
Accounting Info (AI)	Accounting Info	Additional information that may be used for accounting or troubleshooting purposes, such as an accounting ID.
Correlation Info (CI)	Program ID	Product name and version of the driver on the client
Collection Name (CN)	Package Collection	Name of the collection or library (group of packages) to which DB2 packages are bound.
Package Name (PK)	Package Name Prefix	Name of the first DB2 package accessed by the DRDA requester in the unit of Work.
Process Name (PC)	Application Name	Name of the application currently using the connection
Userid (UI)	Client User	User ID for whom the application using the connection is performing work.

## Security

The driver supports authentication and encryption. Security connection options are located on the [Security tab](#) of the driver Setup dialog box. See ["Using Security" on page 77](#) for a general description of security and its implementation. The following security information is specific to the DB2 Wire Protocol Driver.

## Authentication

If you are using Kerberos, verify that your environment meets the requirements listed in [Table 5-5](#) before you configure the driver for Kerberos authentication.

**Table 5-5. Kerberos Authentication Requirements for the DB2 Wire Protocol Driver**

Component	Requirements
Database server	<p>The database server must be running one of the following database versions:</p> <ul style="list-style-type: none"> <li>■ DB2 V8.1 or higher for Linux/UNIX/Windows</li> <li>■ DB2 V8.x or higher for z/OS</li> </ul>
Kerberos server	<p>The Kerberos server is the machine where the user IDs for authentication are administered. The Kerberos server is also the location of the Kerberos KDC.</p> <p>Network authentication must be provided by one of the following methods:</p> <ul style="list-style-type: none"> <li>■ Windows Active Directory on one of the following operating systems: <ul style="list-style-type: none"> <li>• Windows Server 2003</li> <li>• Windows 2000 Server Service Pack 3 or higher</li> </ul> </li> <li>■ MIT Kerberos 1.4.2 or higher</li> </ul>

## Connection Pooling

The driver supports connection pooling and its related connection options. Connection pooling connection options are located on the [Pooling tab](#) of the driver Setup dialog box. See ["Using DataDirect Connection Pooling" on page 82](#) for a general description of connection pooling and its implementation.

## DataDirect Bulk Load

The driver supports DataDirect bulk load and its related connection options. Bulk load connection options are located on the [Bulk tab](#) of the driver Setup dialog box. See ["Using DataDirect Bulk Load" on page 85](#) for a general description of DataDirect bulk load and its implementation.

---

## Cursor Stability Isolation Level

The DB2 Cursor Stability (CS) isolation level has been enhanced in DB2 V9.7 to reduce significantly instances of lock wait and deadlock. In previous DB2 versions, CS prevented any row that was changed by other applications from being read until the change was committed.

In this enhanced implementation, CS, where possible, avoids a read operation waiting for a row to commit before returning a value. CS now returns the currently committed result, ignoring what might happen to an uncommitted operation. Some exceptions, such as updatable cursors, exist; currently committed results cannot be returned immediately when the row might be updated based upon its previous contents. CS behavior is determined through the [Concurrent Access Resolution](#) connection option.

Consider the following example, in which deadlocks are avoided under the currently committed semantics. In this scenario, two applications update two separate tables, but do not yet commit. Each application then attempts to read (with a read-only cursor) from the table that the other application has updated.

---

**Table 5-6. Cursor Stability Examples**

---

Step	Application A	Application B
1	UPDATE T1 SET col1 = ? WHERE col2 = ?	UPDATE T2 SET col1 = ? WHERE col2 = ?
2	SELECT col1, col3, col4 FROM T2 WHERE col2 >= ?	SELECT col1, col5, FROM T1 WHERE col5 = ? AND col2 = ?
3	Commit	Commit

---

Without currently committed semantics, these applications running under the cursor stability isolation level might create a deadlock, causing one of the applications to fail. This happens when each application needs to read data that is being updated by the other application.

Under currently committed semantics, if the query in step 2 (for either application) happens to require the data currently being updated by the other application, that application does not wait for the lock to be released, making a deadlock impossible. The previously committed version of the data is located and used instead.

---

## XQuery Expressions

The DB2 Wire Protocol driver supports execution of XQuery expressions in DB2 V9.1 and higher for Linux/UNIX/Windows. IBM provides a tutorial on this topic at the following URL:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp>

---

## Stored Procedure Support

The DB2 Wire Protocol driver supports DB2 Remote Procedure Calls (RPCs) as follows:

- Multiple result sets are 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 supported as stored procedure parameters.

---

## Unexpected Characters

Users are sometimes surprised when they insert a character into a database, only to have a different character displayed when they fetch it from the database. There are many reasons this can happen, but it most often involves code page issues, not driver errors.

Client and server machines in a database system each use code pages, which can be identified by a name or a number, such as Shift\_JIS (Japanese) or cp1252 (Windows English). A code page is a mapping that associates a sequence of bits, called a code point, with a specific character. Code pages include the characters and symbols of one or more languages. Regardless of geographical location, a machine can be configured to use a specific code page. Most of the time, a client and database server would use similar, if not identical, code pages. For example, a client and server might use two different Japanese code pages, such as Shift\_JIS and EUC\_JP, but they would still share many Japanese characters in common. These characters might, however, be represented by different code points in each code page. This introduces the need to convert between code pages to maintain data integrity. In some cases, no one-to-one character correspondence exists between the two code points. This causes a substitution character to be used, which can result in displaying an unexpected character on a fetch.

When the driver on the client machine opens a connection with the database server, the driver determines the code pages being used on the client and the server. This is

determined from the Active Code Page on a Windows-based machine. If the client machine is UNIX-based, the driver checks the IANAAppCodePage attribute (see ["IANAAppCodePage" on page 169](#)). If it does not find a specific setting for IACP, it defaults to a value of ISO\_8859\_1 Latin\_1.

If the client and server code pages are compatible, the driver transmits data in the code page of the client. Even though the pages are compatible, a one-to-one correspondence for every character may not exist. If the client and server code pages are completely dissimilar, for example, Russian and Japanese, then many substitutions occur because very few, if any, of the characters are mapped between the two code pages.

The following is a specific example of an unexpected character:

- The client machine is running the Japanese code page EUC\_JP.
- The DB2 server is running the Japanese code page Shift\_JIS.
- When you insert the EUC\_JP code point 0xA1BD and then fetch it back, you do not see the character you expected. In fact, what displays on the client may not be a recognizable character.

This substitution occurs because the code points do not correspond in the two code pages. EUC\_JP code point 0xA1BD is converted to UTF-16 code point 0x2014. Code point 0x2014 does not map to anything in Shift\_JIS, resulting in the Shift\_JIS substitution code point, 0x3F, being sent to, and stored in, the database. When this character is retrieved, depending on the client display, it may not display as a recognizable character.

This is not a driver error. It occurs because the code points map differently and because some characters do not exist in a code page. The best way to avoid these problems is to use the same code page on both the client and server machines.

---

## Support for DB2 pureScale

IBM introduced DB2 pureScale to provide scaleout active and active services for IBM DB2 running on AIX on Power Systems servers. It is designed to deliver distributed availability and scalability in a clustered database system. DB2 pureScale allows a single physical DB2 database to be accessed by concurrent instances of DB2 running across several different cluster members.

A DB2 pureScale shared disk cluster is composed of a group of independent servers, or members, that cooperate as a single system. A cluster architecture such as this provides applications access to more computing power when needed, while allowing computing resources to be used for other applications when database resources are not as heavily required. For example, in the event of a sudden increase in network traffic, a DB2 pureScale cluster can distribute the load over many nodes, a feature referred to as *transaction level workload balancing*. DB2 pureScale features are available to you simply by connecting to a DB2 pureScale system with the DB2 driver. No additional configuration is required.

*Connection failover* and *client load balancing* can be used in conjunction with a DB2 pureScale shared disk cluster, but they are not specifically part of DB2 pureScale. See ["Using Failover" on page 65](#) for details about how these features work in DataDirect Connect Series for ODBC drivers.

---

## Persisting a Result Set as an XML Data File

The driver allows you to persist a result as an XML data file with embedded schema. See ["Persisting a Result Set as an XML Data File" on page 60](#) for details about implementation.

---

## Isolation and Lock Levels Supported

DB2 supports isolation level 0 (read uncommitted), isolation level 1 (read committed), isolation level 2 (repeatable read), isolation level 3 (serializable), and, on DB2 for iSeries only, isolation level 4 (none).

Refer to [Chapter 7 "Locking and Isolation Levels"](#) in the *DataDirect Connect Series for ODBC Reference* for details.

---

## SQL Support

The driver supports the minimum SQL grammar.

---

## ODBC Conformance Level

The driver is Level 1 compliant, that is, it supports all ODBC Core and Level 1 functions.

In addition, the following functions are supported:

- SQLProcedures
- SQLProcedureColumns

Refer to [Chapter 2 "ODBC API and Scalar Functions"](#) in the *DataDirect Connect Series for ODBC Reference* for a list of the API functions supported by the DB2 Wire Protocol driver.

---

## Number of Connections and Statements Supported

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

---

## Using Arrays of Parameters

DB2 for Linux/UNIX/Windows natively supports parameter arrays, and the DB2 Wire Protocol driver, in turn, supports them when connected to these DB2 databases. When designing an application for performance, using native parameter arrays for bulk inserts or updates, for example, can improve performance. Refer to [Chapter 5 “Designing ODBC Applications for Performance Optimization”](#) in the *DataDirect Connect Series for ODBC Reference* for more information about using arrays of parameters to improve performance.

The DB2 Wire Protocol driver accepts a proprietary statement attribute called `SQL_ATTR_PARAM_ARRAY_ATOMIC`. It has two values: `SQL_PA_ATOMIC_YES (1)` and `SQL_PA_ATOMIC_NO (0)`.

When set to `SQL_PA_ATOMIC_YES`, the default, parameter array operations are atomic, meaning that if one row in the parameter array fails, then the entire array must fail.

When set to `SQL_PA_ATOMIC_NO`, parameter array operations are not atomic, meaning that the parameter array continues to be processed, even if one of the rows fails.

## 6 The Informix Wire Protocol Driver

The DataDirect Connect *for* ODBC and DataDirect Connect64 *for* ODBC Informix Wire Protocol driver (the Informix Wire Protocol driver) each support multiple connections to the following Informix database servers:

- Informix Dynamic Server 11.0, 11.5, 11.7
- Informix Dynamic Server 10.0
- Informix Dynamic Server 9.2, 9.3, 9.4

The Informix Wire Protocol driver is supported in the Windows, UNIX, and Linux environments. See ["Environment-Specific Information" on page 45](#) for detailed information about the environments supported by this driver.

Refer to the readme file shipped with your DataDirect product for the file name of the Informix Wire Protocol driver.

NOTE: The Informix Wire Protocol driver does not require any Informix client software. Progress DataDirect also provides an Informix client-based driver that can access earlier versions of Informix databases. See ["The Informix Driver" on page 587](#) for details.

---

### Driver Requirements

The driver has no client requirements.

---

### Configuring and Connecting to Data Sources

After you install the driver, you configure data sources to connect to the database. See [Chapter 1 "Quick Start Connect" on page 35](#) for an explanation of different types of data sources. The data source contains connection options that allow you to tune the driver for specific performance. If you want to use a data source but need to change some of its values, you can either modify the data source or override its values at connection time through a connection string.

If you choose to use a connection string, you must use specific connection string attributes. See ["Using a Connection String" on page 196](#) and [Table 6-1 on page 198](#) for an alphabetical list of driver connection string attributes and their initial default values.

#### Data Source Configuration in the UNIX/Linux odbc.ini File

On UNIX and Linux, you must set up the proper ODBC environment before configuring data sources. See ["Environment Configuration" on page 37](#) for basic setup information and ["Environment Variables" on page 97](#) for more detail about this procedure.

Data sources for UNIX and Linux are stored in the system information file (by default, `odbc.ini`). If you have a Motif GUI environment on Linux, you can configure and modify data sources through the DataDirect ODBC Data Source Administrator for Linux (the Linux ODBC Administrator) using a driver Setup dialog box. (See ["Configuration Through the Administrator" on page 100](#) for a detailed explanation of the Administrator.)

If you do not have a GUI environment, you can configure and modify data sources directly by editing the `odbc.ini` file and storing default connection values there. See ["Configuration Through the System Information \(odbc.ini\) File" on page 103](#) for detailed information about the specific steps necessary to configure a data source.

[Table 6-1 on page 198](#) lists driver connection string attributes that must be used in the `odbc.ini` file to set the value of the attributes. Note that only the long name of the attribute can be used in the file. The default listed in the table is the initial default value when the driver is installed.

## Data Source Configuration through a GUI

On Windows, data sources are stored in the Windows Registry. You can configure and modify data sources through the ODBC Administrator using a driver Setup dialog box, as described in this section.

On UNIX and Linux, data sources are stored in the `odbc.ini` file. On Linux, you can configure and modify data sources through the Linux ODBC Administrator using a driver Setup dialog box, as described in this section.

NOTE: This book shows dialog box images that are specific to Windows. If you are using the drivers in the Linux environment, the dialog box that you see may differ slightly from the Windows version. Windows-only and UNIX-only connection options are specifically noted by icons in the Setup dialog box descriptions.

When the driver is first installed, the values of its connection options are set by default. These values appear on the driver Setup dialog box tabs when you create a new data source. You can change these default values by modifying the data source. In the following procedure, the description of each tab is followed by a table that lists the connection options for that tab and their initial default values. This table links you to a complete description of the options and their connection string attribute equivalents. The connection string attributes are used to override the default values of the data source if you want to change these values at connection time.

### To configure an Informix data source:

- 1 Start the ODBC Administrator:
  - On Windows, start the ODBC Administrator by selecting its icon from the DataDirect Connect program group.
  - On Linux, change to the `install_dir/tools` directory and, at a command prompt, enter:
 

```
odbcadmin
```

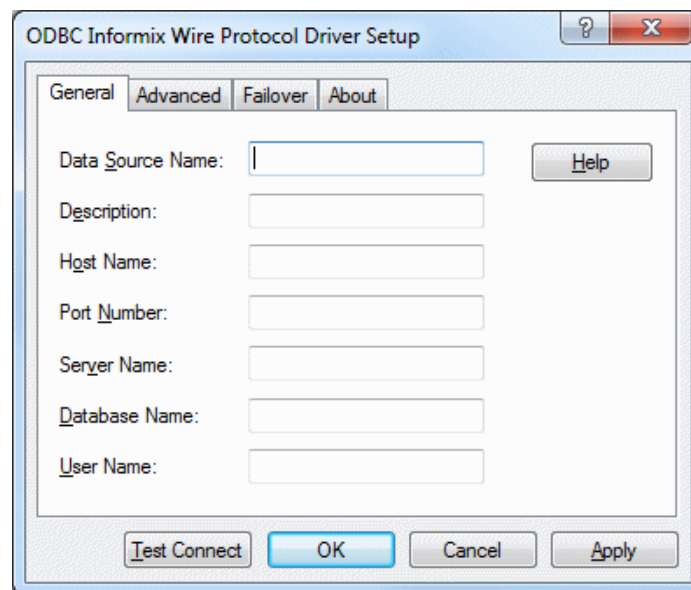
where `install_dir` is the path to the product installation directory.



2 Select a tab:

- **User DSN:** If you are configuring an existing user data source, select the data source name and click **Configure** to display the driver Setup dialog box.  
If you are configuring a new user data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.
- **System DSN:** If you are configuring an existing system data source, select the data source name and click **Configure** to display the driver Setup dialog box.  
If you are configuring a new system data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.
- **File DSN:** If you are configuring an existing file data source, select the data source file and click **Configure** to display the driver Setup dialog box.  
If you are configuring a new file data source, click **Add** to display a list of installed drivers; then, select a driver. Click **Advanced** if you want to specify attributes; otherwise, click **Next** to proceed. Specify a name for the data source and click **Next**. Verify the data source information; then, click **Finish** to display the driver Setup dialog box.

The General tab of the Setup dialog box appears by default.



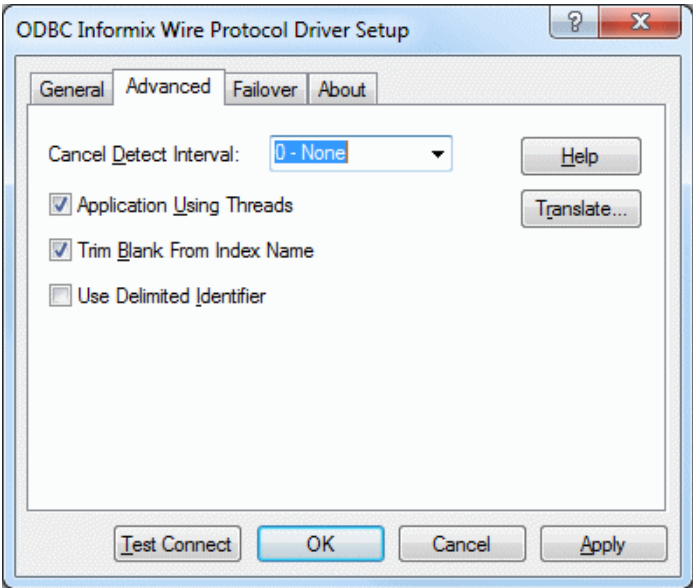
NOTE: The General tab displays only fields that are required for creating a data source. The fields on all other tabs are optional, unless noted otherwise.

- 3 On this tab, provide values for the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: General	Default
<a href="#">Data Source Name (see page 201)</a>	None
<a href="#">Description (see page 201)</a>	None
<a href="#">Host Name (see page 201)</a>	None

Connection Options: General	Default
<a href="#">Port Number (see page 203)</a>	None
<a href="#">Server Name (see page 203)</a>	None
<a href="#">Database Name (see page 200)</a>	None
<a href="#">User Name (see page 204)</a>	None

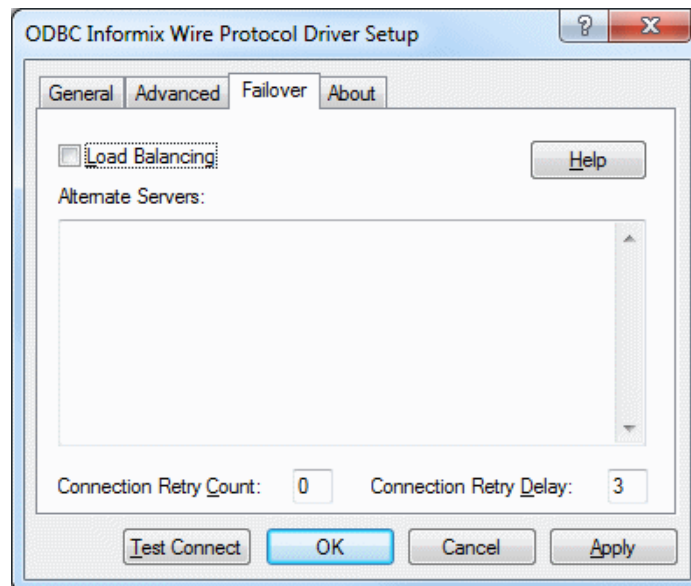
- 4 Optionally, click the **Advanced** tab to specify additional data source settings.



On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Advanced	Default
<a href="#">Cancel Detect Interval (see page 199)</a>	0 - None
<a href="#">Application Using Threads (see page 199)</a>	Enabled
<a href="#">Trim Blank From Index Name (see page 203)</a>	Enabled
<a href="#">Use Delimited Identifier (see page 203)</a>	Disabled
<a href="#">IANAAppCodePage (see page 201)</a>	4 (ISO 8559-1 Latin-1)
UNIX ONLY	

- 5 Optionally, click the **Failover** tab to specify failover data source settings.



See ["Using Failover" on page 65](#) for a general description of failover and its related connection options.

On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Failover	Default
<a href="#">Load Balancing (see page 202)</a>	Disabled
<a href="#">Alternate Servers (see page 198)</a>	None
<a href="#">Connection Retry Count (see page 200)</a>	0
<a href="#">Connection Retry Delay (see page 200)</a>	3

- 6 At any point during the configuration process, you can click **Test Connect** to attempt to connect to the data source using the connection options specified in the driver Setup dialog box. A logon dialog box appears (see ["Using a Logon Dialog Box" on page 197](#) for details). 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 incorrect environment or connection value, it displays an appropriate error message. Click **OK**.

NOTE: If you are configuring alternate servers for use with the connection failover feature, be aware that the Test Connect button tests only the primary server, not the alternate servers.

- 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.

## Using a Connection String

If you want to use a connection string for connecting to a database, or if your application requires it, you must specify either a DSN (data source name), a File DSN, or a DSN-less connection in the string. The difference is whether you use the DSN=, FILEDSN=, or the DRIVER= keyword in the connection string, as described in the ODBC specification. A DSN or FILEDSN connection string tells the driver where to find the default connection information. Optionally, you may specify *attribute=value* pairs in the connection string to override the default values stored in the data source.

The DSN connection string has the form:

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

The FILEDSN connection string has the form:

```
FILEDSN=filename.dsn[;attribute=value[;attribute=value]...]
```

The DSN-less connection string specifies a driver instead of a data source. All connection information must be entered in the connection string because the information is not stored in a data source.

The DSN-less connection string has the form:

```
DRIVER=[{ }driver_name[ } ] [;attribute=value[;attribute=value]...]
```

[Table 6-1 on page 198](#) lists the long and short names for each attribute, as well as the initial default value when the driver is first installed. You can specify either long or short names in the connection string.

An example of a DSN connection string with overriding attribute values for Informix is:

```
DSN=INFORMIX TABLES;DB=PAYROLL
```

A FILEDSN connection string is similar except for the initial keyword:

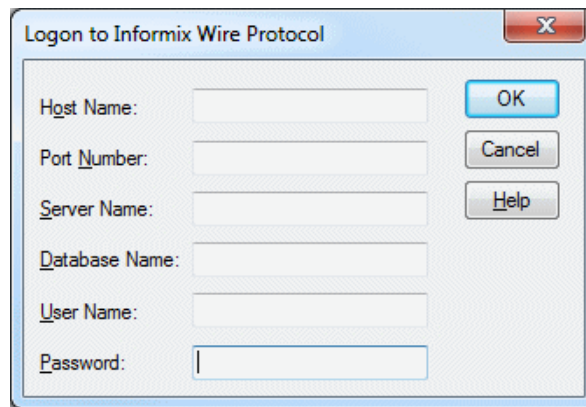
```
FILEDSN=Informix.dsn;DB=DBPAYROLL
```

A DSN-less connection string must provide all necessary connection information:

```
DRIVER=DataDirect 7.1 Informix Wire Protocol;HOST=INF2;PORT=4321;  
SRVR=ACCT;DB=PAYROLL;UID=JOHN;PWD=XYZZY
```

## 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, provide the following information:

- 1 In the Host Name field, type the name or IP address of the host machine on which the Informix server resides.  
  
The IP address can be specified in either IPv4 or IPv6 format, or a combination of the two. See ["Using IP Addresses" on page 54](#) for details concerning these formats.
- 2 In the Port Number field, type the port number of the server listener.
- 3 In the Server Name field, type the name of the Informix server.
- 4 In the Database Name field, type the name of the database to which you want to connect.
- 5 If required, type your user name as specified on the Informix server.
- 6 If required, type your password.
- 7 Click **OK** to complete the logon and to update these values in the Registry.

---

## Connection Option Descriptions

The following connection option descriptions are listed alphabetically by the GUI name that appears on the driver Setup dialog box. The connection string attribute name, along with its short name, is listed immediately underneath the GUI name. For example:

### Application Using Threads

Attribute ApplicationUsingThreads (AUT)

In most cases, the GUI name and the attribute name are the same; however, some exceptions exist. If you need to look up an option by its connection string attribute name, please refer to the alphabetical table of connection string attribute names.

Also, a few connection string attributes, for example, Password, do not have equivalent options that appear on the GUI. They are in the list of descriptions alphabetically by their attribute names.

Table 6-1 lists the connection string attributes supported by the Informix Wire Protocol driver.

**Table 6-1. Informix Wire Protocol Attribute Names**

Attribute (Short Name)	Default
AlternateServers (ASRV)	None
ApplicationUsingThreads (AUT)	1 (Enabled)
CancelDetectInterval (CDI)	0 (None)
ConnectionRetryCount (CRC)	0
ConnectionRetryDelay (CRD)	3
Database (DB)	None
DataSourceName (DSN)	None
Description (n/a)	None
HostName (HOST)	None
IANAAppCodePage (IACP) UNIX ONLY	4 (ISO 8559-1 Latin-1)
LoadBalancing (LB)	0 (Disabled)
LogonID (UID)	None
Password (PWD)	None
PortNumber (PORT)	None
ServerName (SRVR)	None
TrimBlankFromIndexName (TBFIN)	1 (Enabled)
UseDelimitedIdentifier (UDI)	0 (Disabled)

## Alternate Servers

Attribute	AlternateServers (ASRV)
Description	A list of alternate database servers to which the driver tries to connect if the primary database server is unavailable. Specifying a value for this option enables connection failover for the driver. The value you specify must be in the form of a string that defines the physical location of each alternate server. All of the other required connection information for each alternate server is the same as what is defined for the primary server connection.
Valid Values	(Database=databasename:HostName=hostvalue: PortNumber=portvalue:ServerName=servervalue[, . . .])

You must specify the database, host name, port number, and the server name.

NOTE: An alternate server address in IPv6 format must be enclosed in double quotation marks.

**Example** The following Alternate Servers value defines two alternate database servers for connection failover:

```
(Database=Infdb1:HostName=Informixhost1:PortNumber=5177:ServerName=accounting1,
Database=Infdb2:HostName=Informixhost2:PortNumber=5178:ServerName=accounting2)
```

**Default** None

**GUI Tab** [Failover tab](#) on page 195

## Application Using Threads

**Attribute** ApplicationUsingThreads (AUT)

**Description** Determines whether the driver works with applications using multiple ODBC threads.

This connection option can affect performance. See ["Performance Considerations" on page 204](#) for details.

**Valid Values** 0 | 1

If set to 1 (Enabled), the driver works with single-threaded and multi-threaded applications.

If set to 0 (Disabled), the driver does not work with multi-threaded applications. If using the driver with single-threaded applications, this value avoids additional processing required for ODBC thread-safety standards.

**Default** 1 (Enabled)

**GUI Tab** [Advanced tab](#) on page 194

## Cancel Detect Interval

**Attribute** CancelDetectInterval (CDI)

**Description** Determines whether long-running queries in threaded applications can be cancelled if the application issues a SQLCancel.

This connection option can affect performance. See ["Performance Considerations" on page 204](#) for details.

**Valid Values** 0 |  $x$

where  $x$  is the number of seconds the driver waits before checking for SQLCancel calls.

If set to 0 (None), the driver does not allow long-running queries in threaded applications to be canceled, even if the application issues a SQLCancel.

If set to  $x$  (seconds), for every pending query, the driver checks for SQLCancel calls at the specified interval. If the driver determines that a SQLCancel has been issued, the driver cancels the query.

**Example** If you specify 5, for every pending query, the driver checks every five seconds to see whether the application has issued a SQLCancel call. If it detects a SQLCancel call, the driver cancels the query.

**Default** 0 (None)

**GUI Tab** [Advanced tab](#) on page 194

## Connection Retry Count

Attribute	ConnectionRetryCount (CRC)
Description	<p>The number of times the driver retries connection attempts to the primary database server, and if specified, alternate servers until a successful connection is established.</p> <p>This option and the Connection Retry Delay connection option, which specifies the wait interval between attempts, can be used in conjunction with failover.</p>
Valid Values	<p>0   <math>x</math></p> <p>where <math>x</math> is a positive integer from 1 to 65535.</p> <p>If set to 0, the driver does not try to connect after the initial unsuccessful attempt.</p> <p>If set to <math>x</math>, the driver retries connection attempts the specified number of times. If a connection is not established during the retry attempts, the driver returns an error that is generated by the last server to which it tried to connect.</p>
Default	0
GUI Tab	See <a href="#">Failover tab</a> on page 195

## Connection Retry Delay

Attribute	ConnectionRetryDelay (CRD)
Description	<p>The number of seconds the driver waits between connection retry attempts when Connection Retry Count is set to a positive integer.</p> <p>This option and the Connection Retry Count connection option can be used in conjunction with failover.</p>
Valid Values	<p>0   <math>x</math></p> <p>where <math>x</math> is a positive integer from 1 to 65535.</p> <p>If set to 0, there is no delay between retries.</p> <p>If set to <math>x</math>, the driver waits the specified number of seconds between connection retry attempts.</p>
Default	3
GUI Tab	See <a href="#">Failover tab</a> on page 195

## Database Name

Attribute	Database (DB)
Description	The name of the database to which you want to connect.
Valid Values	<p><i>database_name</i></p> <p>where <i>database_name</i> is the name of a valid database.</p>
Default	None
GUI Tab	<a href="#">General tab</a> on page 193



## Data Source Name

Attribute	DataSourceName (DSN)
Description	The name of a data source in your Windows Registry or odbc.ini file.
Valid Values	<i>string</i>  where <i>string</i> is the name of a data source.
Default	None
GUI Tab	<a href="#">General tab</a> on page 193

## Description

Attribute	Description (n/a)
Description	An optional long description of a data source. This description is not used as a runtime connection attribute, but does appear in the ODBC.INI section of the Registry and in the odbc.ini file.
Valid Values	<i>string</i>  where <i>string</i> is a description of a data source.
Default	None
GUI Tab	<a href="#">General tab</a> on page 193

## Host Name

Attribute	HostName (HOST)
Description	The name or the IP address of the server to which you want to connect.
Valid Values	<i>server_name</i>   <i>IP_address</i>  where:  <i>server_name</i> is the name of the server to which you want to connect.  <i>IP_address</i> is the IP address of the server to which you want to connect.  The IP address can be specified in either IPv4 or IPv6 format, or a combination of the two. See <a href="#">"Using IP Addresses" on page 54</a> for details about these formats.
Default	None
GUI Tab	<a href="#">General tab</a> on page 193

## IANAAppCodePage

Attribute	IANAAppCodePage (IACP)
Description	An Internet Assigned Numbers Authority (IANA) value. You must specify a value for this option if your application is not Unicode-enabled and/or if your database character set is not Unicode (refer to <a href="#">Chapter 4 "Internationalization, Localization, and Unicode"</a> in the <i>DataDirect Connect Series for ODBC Reference</i> for details). The value you specify must match the database character encoding and the system locale.

The Driver Manager checks for the value of IANAAppCodePage in the following order:

- In the connection string
- In the Data Source section of the system information file (odbc.ini)
- In the ODBC section of the system information file (odbc.ini)

Valid Values *IANA\_code\_page*

where *IANA\_code\_page* is one of the valid values listed in [Chapter 1 “Values for the Attribute IANAAppCodePage”](#) in the *DataDirect Connect Series for ODBC Reference*. The value must match the database character encoding and the system locale.

Default 4 (ISO 8559-1 Latin-1)

GUI Tab [Advanced tab](#) on page 194

### Load Balancing

Attribute LoadBalancing (LB)

Description Determines whether the driver uses client load balancing in its attempts to connect to the database servers (primary and alternate). You can specify one or multiple alternate servers by setting the Alternate Servers option.

Valid Values 0 | 1

If set to 1 (Enabled), the driver uses client load balancing and attempts to connect to the database servers (primary and alternate servers) in random order.

If set to 0 (Disabled), the driver does not use client load balancing and connects to each server based on their sequential order (primary server first, then, alternate servers in the order they are specified).

NOTE: This option has no effect unless alternate servers are defined for the Alternate Servers connection option.

Default 0 (Disabled)

GUI Tab [Failover tab](#) on page 195

### Password

Attribute Password (PWD)

Description The password that the application uses to connect to your database. The Password option cannot be specified through the driver Setup dialog box and should not be stored in a data source. It is specified through the Logon dialog box or a connection string.

Valid Values *pwd*

where *pwd* is a valid password.

Default None

GUI Tab n/a

## Port Number

Attribute	PortNumber (PORT)
Description	The port number of the server listener.
Valid Values	<i>port_name</i>  where the <i>port_name</i> is the port number of the server listener. Check with your database administrator for the correct number.
Default	None
GUI Tab	<a href="#">General tab</a> on page 193

## Server Name

Attribute	ServerName (SRVR)
Description	The name of the Informix server.
Valid Values	<i>server_name</i>  where <i>server_name</i> is a name that uniquely identifies the Informix server.
Default	None
GUI Tab	<a href="#">General tab</a> on page 193

## Trim Blank From Index Name

Attribute	TrimBlankFromIndexName (TBFIN)
Description	Determines whether the driver trims leading spaces from system-generated index names. Some applications cannot process a leading space in index names.
Valid Values	If set to 1 (Enabled), the driver trims leading spaces from system-generated index names.  If set to 0 (Disabled), the driver does not trim leading spaces from system-generated index names.
Default	1 (Enabled)
GUI Tab	<a href="#">Advanced tab</a> on page 194

## Use Delimited Identifier

Attribute	UseDelimitedIdentifier (UDI)
Description	Determines whether the driver sets the Informix DELIMIDENT environment variable. The DELIMIDENT environment variable specifies that strings enclosed between double quotation marks (") are delimited database identifiers.
Valid Values	0   1  If set to 1 (enabled), the Informix server interprets strings enclosed in double quotation marks as identifiers, not as string literals.  If set to 0 (disabled), the Informix server interprets strings enclosed in double quotation marks as string literals, not as identifiers.

Default 0 (Disabled)  
 GUI Tab [Advanced tab](#) on page 194

## User Name

Attribute LogonID (UID)  
 Description The default user ID that is used to connect to your database. Your ODBC application may override this value or you may override it in the logon dialog box or connection string.  
 Valid Values *userid*  
 where *userid* is a valid user ID with permissions to access the database.  
 Default None  
 GUI Tab [General tab](#) on page 193

---

## Performance Considerations

The following connection options can enhance driver performance. You can also enhance performance through efficient application design. Refer to [Chapter 5 “Designing ODBC Applications for Performance Optimization”](#) in the *DataDirect Connect Series for ODBC Reference* for details.

**Application Using Threads (ApplicationUsingThreads):** The driver coordinates concurrent database operations (operations from different threads) by acquiring locks. Although locking prevents errors in the driver, it also decreases performance. If your application does not make ODBC calls from different threads, the driver has no reason to coordinate operations. In this case, the ApplicationUsingThreads attribute should be disabled (set to 0).

NOTE: If you are using a multi-threaded application, you must enable the Application Using Threads option.

**Cancel Detect Interval (CancelDetectInterval):** If your application uses threads, it may allow canceling of long running queries (may issue synchronous SQLCancel calls). If your application does not issue synchronous SQLCancel calls, the driver can improve performance if the CancelDetectInterval attribute is disabled (set to 0). In this case, the driver does not incur the overhead of periodically checking for SQLCancel. In the case where your application does issue synchronous SQLCancel calls, this attribute should be set to a value that specifies how often the driver checks to see if a long running query has been canceled.

# Data Types

Table 6-2 shows how the Informix data types map to the standard ODBC data types.

**Table 6-2. Informix Data Types**

Informix	ODBC
BLOB	SQL_LONGVARBINARY
BOOLEAN	SQL_BIT
BYTE	SQL_LONGVARBINARY
CHAR	SQL_CHAR
CLOB	SQL_LONGVARCHAR
DATE	SQL_TYPE_DATE
DATETIME YEAR TO FRACTION(f) <sup>1</sup>	SQL_TYPE_TIMESTAMP
DATETIME YEAR TO SECOND	SQL_TYPE_TIMESTAMP
DATETIME YEAR TO DAY	SQL_TYPE_DATE
DATETIME HOUR TO SECOND	SQL_TYPE_TIME
DATETIME HOUR TO FRACTION(f) <sup>1</sup>	SQL_TYPE_TIME
DECIMAL	SQL_DECIMAL
FLOAT	SQL_DOUBLE
INT8	SQL_BIGINT
INTEGER	SQL_INTEGER
INTERVAL YEAR(p) TO YEAR	SQL_INTERVAL_YEAR
INTERVAL YEAR(p) TO MONTH	SQL_INTERVAL_YEAR_TO_MONTH
INTERVAL MONTH(p) TO MONTH	SQL_INTERVAL_MONTH
INTERVAL DAY(p) TO DAY	SQL_INTERVAL_DAY
INTERVAL DAY(p) TO HOUR	SQL_INTERVAL_DAY_TO_HOUR
INTERVAL DAY(p) TO MINUTE	SQL_INTERVAL_DAY_TO_MINUTE
INTERVAL DAY(p) TO SECOND	SQL_INTERVAL_DAY_TO_SECOND
INTERVAL DAY(p) TO FRACTION(f) <sup>1</sup>	SQL_INTERVAL_DAY_TO_SECOND
INTERVAL HOUR(p) TO HOUR	SQL_INTERVAL_HOUR
INTERVAL HOUR(p) TO MINUTE	SQL_INTERVAL_HOUR_TO_MINUTE
INTERVAL HOUR(p) TO SECOND	SQL_INTERVAL_HOUR_TO_SECOND
INTERVAL HOUR(p) TO FRACTION(f) <sup>1</sup>	SQL_INTERVAL_HOUR_TO_SECOND
INTERVAL MINUTE(p) TO MINUTE	SQL_INTERVAL_MINUTE
INTERVAL MINUTE(p) TO SECOND	SQL_INTERVAL_MINUTE_TO_SECOND
INTERVAL MINUTE(p) TO FRACTION(f) <sup>1</sup>	SQL_INTERVAL_MINUTE_TO_SECOND
INTERVAL SECOND(p) TO SECOND	SQL_INTERVAL_SECOND
INTERVAL SECOND(p) TO FRACTION(f) <sup>1</sup>	SQL_INTERVAL_SECOND
LVARCHAR(p) <sup>2</sup>	SQL_VARCHAR
MONEY	SQL_DECIMAL

**Table 6-2. Informix Data Types** (cont.)

Informix	ODBC
NCHAR	SQL_CHAR
NVARCHAR	SQL_VARCHAR
SERIAL	SQL_INTEGER
SERIAL8	SQL_BIGINT
SMALLFLOAT	SQL_REAL
SMALLINT	SQL_SMALLINT
TEXT	SQL_LONGVARCHAR
VARCHAR	SQL_VARCHAR

- 1. (f) can have a value of 1, 2, 3, 4, or 5. The precision is type-dependent and the scale is 5.
- 2. Supported only on Informix 9.4 and higher servers.

See ["Retrieving Data Type Information" on page 59](#) for information about retrieving data types.

## Advanced Features

The driver supports failover and its related connection options. Failover connection options are located on the [Failover tab](#) of the driver Setup dialog box. See ["Using Failover" on page 65](#) for a general description of failover and its implementation.

## MTS Support

On Windows, the driver can take advantage of Microsoft Transaction Server (MTS) capabilities, specifically, the Distributed Transaction Coordinator (DTC) using the XA Protocol. For a general discussion of MTS and DTC, refer to the help file of the Microsoft Transaction Server SDK.

NOTE: The DataDirect Connect *for* ODBC 32-bit drivers can operate in a 64-bit Windows environment; however, they do not support DTC in this environment. Only the DataDirect Connect64 *for* ODBC 64-bit drivers support DTC in a 64-bit Windows environment.

## Persisting a Result Set as an XML Data File

The driver allows you to persist a result as an XML data file with embedded schema. See ["Persisting a Result Set as an XML Data File" on page 60](#) for details about implementation.

---

## Isolation and Lock Levels Supported

Informix supports isolation levels 0 (read uncommitted), 1 (read committed), and 3 (serializable). Informix supports record-level locking.

Refer to [Chapter 7 “Locking and Isolation Levels”](#) in the *DataDirect Connect Series for ODBC Reference* for details.

---

## SQL Support

The driver supports the minimum SQL grammar.

---

## ODBC Conformance Level

The driver is Level 1 compliant, that is, it supports all ODBC Core and Level 1 functions.

In addition, the following functions are supported:

- SQLColumnPrivileges
- SQLForeignKeys
- SQLTablePrivileges

Refer to [Chapter 2 “ODBC API and Scalar Functions”](#) in the *DataDirect Connect Series for ODBC Reference* for a list of the API functions supported by the Informix Wire Protocol driver.

---

## Number of Connections and Statements Supported

The Informix Wire Protocol driver supports multiple connections and multiple statements per connection to the Informix database system.





# 7 The MySQL Wire Protocol Driver

The DataDirect Connect *for* ODBC and DataDirect Connect64 *for* ODBC MySQL Wire Protocol driver (the MySQL Wire Protocol driver) each support multiple connections to the following server and storage engines:

- MySQL 5.1 server
- MySQL 5.0.x server
- Storage engines
  - InnoDB – Transactional
  - MyISAM – Non-Transactional
  - Memory (formerly HEAP) – Non-Transactional

NOTE: The DataDirect Connect Series *for* ODBC drivers for MySQL Enterprise were developed using the MySQL Protocol Documentation whose copyright is owned by, and licensed by Progress DataDirect from, MySQL AB. If any of the DataDirect Connect Series *for* ODBC is licensed for the MySQL database, the following shall apply: You must purchase commercially licensed MySQL database software or a MySQL Enterprise subscription to use the DataDirect Connect Series *for* ODBC drivers for MySQL Enterprise with MySQL software.

The MySQL Wire Protocol driver is supported in the Windows, UNIX, and Linux environments. See ["Environment-Specific Information" on page 45](#) for detailed information about the environments supported by this driver.

See the readme file shipped with your DataDirect Connect product for the file name of the MySQL Wire Protocol driver.

---

## Driver Requirements

The driver has no client requirements.

---

## Configuring and Connecting to Data Sources

After you install the driver, you configure data sources to connect to the database. See [Chapter 1 "Quick Start Connect" on page 35](#) for an explanation of different types of data sources. The data source contains connection options that allow you to tune the driver for specific performance. If you want to use a data source but need to change some of its values, you can either modify the data source or override its values at connection time through a connection string.

If you choose to use a connection string, you must use specific connection string attributes. See ["Using a Connection String" on page 217](#) and [Table 7-1 on page 219](#) for an alphabetical list of driver connection string attributes and their initial default values.

## Data Source Configuration in the UNIX/Linux odbc.ini File

On UNIX and Linux, you must set up the proper ODBC environment before configuring data sources. See ["Environment Configuration" on page 37](#) for basic setup information and ["Environment Variables" on page 97](#) for more detail about this procedure.

Data sources for UNIX and Linux are stored in the system information file (by default, odbc.ini). If you have a Motif GUI environment on Linux, you can configure and modify data sources through the DataDirect ODBC Data Source Administrator for Linux (the Linux ODBC Administrator) using a driver Setup dialog box. (See ["Configuration Through the Administrator" on page 100](#) for a detailed explanation of the Administrator.)

If you do not have a GUI environment, you can configure and modify data sources directly by editing the odbc.ini file and storing default connection values there. See ["Configuration Through the System Information \(odbc.ini\) File" on page 103](#) for detailed information about the specific steps necessary to configure a data source.

[Table 7-1 on page 219](#) lists driver connection string attributes that must be used in the odbc.ini file to set the value of the attributes. Note that only the long name of the attribute can be used in the file. The default listed in the table is the initial default value when the driver is installed.

## Data Source Configuration through a GUI

On Windows, data sources are stored in the Windows Registry. You can configure and modify data sources through the ODBC Administrator using a driver Setup dialog box, as described in this section.

On UNIX and Linux, data sources are stored in the odbc.ini file. On Linux, you can configure and modify data sources through the Linux ODBC Administrator using a driver Setup dialog box, as described in this section.

**NOTE:** This book shows dialog box images that are specific to Windows. If you are using the drivers in the Linux environment, the dialog box that you see may differ slightly from the Windows version. Windows-only and UNIX-only connection options are specifically noted by icons in the Setup dialog box descriptions.

When the driver is first installed, the values of its connection options are set by default. These values appear on the driver Setup dialog box tabs when you create a new data source. You can change these default values by modifying the data source. In the following procedure, the description of each tab is followed by a table that lists the connection options for that tab and their initial default values. This table links you to a complete description of the options and their connection string attribute equivalents. The connection string attributes are used to override the default values of the data source if you want to change these values at connection time.

**To configure a MySQL data source on Windows:****1** Start the ODBC Administrator:

- On Windows, start the ODBC Administrator by selecting its icon from the DataDirect Connect program group.
- On Linux, change to the *install\_dir/tools* directory and, at a command prompt, enter:

```
odbcadmin
```

where *install\_dir* is the path to the product installation directory.

**2** Select a tab:

- **User DSN:** If you are configuring an existing user data source, select the data source name and click **Configure** to display the driver Setup dialog box.

If you are configuring a new user data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.

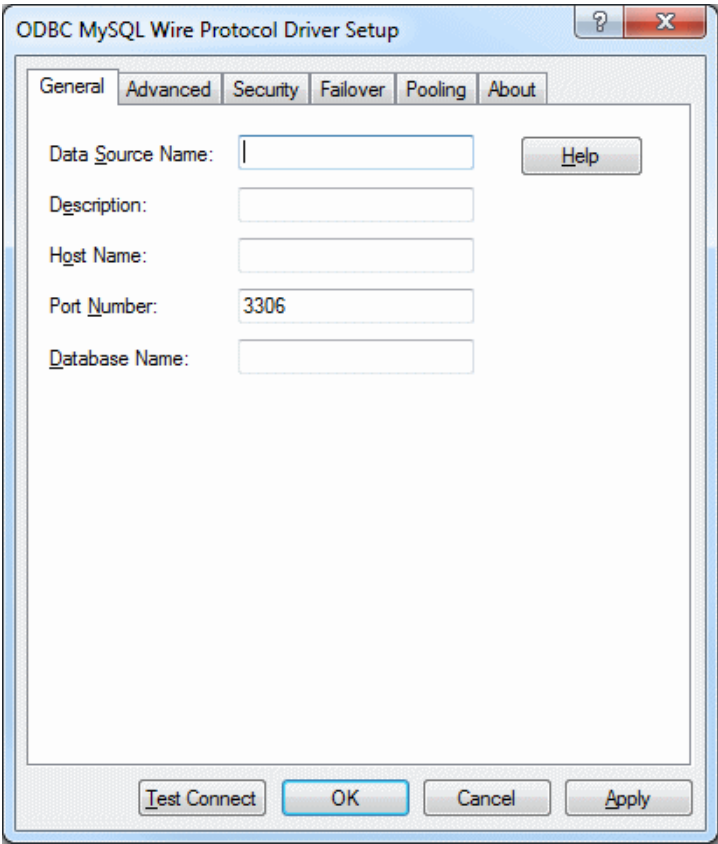
- **System DSN:** If you are configuring an existing system data source, select the data source name and click **Configure** to display the driver Setup dialog box.

If you are configuring a new system data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.

- **File DSN:** If you are configuring an existing file data source, select the data source file and click **Configure** to display the driver Setup dialog box.

If you are configuring a new file data source, click **Add** to display a list of installed drivers; then, select a driver. Click **Advanced** if you want to specify attributes; otherwise, click **Next** to proceed. Specify a name for the data source and click **Next**. Verify the data source information; then, click **Finish** to display the driver Setup dialog box.

The General tab of the Setup dialog box appears by default.

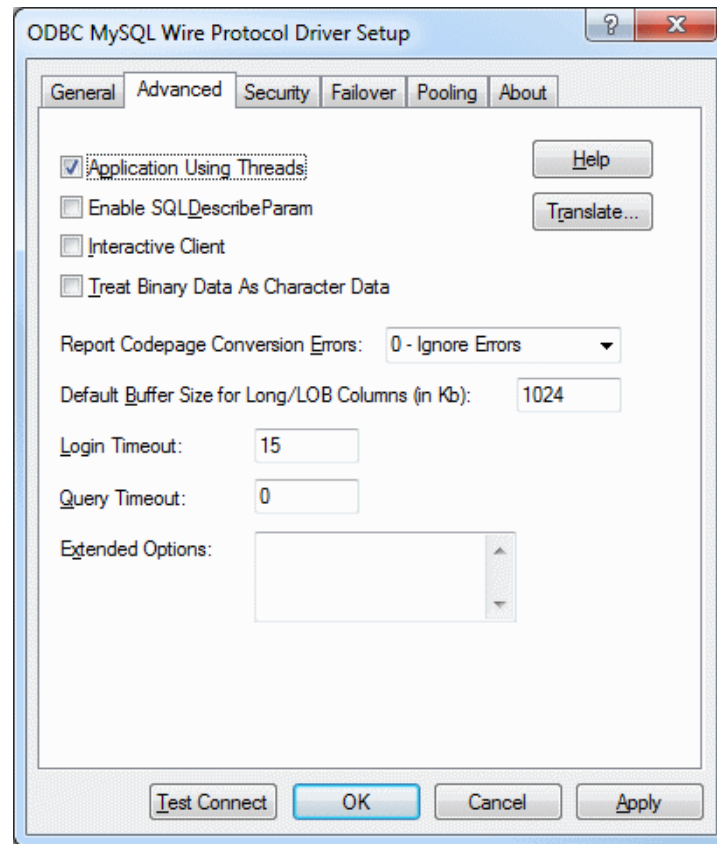


NOTE: The General tab displays only fields that are required for creating a data source. The fields on all other tabs are optional, unless noted otherwise.

- 3 On this tab, provide values for the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: General	Default
<a href="#">Data Source Name (see page 223)</a>	None
<a href="#">Description (see page 224)</a>	None
<a href="#">Host (see page 226)</a>	None
<a href="#">Port Number (see page 231)</a>	3306
<a href="#">Database Name (see page 223)</a>	None

- 4 Optionally, click the **Advanced** tab to specify data source settings.



On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

#### Connection Options: Advanced

	Default
<a href="#">Application Using Threads</a> (see page 221)	Enabled
<a href="#">Enable SQLDescribeParam</a> (see page 224)	Disabled
<a href="#">Interactive Client</a> (see page 228)	Disabled
<a href="#">Treat Binary Data as Character Data</a> (see page 233)	Disabled
<a href="#">Report Codepage Conversion Errors</a> (see page 232)	0 - Ignore Errors
<a href="#">Default Buffer Size for Long/LOB Columns (in Kb)</a> (see page 223)	1024
<a href="#">Login Timeout</a> (see page 230)	15
<a href="#">Query Timeout</a> (see page 232)	0
<a href="#">IANAAppCodePage</a> (see page 227)	4 (ISO 8559-1 Latin-1)
UNIX ONLY	

**Extended Options:** Type a semi-colon separated list of connection options and their values. Use this configuration option to set the value of undocumented connection options that are provided by Progress DataDirect customer support. You can include any valid connection option in the Extended Options string, for example:

```
Database=Server1;UndocumentedOption1=value[;UndocumentedOption2=value;]
```

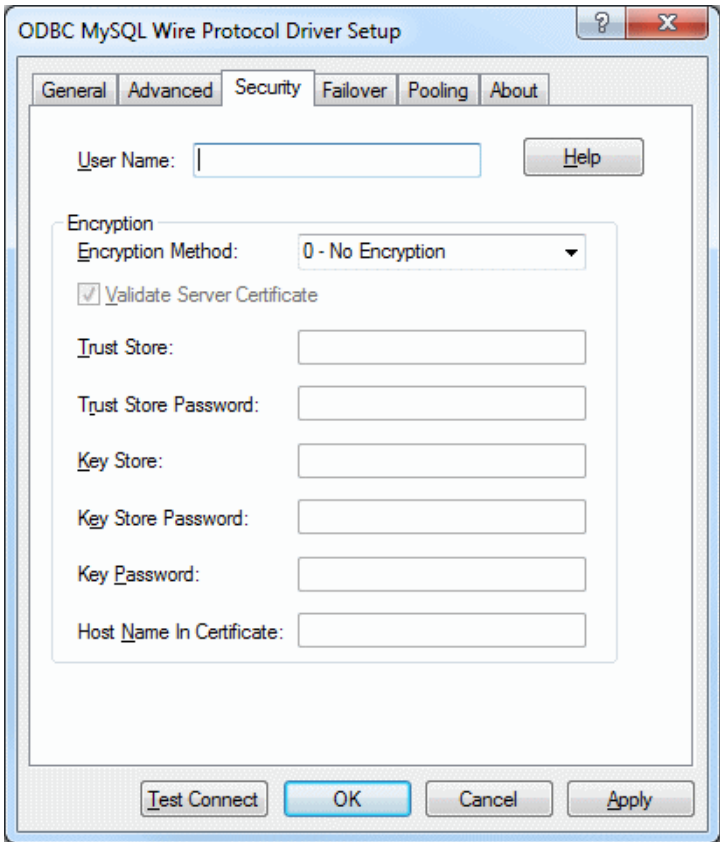
If the Extended Options string contains option values that are also set in the setup dialog or data source, the values of the options specified in the Extended Options string take precedence. However, connection options that are specified on a connection string override any option value specified in the Extended Options string.

NOTE: Do not specify the Extended Options configuration option in a connection string, or the driver will return an error. Instead, applications should specify the individual undocumented connection options in the connection string.

**Translate:** Click **Translate** to display the Select Translator dialog box, which lists the translators specified in the ODBC Translators section of the Registry. Progress 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.

- 5 Optionally, click the **Security** tab to specify security data source settings.



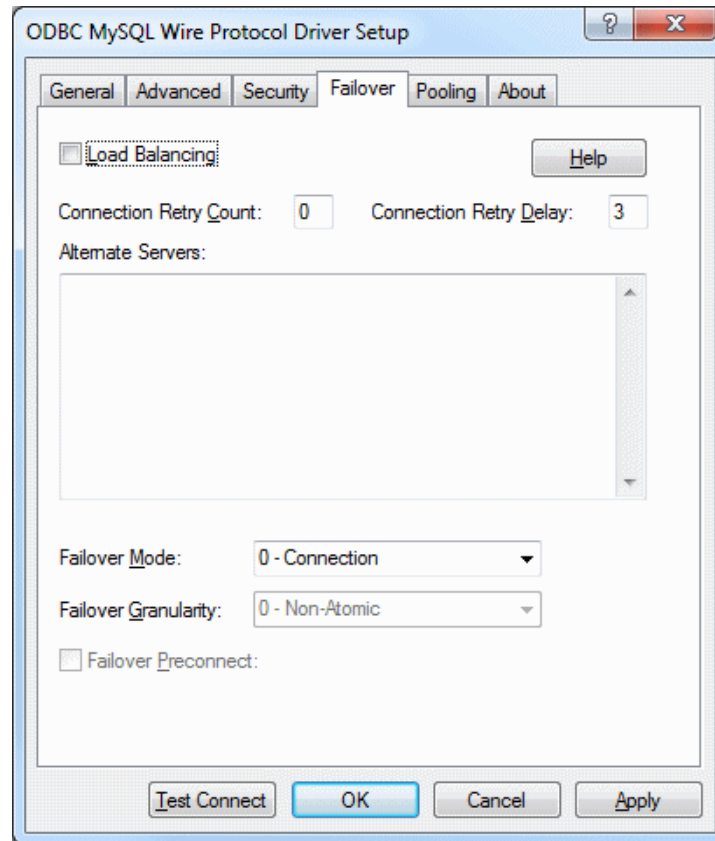
See ["Using Security" on page 77](#) for a general description of authentication and encryption and their configuration requirements.

On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Security	Default
<a href="#">User Name (see page 234)</a>	None
<a href="#">Encryption Method (see page 224)</a>	0 - No Encryption

Connection Options: Security	Default
<a href="#">Validate Server Certificate (see page 234)</a>	Enabled
<a href="#">Truststore (see page 233)</a>	None
<a href="#">Truststore Password (see page 234)</a>	None
<a href="#">Keystore (see page 228)</a>	None
<a href="#">Keystore Password (see page 229)</a>	None
<a href="#">Key Password (see page 228)</a>	None
<a href="#">Host Name In Certificate (see page 226)</a>	None

- 6 Optionally, click the **Failover** tab to specify failover data source settings.



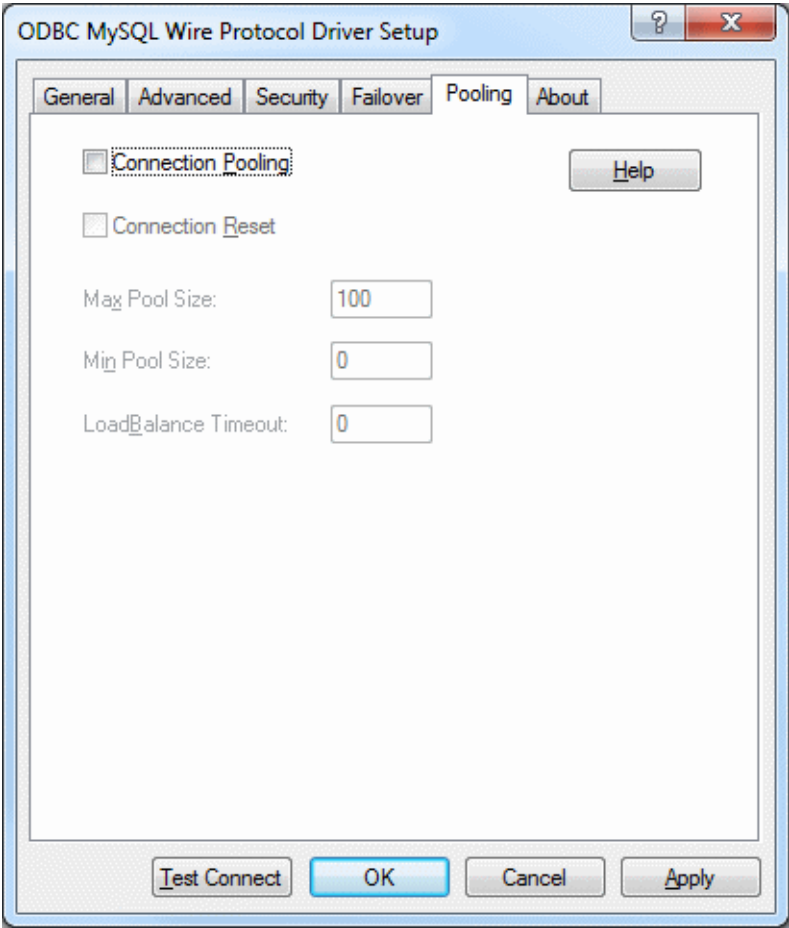
See ["Using Failover" on page 65](#) for a general description of failover and its related connection options.

On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Failover	Default
<a href="#">Load Balancing (see page 229)</a>	Disabled
<a href="#">Connection Retry Count (see page 222)</a>	0
<a href="#">Connection Retry Delay (see page 222)</a>	3
<a href="#">Alternate Servers (see page 220)</a>	None
<a href="#">Failover Mode (see page 225)</a>	0 - Connection

Connection Options: Failover	Default
<a href="#">Failover Granularity (see page 225)</a>	0 - Non-Atomic
<a href="#">Failover Preconnect (see page 226)</a>	Disabled

- 7 Optionally, click the **Pooling** tab to specify connection pooling data source settings.



See ["Using DataDirect Connection Pooling" on page 82](#) for a general description of connection pooling.

On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Pooling	Default
<a href="#">Connection Pooling (see page 221)</a>	Disabled
<a href="#">Connection Reset (see page 221)</a>	Disabled
<a href="#">Max Pool Size (see page 230)</a>	100
<a href="#">Min Pool Size (see page 231)</a>	0
<a href="#">Load Balance Timeout (see page 229)</a>	0



- 8 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 appears; see ["Using a Logon Dialog Box" on page 218](#) for details. 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 displays an appropriate error message.
- Click **OK**.

NOTE: If you are configuring alternate servers for use with the connection failover feature, be aware that the Test Connect button tests only the primary server, not the alternate servers.

- 9 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.

## Using a Connection String

If you want to use a connection string for connecting to a database, or if your application requires it, you must specify either a DSN (data source name), a File DSN, or a DSN-less connection in the string. The difference is whether you use the `DSN=`, `FILEDSN=`, or the `DRIVER=` keyword in the connection string, as described in the ODBC specification. A DSN or FILEDSN connection string tells the driver where to find the default connection information. Optionally, you may specify *attribute=value* pairs in the connection string to override the default values stored in the data source.

The DSN connection string has the form:

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

The FILEDSN connection string has the form:

```
FILEDSN=filename.dsn[;attribute=value[;attribute=value]...]
```

The DSN-less connection string specifies a driver instead of a data source. All connection information must be entered in the connection string because the information is not stored in a data source.

The DSN-less connection string has the form:

```
DRIVER=[{driver_name}] [;attribute=value[;attribute=value]...]
```

[Table 7-1](#) lists the long and short names for each attribute, as well as the initial default value when the driver is first installed. You can specify either long or short names in the connection string.

An example of a DSN connection string with overriding attribute values for MySQL is:

```
DSN=MySQL TABLES;DB=PAYROLL
```

A FILEDSN connection string is similar except for the initial keyword:

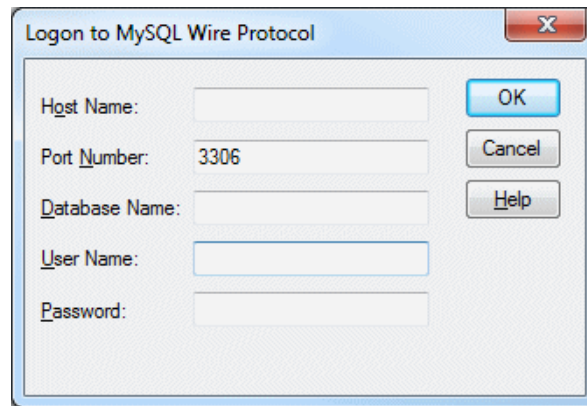
```
FILEDSN=MySQL.dsn;DB=DBPAYROLL
```

A DSN-less connection string must provide all necessary connection information:

```
DRIVER=DataDirect 7.1 MySQL Wire Protocol;HOST=MySQL2;PORT=3306;  
DB=PAYROLL;UID=JOHN;PWD=XYZZY
```

## 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 Host Name field, type either the name or the IP address of the server to which you want to connect. The IP address must be in IPv4 format.
- 2 In the Port Number field, type the port number of the server listener. The default is 3306.
- 3 In the Database Name field, type the name of the database to which you want to connect.
- 4 If required, type your user name as specified on the MySQL server.
- 5 If required, type your password.
- 6 Click **OK** to complete the logon and to update these values in the Registry.

## Connection Options Descriptions

The following connection option descriptions are listed alphabetically by the GUI name that appears on the driver Setup dialog box. The connection string attribute name, along with its short name, is listed immediately underneath the GUI name. For example:

### Application Using Threads

Attribute    ApplicationUsingThreads (AUT)

In most cases, the GUI name and the attribute name are the same; however, some exceptions exist. If you need to look up an option by its connection string attribute name, please refer to the alphabetical table of connection string attribute names.

Also, a few connection string attributes, for example, Password, do not have equivalent options that appear on the GUI. They are in the list of descriptions alphabetically by their attribute names.

[Table 7-1](#) lists the connection string attributes supported by the MySQL Wire Protocol driver.

**Table 7-1. MySQL Wire Protocol Attribute Names**

Attribute (Short Name)	Default
<a href="#">AlternateServers (ASRV)</a>	None
<a href="#">ApplicationUsingThreads (AUT)</a>	1 (Enabled)
<a href="#">ConnectionReset (CR)</a>	0 (Disabled)
<a href="#">ConnectionRetryCount</a>	0
<a href="#">ConnectionRetryDelay</a>	3
<a href="#">Database (DB)</a>	None
<a href="#">DataSourceName (DSN)</a>	None
<a href="#">DefaultLongDataBuffLen (DLDBL)</a>	1024
<a href="#">Description (n/a)</a>	None
<a href="#">EnableDescribeParam (EDP)</a>	0 (Disabled)
<a href="#">EncryptionMethod (EM)</a>	0 (Disabled)
<a href="#">FailoverGranularity (FG)</a>	0 (Non-Atomic)
<a href="#">FailoverMode (FM)</a>	0 (Connection)
<a href="#">FailoverPreconnect (FP)</a>	0 (Disabled)
<a href="#">HostName (HOST)</a>	None
<a href="#">HostNameInCertificate (HNIC)</a>	None
<a href="#">IANAAppCodePage (IACP) UNIX ONLY</a>	4 (ISO 8559-1 Latin-1)
<a href="#">InteractiveClient (IC)</a>	0 (Disabled)
<a href="#">KeyPassword (KP)</a>	None
<a href="#">Keystore (KS)</a>	None
<a href="#">KeystorePassword (KSP)</a>	None
<a href="#">LoadBalanceTimeout (LBT)</a>	0 (Disabled)

**Table 7-1. MySQL Wire Protocol Attribute Names** (cont.)

Attribute (Short Name)	Default
LoadBalancing (LB)	0 (Disabled)
LoginTimeout (LT)	15
LogonID (UID)	None
MaxPoolSize (MXPS)	100
MinPoolSize (MNPS)	0
Password (PWD)	None
Pooling (POOL)	0 (Disabled)
PortNumber (PORT)	3306
QueryTimeout (QT)	0
ReportCodepageConversionErrors (RCCE)	0 (Ignore Errors)
TreatBinaryAsChar (TBAC)	0 (Disabled)
Truststore (TS)	None
TruststorePassword (TSP)	None
ValidateServerCertificate (VSC)	1 (Enabled)

## Alternate Servers

Attribute	AlternateServers (ASRV)
Description	A list of alternate database servers to which the driver tries to connect if the primary database server is unavailable. Specifying a value for this option enables connection failover for the driver. The value you specify must be in the form of a string that defines the physical location of each alternate server. All of the other required connection information for each alternate server is the same as what is defined for the primary server connection.
Valid Values	<p>(Database=databasename:HostName=hostvalue:PortNumber=portvalue[, . . .])</p> <p>You must specify the database name, host name, and port number. The string has the format:</p>
Example	<p>The following Alternate Servers value defines two alternate database servers for connection failover:</p> <pre>(Database=MySQLdb1:HostName=MySQLhost1:PortNumber=5177, Database=MySQLdb2:HostName=MySQLhost2:PortNumber=5178)</pre>
Default	None
GUI Tab	<a href="#">Failover tab</a> on page 215

## Application Using Threads

Attribute	ApplicationUsingThreads (AUT)
Description	<p>Determines whether the driver works with applications using multiple ODBC threads.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 235</a> for details.</p>
Valid Values	<p>0   1</p> <p>If set to 1 (Enabled), the driver works with single-threaded and multi-threaded applications.</p> <p>If set to 0 (Disabled), the driver does not work with multi-threaded applications. If using the driver with single-threaded applications, this value avoids additional processing required for ODBC thread-safety standards.</p>
Default	1 (Enabled)
GUI tab	<a href="#">Advanced tab</a> on page 213

## Connection Pooling

Attribute	Pooling (POOL)
Description	<p>Specifies whether to use the driver's connection pooling.</p> <p>NOTE: The application must be thread-enabled to use connection pooling.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 235</a> for details.</p>
Valid Values	<p>0   1</p> <p>If set to 1 (Enabled), the driver uses connection pooling.</p> <p>If set to 0 (Disabled), the driver does not use connection pooling.</p>
Default	0 (Disabled)
GUI Tab	<a href="#">Pooling tab</a> on page 216

## Connection Reset

Attribute	ConnectionReset (CR)
Description	<p>Determines whether the state of connections that are removed from the connection pool for reuse by the application is reset to the initial configuration of the connection.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 235</a> for details.</p>

Valid Values 0 | 1

If set to 1 (Enabled), the state of connections removed from the connection pool for reuse by an application is reset to the initial configuration of the connection. Resetting the state can negatively impact performance because additional commands must be sent over the network to the server to reset the state of the connection.

If set to 0 (Disabled), the state of connections is not reset.

Default 0 (Disabled)

GUI Tab [Pooling tab](#) on page 216

## Connection Retry Count

Attribute ConnectionRetryCount

Description The number of times the driver retries connection attempts to the primary database server, and if specified, alternate servers until a successful connection is established.

This option and the Connection Retry Delay connection option, which specifies the wait interval between attempts, can be used in conjunction with failover.

Valid Values 0 |  $x$

where  $x$  is a positive integer from 1 to 65535.

If set to 0, the driver does not try to connect after the initial unsuccessful attempt.

If set to  $x$ , the driver retries connection attempts the specified number of times. If a connection is not established during the retry attempts, the driver returns an error that is generated by the last server to which it tried to connect.

Default 0

GUI Tab [Failover tab](#) on page 215

## Connection Retry Delay

Attribute ConnectionRetryDelay

Description The number of seconds the driver waits between connection retry attempts when Connection Retry Count is set to a positive integer.

This option and the Connection Retry Count connection option can be used in conjunction with failover.

Valid Values 0 |  $x$

where  $x$  is a positive integer from 1 to 65535.

If set to 0, there is no delay between retries.

If set to  $x$ , the driver waits the specified number of seconds between connection retry attempts.

Default 3

GUI Tab [Failover tab](#) on page 215

## Data Source Name

Attribute	DataSourceName (DSN)
Description	The name of a data source in your Windows Registry or odbc.ini file.
Valid Values	<i>string</i>  where <i>string</i> is the name of a data source.
Default	None
GUI Tab	<a href="#">General tab</a> on page 212

## Database Name

Attribute	Database (DB)
Description	The name of the database to which you want to connect.
Valid Values	<i>database_name</i>  where <i>database_name</i> is the name of a valid database.
Default	None
GUI Tab	<a href="#">General tab</a> on page 212

## Default Buffer Size for Long/LOB Columns (in Kb)

Attribute	DefaultLongDataBuffLen (DLDBL)
Description	The maximum length of data (in KB) the driver can fetch from long columns in a single round trip and the maximum length of data that the driver can send using the SQL_DATA_AT_EXEC parameter.  This connection option can affect performance. See <a href="#">"Performance Considerations" on page 235</a> for details.
Valid Values	An integer in multiples of 1024  The value must be in multiples of 1024 (for example, 1024, 2048). You need to increase the default value if the total size of any Long data exceeds 1 MB. This value is multiplied by 1024 to determine the total maximum length of fetched data. For example, if you enter a value of 2048, the maximum length of data would be 1024 x 2048, or 2097152 (2 MB).
Default	1024
GUI tab	<a href="#">Advanced tab</a> on page 213

## Description

Attribute	Description (n/a)
Description	An optional long description of a data source. This description is not used as a runtime connection attribute, but does appear in the ODBC.INI section of the Registry and in the <code>odbc.ini</code> file.
Valid Values	<i>string</i>  where <i>string</i> is a description of a data source.
Default	None
GUI Tab	<a href="#">General tab</a> on page 212

## Enable SQLDescribeParam

Attribute	EnableDescribeParam (EDP)
Description	Determines whether the driver uses the SQLDescribeParam function, which describes parameters as a data type of SQL_VARCHAR with a length of 255 for statements.
Valid Values	0   1  If set to 1 (enabled), the SQLDescribeParam function describes parameters as a data type of SQL_VARCHAR with a length of 255 for statements.  If set to 0 (disabled), the SQLDescribeParam function returns the standard ODBC error IM001.
Default	0 (Disabled)
GUI tab	<a href="#">Advanced tab</a> on page 213

## Encryption Method

Attribute	EncryptionMethod (EM)
Description	The method the driver uses to encrypt data sent between the driver and the database server. If the specified encryption method is not supported by the database server, the connection fails and the driver returns an error.  This connection option can affect performance. See <a href="#">"Performance Considerations" on page 235</a> for details.
Valid Values	0   1   2  If set to 0 (No Encryption), data is not encrypted.  If set to 1 (SSL), data is encrypted using SSL. If the server supports protocol negotiation, the driver and server negotiate the use of TLS v1, SSL v3, or SSL v2 in that order.
Default	0 (No Encryption)
GUI Tab	<a href="#">Security tab</a> on page 214



## Failover Granularity

Attribute	FailoverGranularity (FG)
Description	<p>Determines whether the driver fails the entire failover process or continues with the process if errors occur while trying to reestablish a lost connection.</p> <p>This option applies only when Failover Mode is set to 1 (Extended Connection) or 2 (Select).</p> <p>The Alternate Servers option specifies one or multiple alternate servers for failover and is required for all failover methods.</p>
Valid Values	<p>0   1   2   3</p> <p>If set to 0 (Non-Atomic), the driver continues with the failover process and posts any errors on the statement on which they occur.</p> <p>If set to 1 (Atomic) the driver fails the entire failover process if an error is generated as the result of anything other than executing and repositioning a Select statement. If an error is generated as a result of repositioning a result set to the last row position, the driver continues with the failover process, but generates a warning that the Select statement must be reissued.</p> <p>If set to 2 (Atomic Including Repositioning), the driver fails the entire failover process if any error is generated as the result of restoring the state of the connection or the state of work in progress.</p> <p>If set to 3 (Disable Integrity Check), the driver does not verify that the rows that were restored during the failover process match the original rows. This value applies only when Failover Mode is set to 2 (Select).</p>
Default	0 (Non-Atomic)
GUI Tab	<a href="#">Failover tab</a> on page 215

## Failover Mode

Attribute	FailoverMode (FM)
Description	<p>Specifies the type of failover method the driver uses.</p> <p>The Alternate Servers option specifies one or multiple alternate servers for failover and is required for all failover methods.</p>
Valid Values	<p>0   1   2</p> <p>If set to 0 (Connection), the driver provides failover protection for new connections only.</p> <p>If set to 1 (Extended Connection), the driver provides failover protection for new and lost connections, but not any work in progress.</p> <p>If set to 2 (Select), the driver provides failover protection for new and lost connections. In addition, it preserves the state of work performed by the last Select statement executed.</p>
Default	0 (Connection)
GUI Tab	<a href="#">Failover tab</a> on page 215

## Failover Preconnect

Attribute	FailoverPreconnect (FP)
Description	<p>Specifies whether the driver tries to connect to the primary and an alternate server at the same time.</p> <p>This attribute applies only when Failover Mode is set to 1 (Extended Connection) or 2 (Select) and at least one alternate server is specified.</p> <p>The Alternate Servers option specifies one or multiple alternate servers for failover and is required for all failover methods.</p>
Valid Values	<p>0   1</p> <p>If set to 0 (Disabled), the driver tries to connect to an alternate server only when failover is caused by an unsuccessful connection attempt or a lost connection. This value provides the best performance, but your application typically experiences a short wait while the failover connection is attempted.</p> <p>If set to 1 (Enabled), the driver tries to connect to the primary and an alternate server at the same time. This can be useful if your application is time-sensitive and cannot absorb the wait for the failover connection to succeed.</p>
Default	0 (Disabled)
GUI Tab	<a href="#">Failover tab</a> on page 215

## Host

Attribute	HostName (HOST)
Description	The name or the IP address of the server to which you want to connect.
Valid Values	<p><i>server_name</i>   <i>IP_address</i></p> <p>where:</p> <p><i>server_name</i> is the name of the server to which you want to connect.</p> <p><i>IP_address</i> is the IP address of the server to which you want to connect.</p> <p>The IP address must be in IPv4 format.</p>
Default	None
GUI Tab	<a href="#">General tab</a> on page 212

## Host Name In Certificate

Attribute	HostNameInCertificate (HNIC)
Description	A host name for certificate validation when SSL encryption is enabled (Encryption Method=1) and validation is enabled (Validate Server Certificate=1). This option provides additional security against man-in-the-middle (MITM) attacks by ensuring that the server the driver is connecting to is the server that was requested.

Valid Values *host\_name* | *#SERVERNAME#*

where the *host\_name* is the host name specified in the certificate. Consult your SSL administrator for the correct value.

If set to a host name, the driver examines the *subjectAltName* values included in the certificate. If a *dnsName* value is present in the *subjectAltName* values, then the driver compares the value specified for Host Name In Certificate with the *dnsName* value. The connection succeeds if the values match. The connection fails if the Host Name In Certificate value does not match the *dnsName* value.

If no *subjectAltName* values exist or a *dnsName* value is not in the list of *subjectAltName* values, then the driver compares the value specified for Host Name In Certificate with the *commonName* part of the Subject name in the certificate. The *commonName* typically contains the host name of the machine for which the certificate was created. The connection succeeds if the values match. The connection fails if the Host Name In Certificate value does not match the *commonName*. If multiple *commonName* parts exist in the Subject name of the certificate, the connection succeeds if the Host Name In Certificate value matches any of the *commonName* parts.

If set to *#SERVERNAME#*, the driver compares the host server name specified as part of a data source or connection string to the *dnsName* or the *commonName* value.

Default None

GUI Tab [Security tab](#) on page 214

## IANAAppCodePage

Attribute IANAAppCodePage (IACP)

Description An Internet Assigned Numbers Authority (IANA) value. You must specify a value for this option if your application is not Unicode-enabled or if your database character set is not Unicode. Refer to [Chapter 4 “Internationalization, Localization, and Unicode”](#) in the *DataDirect Connect Series for ODBC Reference* for details.

The driver uses the specified IANA code page to convert "W" (wide) functions to ANSI.

The driver and Driver Manager both check for the value of IANAAppCodePage in the following order:

- In the connection string
- In the Data Source section of the system information file (odbc.ini)
- In the ODBC section of the system information file (odbc.ini)

If the driver does not find an IANAAppCodePage value, the driver uses the default value of 4 (ISO 8859-1 Latin-1).

Valid Values *IANA\_code\_page*

where *IANA\_code\_page* is one of the valid values listed in [Chapter 1 “Values for the Attribute IANAAppCodePage”](#) in the *DataDirect Connect Series for ODBC Reference*. The value must match the database character encoding and the system locale.

Default 4 (ISO 8559-1 Latin-1)

GUI tab [Advanced tab](#) on page 213

## Interactive Client

Attribute	InteractiveClient (IC)
Description	Determines how long a connection can be idle before the server disconnects it.  NOTE: The wait_timeout variable controlled by the Interactive Client option is a session variable that can be modified by the application after the connection has been established. The Interactive Client option controls only the initial value of the wait_timeout session variable.
Valid Values	0   1  If set to 1 (Enabled), the driver initializes the wait_time session variable for the connection with the value of the global interactive_timeout variable.  If set to 0 (Disabled), the driver initializes the wait_timeout session variable with the value of the global wait_timeout variable.
Default	0 (Disabled)
GUI tab	<a href="#">Advanced tab</a> on page 213

## Key Password

Attribute	KeyPassword (KP)
Description	The password used to access the individual keys in the keystore file when SSL is enabled (Encryption Method=1) and SSL client authentication is enabled on the database server. Keys stored in a keystore can be individually password-protected. To extract the key from the keystore, the driver must have the password of the key.
Valid Values	<i>key_password</i>  where <i>key_password</i> is the password of a key in the keystore.
Default	None
GUI Tab	<a href="#">Security tab</a> on page 214

## Keystore

Attribute	Keystore (KS)
Description	The name of the directory containing the keystore file to be used when SSL is enabled (Encryption Method=1) and SSL client authentication is enabled on the database server. The keystore file contains the certificates that the client sends to the server in response to the server's certificate request. If you do not specify a directory, the current directory is used.  NOTE: The keystore and truststore files may be the same file.
Valid Values	<i>keystore_directory</i>  where <i>keystore_directory</i> is the location of the keystore file.
Default	None
GUI Tab	<a href="#">Security tab</a> on page 214

## Keystore Password

Attribute	KeystorePassword (KSP)
Description	<p>The password used to access the keystore file when SSL is enabled (Encryption Method=1) and SSL client authentication is enabled on the database server. The keystore file contains the certificates that the client sends to the server in response to the server's certificate request.</p> <p>NOTE: The keystore and truststore files may be the same file; therefore, they may have the same password.</p>
Valid Values	<p><i>keystore_password</i></p> <p>where <i>keystore_password</i> is the password of the keystore file.</p>
Default	None
GUI Tab	<a href="#">Security tab</a> on page 214

## Load Balance Timeout

Attribute	LoadBalanceTimeout (LBT)
Description	<p>The number of seconds to keep inactive connections open in a connection pool. An inactive connection is a database session that is not associated with an ODBC connection handle, that is, a connection in the pool that is not in use by an application.</p> <p>NOTE: The Min Pool Size option may cause some connections to ignore this value.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 235</a> for details.</p>
Valid Values	<p>0   <i>x</i></p> <p>where <i>x</i> is a positive integer that specifies a number of seconds.</p> <p>If set to 0, inactive connections are kept open.</p> <p>If set to <i>x</i>, inactive connections are closed after the specified number of seconds passes.</p>
Default	0 (Disabled)
GUI Tab	<a href="#">Pooling tab</a> on page 216

## Load Balancing

Attribute	LoadBalancing (LB)
Description	Determines whether the driver uses client load balancing in its attempts to connect to the database servers (primary and alternate). You can specify one or multiple alternate servers by setting the Alternate Servers option.
Valid Values	<p>0   1</p> <p>If set to 1 (Enabled), the driver uses client load balancing and attempts to connect to the database servers (primary and alternate servers) in random order.</p>

If set to 0 (Disabled), the driver does not use client load balancing and connects to each server based on their sequential order (primary server first, then, alternate servers in the order they are specified).

NOTE: This option has no effect unless alternate servers are defined for the Alternate Servers connection option.

- Default 0 (Disabled)
- GUI Tab [Failover tab](#) on page 215

Login Timeout

- Attribute LoginTimeout (LT)
- Description The number of seconds the driver waits for a connection to be established before returning control to the application and generating a timeout error. To override the value that is set by this connection option for an individual connection, set a different value in the SQL\_ATTR\_LOGIN\_TIMEOUT connection attribute using the SQLSetConnectAttr() function.
- Valid Values -1 | 0 | x

where x is a positive integer that specifies a number of seconds.

If set to -1, the connection request does not time out. The driver silently ignores the SQL\_ATTR\_LOGIN\_TIMEOUT attribute.

If set to 0, the connection request does not time out, but the driver responds to the SQL\_ATTR\_LOGIN\_TIMEOUT attribute.

If set to x, the connection request times out after the specified number of seconds unless the application overrides this setting with the SQL\_ATTR\_LOGIN\_TIMEOUT attribute.

- Default 15
- GUI tab [Advanced tab](#) on page 213

Max Pool Size

- Attribute MaxPoolSize (MXPS)
- Description The maximum number of connections allowed within a single connection pool. When the maximum number of connections is reached, no additional connections can be created in the connection pool.  
  
This connection option can affect performance. See ["Performance Considerations" on page 235](#) for details.
- Valid Values An integer from 1 to 65535

For example, if set to 20, the maximum number of connections allowed in the pool is 20.

- Default 100
- GUI Tab [Pooling tab](#) on page 216

## Min Pool Size

Attribute	MinPoolSize (MNPS)
Description	<p>The minimum number of connections that are opened and placed in a connection pool, in addition to the active connection, when the pool is created. The connection pool retains this number of connections, even when some connections exceed their Load Balance Timeout value.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 235</a> for details.</p>
Valid Values	<p>0   <i>x</i></p> <p>where <i>x</i> is an integer from 1 to 65535.</p> <p>For example, if set to 5, the start-up number of connections in the pool is 5 in addition to the current existing connection.</p> <p>If set to 0, no connections are opened in addition to the current existing connection.</p>
Default	0
GUI Tab	<a href="#">Pooling tab</a> on page 216

## Password

Attribute	Password (PWD)
Description	The password that the application uses to connect to your database. The Password option cannot be specified through the driver Setup dialog box and should not be stored in a data source. It is specified through the Logon dialog box or a connection string.
Valid Values	<p><i>pwd</i></p> <p>where <i>pwd</i> is a valid password.</p>
Default	None
GUI Tab	n/a

## Port Number

Attribute	PortNumber (PORT)
Description	<p>The port number of the server listener.</p> <p>NOTE: This option is mutually exclusive with the Server Name and TNSNames File options.</p>
Valid Values	<p><i>port_name</i></p> <p>where the <i>port_name</i> is the port number of the server listener. Check with your database administrator for the correct number.</p>
Default	3306
GUI Tab	<a href="#">General tab</a> on page 212

## Query Timeout

Attribute	QueryTimeout (QT)
Description	The number of seconds for the default query timeout for all statements that are created by a connection. To override the value set by this connection option for an individual statement, set a different value in the SQL_ATTR_QUERY_TIMEOUT statement attribute on the SQLSetStmtAttr() function.
Valid Values	-1   0   <i>x</i>  where <i>x</i> is a positive integer that specifies a number of seconds.  If set to -1, the query does not time out. The driver silently ignores the SQL_ATTR_QUERY_TIMEOUT attribute.  If set to 0, the query does not time out, but the driver responds to the SQL_ATTR_QUERY_TIMEOUT attribute.  If set to <i>x</i> , all queries time out after the specified number of seconds unless the application overrides this value by setting the SQL_ATTR_QUERY_TIMEOUT attribute.
Default	0
GUI tab	<a href="#">Advanced tab</a> on page 213

## Report Codepage Conversion Errors

Attribute	ReportCodepageConversionErrors (RCCE)
Description	Specifies how the driver handles code page conversion errors that occur when a character cannot be converted from one character set to another.  An error message or warning can occur if an ODBC call causes a conversion error, or if an error occurs during code page conversions to and from the database or to and from the application. The error or warning generated is <code>Code page conversion error encountered</code> . In the case of parameter data conversion errors, the driver adds the following sentence: <code>Error in parameter <i>x</i></code> , where <i>x</i> is the parameter number. The standard rules for returning specific row and column errors for bulk operations apply.
Valid Values	0   1   2  If set to 0 (Ignore Errors), the driver substitutes 0x1A for each character that cannot be converted and does not return a warning or error.  If set to 1 (Return Error), the driver returns an error instead of substituting 0x1A for unconverted characters.  If set to 2 (Return Warning), the driver substitutes 0x1A for each character that cannot be converted and returns a warning.
Default	0 (Ignore Errors)
GUI tab	<a href="#">Advanced tab</a> on page 213



## Treat Binary Data as Character Data

Attribute	TreatBinaryAsChar (TBAC)
Description	Allows data that MySQL stores as BINARY or VARBINARY to be described and returned as CHAR or VARCHAR values, respectively.
Valid Values	0   1  If set to 1 (Enabled), the driver describes and returns data that MySQL stores as BINARY or VARBINARY as CHAR or VARCHAR values, respectively.  If set to 0 (Disabled), the driver describes and returns data that MySQL describes as BINARY or VARBINARY as BINARY or VARBINARY values, respectively.
Example	Create the following MySQL table:  <pre>CREATE TABLE binTable (col1 binary(3))</pre> Then, execute the following Insert statement:  <pre>INSERT INTO binTable values('abc')</pre> Then, execute the following query:  <pre>SELECT col1 FROM binTable</pre> Using this example, the driver would return the value of col1 as a CHAR value, "abc", instead of a BINARY value "616263".
Default	0 (Disabled)
GUI tab	<a href="#">Advanced tab</a> on page 213

## Truststore

Attribute	Truststore (TS)
Description	The directory that contains the truststore file and the truststore file name to be used when SSL is enabled (Encryption Method=1) and server authentication is used. The truststore file contains a list of the valid Certificate Authorities (CAs) that are trusted by the client machine for SSL server authentication. If you do not specify a directory, the current directory is used.  NOTE: The truststore and keystore files may be the same file.
Valid Values	<i>truststore_directory\filename</i>  where <i>truststore_directory</i> is the directory where the truststore file is located and <i>filename</i> is the file name of the truststore file.
Default	None
GUI Tab	<a href="#">Security tab</a> on page 214

## Truststore Password

Attribute	TruststorePassword (TSP)
Description	The password that is used to access the truststore file when SSL is enabled (Encryption Method=1) and server authentication is used. The truststore file contains a list of the Certificate Authorities (CAs) that the client trusts.  NOTE: The truststore and keystore files may be the same file; therefore, they may have the same password.
Valid Values	<i>truststore_password</i>  where <i>truststore_password</i> is a valid password for the truststore file.
Default	None
GUI Tab	<a href="#">Security tab</a> on page 214

## User Name

Attribute	LogonID (UID)
Description	The default user ID that is used to connect to your database. Your ODBC application may override this value or you may override it in the logon dialog box or connection string.
Valid Values	<i>userid</i>  where <i>userid</i> is a valid user ID with permissions to access the database.
Default	None
GUI Tab	<a href="#">Security tab</a> on page 214

## Validate Server Certificate

Attribute	ValidateServerCertificate (VSC)
Description	Determines whether the driver validates the certificate that is sent by the database server when SSL encryption is enabled (Encryption Method=1). When using SSL server authentication, any certificate sent by the server must be issued by a trusted Certificate Authority (CA). Allowing the driver to trust any certificate returned from the server even if the issuer is not a trusted CA is useful in test environments because it eliminates the need to specify truststore information on each client in the test environment.  Truststore information is specified using the Trust Store and Trust Store Password options.
Valid Values	0   1  If set to 1 (Enabled), the driver validates the certificate that is sent by the database server. Any certificate from the server must be issued by a trusted CA in the truststore file. If the Host Name In Certificate option is specified, the driver also validates the certificate using a host name. The Host Name In Certificate option provides additional security against man-in-the-middle (MITM) attacks by ensuring that the server the driver is connecting to is the server that was requested.

If set to 0 (Disabled), the driver does not validate the certificate that is sent by the database server. The driver ignores any truststore information specified by the Trust Store and Trust Store Password options.

Default 1 (Enabled)

GUI Tab [Security tab](#) on page 214

---

## Performance Considerations

The following connection options can enhance driver performance. You can also enhance performance through efficient application design. Refer to [Chapter 5 “Designing ODBC Applications for Performance Optimization”](#) in the *DataDirect Connect Series for ODBC Reference* for details.

The option names found on the tabs of the driver Setup dialog box are the same as the connection string attribute names unless otherwise noted in parentheses. The connection string attribute name does not have spaces between the words. For example, the option name Application Using Threads is equivalent to the connection string attribute name ApplicationUsingThreads.

**Application Using Threads (ApplicationUsingThreads):** The driver coordinates concurrent database operations (operations from different threads) by acquiring locks. Although locking prevents errors in the driver, it also decreases performance. If your application does not make ODBC calls from different threads, the driver has no reason to coordinate operations. In this case, the ApplicationUsingThreads attribute should be disabled (set to 0).

NOTE: If you are using a multi-threaded application, you must enable the Application Using Threads option.

**Default Buffer Size for Long/LOB Columns (DefaultLongDataBuffLen):** To improve performance when your application fetches images, pictures, or long text or binary data, a buffer size can be set to accommodate the maximum size of the data. The buffer size should only be large enough to accommodate the maximum amount of data retrieved; otherwise, performance is reduced by transferring large amounts of data into an oversized buffer. If your application retrieves more than 1 MB of data, the buffer size should be increased accordingly.

**Connection Pooling (ConnectionPooling):** If you enable the driver to use connection pooling, you can set additional options that affect performance:

- **Load Balance Timeout:** You can define how long to keep connections in the pool. The time that a connection was last used is compared to the current time and, if the timespan exceeds the value of the Load Balance Timeout option, the connection is destroyed. The Min Pool Size option can cause some connections to ignore this value.
- **Connection Reset:** Resetting a re-used connection to the initial configuration settings impacts performance negatively because the connection must issue additional commands to the server.

- **Max Pool Size:** Setting the maximum number of connections that the pool can contain too low might cause delays while waiting for a connection to become available. Setting the number too high wastes resources.
- **Min Pool Size:** A connection pool is created when the first connection with a unique connection string connects to the database. The pool is populated with connections up to the minimum pool size, if one has been specified. The connection pool retains this number of connections, even when some connections exceed their Load Balance Timeout value.

**Encryption Method (EncryptionMethod):** Data encryption may adversely affect performance because of the additional overhead (mainly CPU usage) that is required to encrypt and decrypt data.

**Failover Mode (FailoverMode):** Although high availability that replays queries after a failure provides increased levels of protection, it can adversely affect performance because of increased overhead.

---

# Data Types

Table 7-2 shows how the MySQL data types map to the standard ODBC data types.

Table 7-2. MySQL Data Types	
MySQL	ODBC
BIGINT	SQL_BIGINT
BIGINT UNSIGNED	SQL_BIGINT
BINARY	SQL_BINARY
BIT	SQL_BINARY
BLOB	SQL_LONGVARBINARY
CHAR	SQL_CHAR
DATE	SQL_TYPE_DATE
DATETIME	SQL_TYPE_TIMESTAMP
DECIMAL	SQL_DECIMAL
DECIMAL UNSIGNED	SQL_DECIMAL
DOUBLE	SQL_DOUBLE
DOUBLE UNSIGNED	SQL_DOUBLE
FLOAT	SQL_REAL
FLOAT UNSIGNED	SQL_REAL
INTEGER	SQL_INTEGER
INTEGER UNSIGNED	SQL_INTEGER
LOBLOB	SQL_LONGVARBINARY
LONGTEXT	SQL_LONGVARCHAR
MEDIUMBLOB	SQL_LONGVARBINARY

**Table 7-2. MySQL Data Types** (cont.)

MySQL	ODBC
MEDIUMINT	SQL_INTEGER
MEDIUMINT UNSIGNED	SQL_INTEGER
MEDIUMTEXT	SQL_LONGVARCHAR
SMALLINT	SQL_SMALLINT
SMALLINT UNSIGNED	SQL_SMALLINT
TEXT	SQL_LONGVARCHAR
TIME	SQL_TYPE_TIME
TIMESTAMP	SQL_TYPE_TIMESTAMP
TINYBLOB	SQL_LONGVARBINARY
TINYINT	SQL_TINYINT
TINYINT UNSIGNED	SQL_TINYINT
TINYTEXT	SQL_LONGVARCHAR
VARBINARY	SQL_VARBINARY
VARCHAR	SQL_VARCHAR
YEAR	SQL_SMALLINT

See ["Retrieving Data Type Information" on page 59](#) for more information about data types.

NOTE: The Treat Binary Data as Character Data connection option affects how certain ODBC data types are reported. See ["Treat Binary Data as Character Data" on page 233](#) for details.

## Unicode Support

When the character set of a character column is Unicode, then the MySQL Wire Protocol driver maps the MySQL data type to Unicode data type as follows:

MySQL Data Type	Mapped to . . .
CHAR	SQL_WCHAR
LONGTEXT	SQL_WLONGVARCHAR
MEDIUMTEXT	SQL_WLONGVARCHAR
TEXT	SQL_WLONGVARCHAR
TINYTEXT	SQL_WLONGVARCHAR
VARCHAR	SQL_WVARCHAR

---

## Advanced Features

The driver supports the following advanced features:

- Failover
- Security
- Connection Pooling

### Failover

The driver supports failover and its related connection options. Failover connection options are located on the [Failover tab](#) of the driver Setup dialog box. See ["Using Failover" on page 65](#) for a general description of failover and its implementation.

### Security

The driver supports authentication and encryption. Security connection options are located on the [Security tab](#) of the driver Setup dialog box. See ["Using Security" on page 77](#) for a general description of security and its implementation. The following security information is specific to the DB2 Wire Protocol Driver.

### Connection Pooling

The driver supports connection pooling and its related connection options. Connection pooling connection options are located on the [Pooling tab](#) of the driver Setup dialog box. See ["Using DataDirect Connection Pooling" on page 82](#) for a general description of connection pooling and its implementation.

---

## Persisting a Result Set as an XML Data File

The driver allows you to persist a result as an XML data file with embedded schema. See ["Persisting a Result Set as an XML Data File" on page 60](#) for details about implementation.

---

## Isolation and Lock Levels Supported

MySQL supports isolation levels 0 (read uncommitted), 1 (read committed), 2 (repeatable read), and 3 (serializable). The default is 1.

MySQL supports record-level locking.

Refer to [Chapter 7 "Locking and Isolation Levels"](#) of the *DataDirect Connect Series for ODBC Reference* for details.

---

## SQL Support

The driver supports the minimum SQL grammar.

---

## ODBC Conformance Level

The driver is Level 1 compliant, that is, it supports all ODBC Core and Level 1 functions.

In addition, the following functions are supported:

- SQLColumnPrivileges
- SQLForeignKeys
- SQLTablePrivileges

Refer to [Chapter 2 “ODBC API and Scalar Functions”](#) of the *DataDirect Connect Series for ODBC Reference* for a list of the API functions supported by the MySQL Wire Protocol driver.

---

## Number of Connections and Statements Supported

The MySQL Wire Protocol driver supports multiple connections and multiple statements per connection to the MySQL database system.





## 8 The Oracle Wire Protocol Driver

The DataDirect Connect *for* ODBC and DataDirect Connect64 *for* ODBC Oracle Wire Protocol driver (the Oracle Wire Protocol driver) each support the following Oracle database servers:

- Oracle 11g R1, R2 (11.1, 11.2)
- Oracle 10g R1, R2 (10.1, 10.2)
- Oracle 9i R1, R2 (9.0.1, 9.2)
- Oracle 8i R3 (8.1.7)

The Oracle Wire Protocol driver is supported in the Windows, UNIX, and Linux environments. See ["Environment-Specific Information" on page 45](#) for detailed information about the Windows, UNIX, and Linux environments supported by this driver.

Refer to the readme file shipped with your DataDirect Connect product for the file name of the Oracle Wire Protocol driver.

NOTE: The Oracle Wire Protocol driver does not require any Oracle client software. Progress DataDirect also provides an Oracle client-based driver; see ["Chapter 13 "The Oracle Driver" on page 477](#) for details.

---

### Driver Requirements

The driver has no client requirements.

---

### Configuring and Connecting to Data Sources

After you install the driver, you configure data sources to connect to the database. See [Chapter 1 "Quick Start Connect" on page 35](#) for an explanation of different types of data sources. The data source contains connection options that allow you to tune the driver for specific performance. If you want to use a data source but need to change some of its values, you can either modify the data source or override its values at connection time through a connection string.

If you choose to use a connection string, you must use specific connection string attributes. See ["Using a Connection String" on page 261](#) and [Table 8-1 on page 264](#) for an alphabetical list of driver connection string attributes and their initial default values.



## Data Source Configuration in the UNIX/Linux odbc.ini File

On UNIX and Linux, you must set up the proper ODBC environment before configuring data sources. See ["Environment Configuration" on page 37](#) for basic setup information and ["Environment Variables" on page 97](#) for more detail about this procedure.

Data sources for UNIX and Linux are stored in the system information file (by default, odbc.ini). If you have a Motif GUI environment on Linux, you can configure and modify data sources through the DataDirect ODBC Data Source Administrator for Linux (the Linux ODBC Administrator) using a driver Setup dialog box. (See ["Configuration Through the Administrator" on page 100](#) for a detailed explanation of the Administrator.)

If you do not have a GUI environment, you can configure and modify data sources directly by editing the odbc.ini file and storing default connection values there. See ["Configuration Through the System Information \(odbc.ini\) File" on page 103](#) for detailed information about the specific steps necessary to configure a data source.

[Table 8-1 on page 264](#) lists driver connection string attributes that must be used in the odbc.ini file to set the value of the attributes. Note that only the long name of the attribute can be used in the file. The default listed in the table is the initial default value when the driver is installed.

## Data Source Configuration through a GUI



On Windows, data sources are stored in the Windows Registry. You can configure and modify data sources through the ODBC Administrator using a driver Setup dialog box, as described in this section.



On UNIX and Linux, data sources are stored in the odbc.ini file. On Linux, you can configure and modify data sources through the Linux ODBC Administrator using a driver Setup dialog box, as described in this section.

**NOTE:** This book shows dialog box images that are specific to Windows. If you are using the drivers in the Linux environment, the dialog box that you see may differ slightly from the Windows version. Windows-only and UNIX-only connection options are specifically noted by icons in the Setup dialog box descriptions.

When the driver is first installed, the values of its connection options are set by default. These values appear on the driver Setup dialog box tabs when you create a new data source. You can change these default values by modifying the data source. In the following procedure, the description of each tab is followed by a table that lists the connection options for that tab and their initial default values. This table links you to a complete description of the options and their connection string attribute equivalents. The connection string attributes are used to override the default values of the data source if you want to change these values at connection time.

**To configure an Oracle data source:****1 Start the ODBC Administrator:**

- On Windows, start the ODBC Administrator by selecting its icon from the DataDirect Connect program group.



- On Linux, change to the *install\_dir/tools* directory and, at a command prompt, enter:

```
odbcadmin
```

where *install\_dir* is the path to the product installation directory.

**2 Select a tab:**

- **User DSN:** If you are configuring an existing user data source, select the data source name and click **Configure** to display the driver Setup dialog box.

If you are configuring a new user data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.



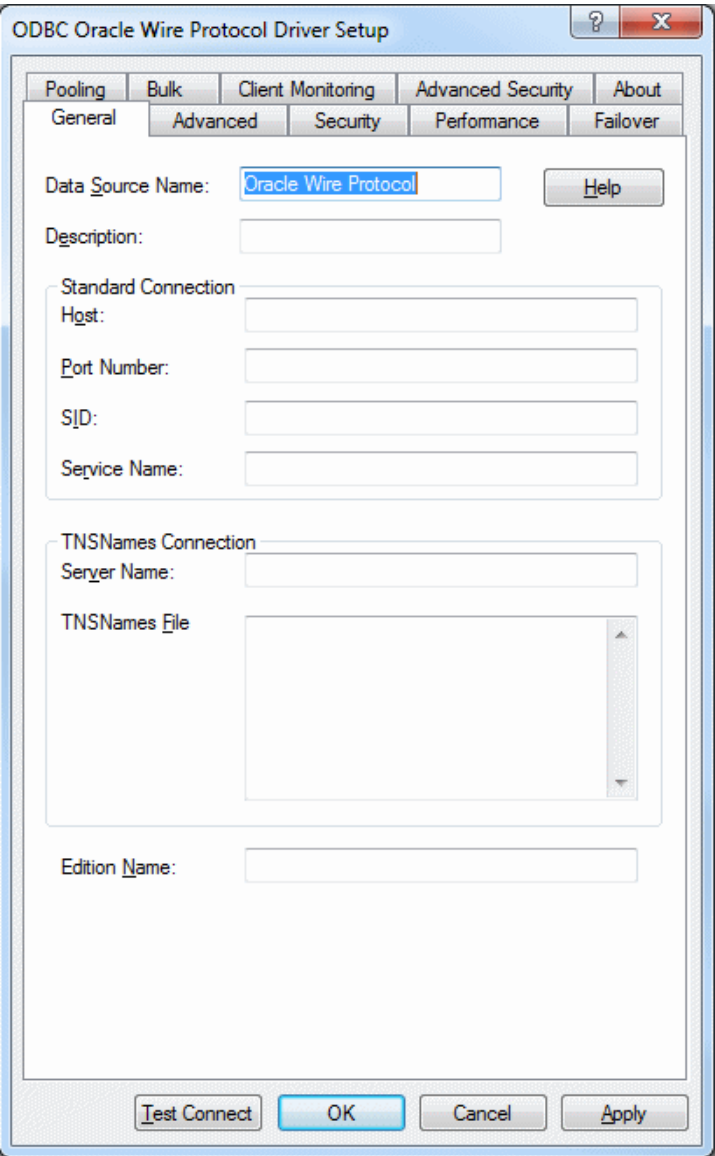
- **System DSN:** If you are configuring an existing system data source, select the data source name and click **Configure** to display the driver Setup dialog box.

If you are configuring a new system data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.

- **File DSN:** If you are configuring an existing file data source, select the data source file and click **Configure** to display the driver Setup dialog box.

If you are configuring a new file data source, click **Add** to display a list of installed drivers; then, select a driver. Click **Advanced** if you want to specify attributes; otherwise, click **Next** to proceed. Specify a name for the data source and click **Next**. Verify the data source information; then, click **Finish** to display the driver Setup dialog box.

The General tab of the Setup dialog box appears by default.



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 this tab, provide values for the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: General	Default
<a href="#">Data Source Name (see page 276)</a>	None
<a href="#">Description (see page 277)</a>	None
<a href="#">Host (see page 285)</a>	None
<a href="#">Port Number (see page 292)</a>	None
<a href="#">SID (see page 296)</a>	None
<a href="#">Service Name (see page 296)</a>	None
<a href="#">Server Name (see page 295)</a>	None

Connection Options: General	Default
<a href="#">TNSNames File (see page 297)</a>	None
<a href="#">Edition Name (see page 277)</a>	None

- 4 Optionally, click the **Advanced** tab to specify additional data source settings.

The screenshot shows the 'ODBC Oracle Wire Protocol Driver Setup' dialog box with the 'Advanced' tab selected. The dialog has several tabs: Pooling, Bulk, Client Monitoring, Advanced Security, About, General, Advanced, Security, Performance, and Failover. The 'Advanced' tab is active, displaying various configuration options. At the bottom, there are buttons for 'Test Connect', 'OK', 'Cancel', and 'Apply'.

Options visible in the Advanced tab:

- Local Timezone Offset: [text box]
- ☐ Enable Timestamp With Timezone
- Default Buffer Size for Long/LOB Columns (in Kb): 1024
- ☒ Application Using Threads
- ☐ Describe at Prepare
- ☐ Catalog Options
- ☐ Enable N-CHAR Support
- ☐ Enable SQLDescribeParam
- ☐ Report Recycle Bin
- ☐ Procedure Returns Results
- ☐ Enable Server Result Cache
- ☐ Fetch TSWTZ as Timestamp:
- Timestamp Escape Mapping: 0 - Version Specific
- Report Codepage Conversion Errors: 0 - Ignore Errors
- Server Process Type: 0 - Server Default
- Initialization String: [text box]
- Login Timeout: 15
- Query Timeout: 0
- Extended Options: [text box]

On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Advanced	Default
<a href="#">Local Timezone Offset (see page 289)</a>	None
<a href="#">Enable Timestamp with Timezone (see page 280)</a>	Disabled

Connection Options: Advanced	Default
Default Buffer Size for Long/LOB Columns (in Kb) (see page 276)	1024
Application Using Threads (see page 268)	Enabled
Catalog Options (see page 272)	Disabled
Enable SQLDescribeParam (see page 279)	Disabled
Procedure Returns Results (see page 292)	Disabled
Describe at Prepare (see page 277)	Disabled
Enable N-CHAR Support (see page 278)	Disabled
Report Recycle Bin (see page 294)	Disabled
Enable Server Result Cache (see page 279)	Disabled
Fetch TSWTZ as Timestamp (see page 284)	Disabled
Timestamp Escape Mapping (see page 296)	0 - Version Specific
Report Codepage Conversion Errors (see page 294)	0 - Ignore Errors
Server Process Type (see page 295)	0 - Server Default
Initialization String (see page 287)	None
Login Timeout (see page 290)	15
Query Timeout (see page 293)	0
IANAAppCodePage (see page 286)	4 (ISO 8559-1 Latin-1)
UNIX ONLY	



**Extended Options:** Type a semi-colon separated list of connection options and their values. Use this configuration option to set the value of undocumented connection options that are provided by Progress DataDirect customer support. You can include any valid connection option in the Extended Options string, for example:

```
Database=Server1;UndocumentedOption1=value[;UndocumentedOption2=value;]
```

If the Extended Options string contains option values that are also set in the setup dialog or data source, the values of the options specified in the Extended Options string take precedence. However, connection options that are specified on a connection string override any option value specified in the Extended Options string.

NOTE: Do not specify the Extended Options configuration option in a connection string, or the driver will return an error. Instead, applications should specify the individual undocumented connection options in the connection string.



**Translate:** Click **Translate** to display the Select Translator dialog box, which lists the translators specified in the ODBC Translators section of the Registry. Progress 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.

- 5 Optionally, click the **Security** tab to specify security data source settings.

The screenshot shows the 'ODBC Oracle Wire Protocol Driver Setup' dialog box with the 'Security' tab selected. The dialog has a title bar with a question mark and a close button. Below the title bar are tabs for 'Pooling', 'Bulk', 'Client Monitoring', 'Advanced Security', and 'About'. Under 'Advanced Security', there are sub-tabs: 'General', 'Advanced', 'Security' (selected), 'Performance', and 'Failover'. A 'Help' button is located in the top right of the main area. The 'Authentication' section contains a 'User Name' text box, an 'Authentication Method' dropdown menu set to '1 - Encrypt Password', and a 'GSS Client Library' text box set to 'native'. The 'Encryption' section contains an 'Encryption Method' dropdown menu set to '0 - No Encryption', a checked 'Validate Server Certificate' checkbox, and text boxes for 'Trust Store', 'Trust Store Password', 'Key Store', 'Key Store Password', 'Key Password', and 'Host Name In Certificate'. At the bottom are buttons for 'Test Connect', 'OK', 'Cancel', and 'Apply'.

See ["Using Security" on page 77](#) for a general description of authentication and encryption and their configuration requirements.

See ["OS Authentication" on page 310](#) for a discussion of Oracle and SSL encryption.

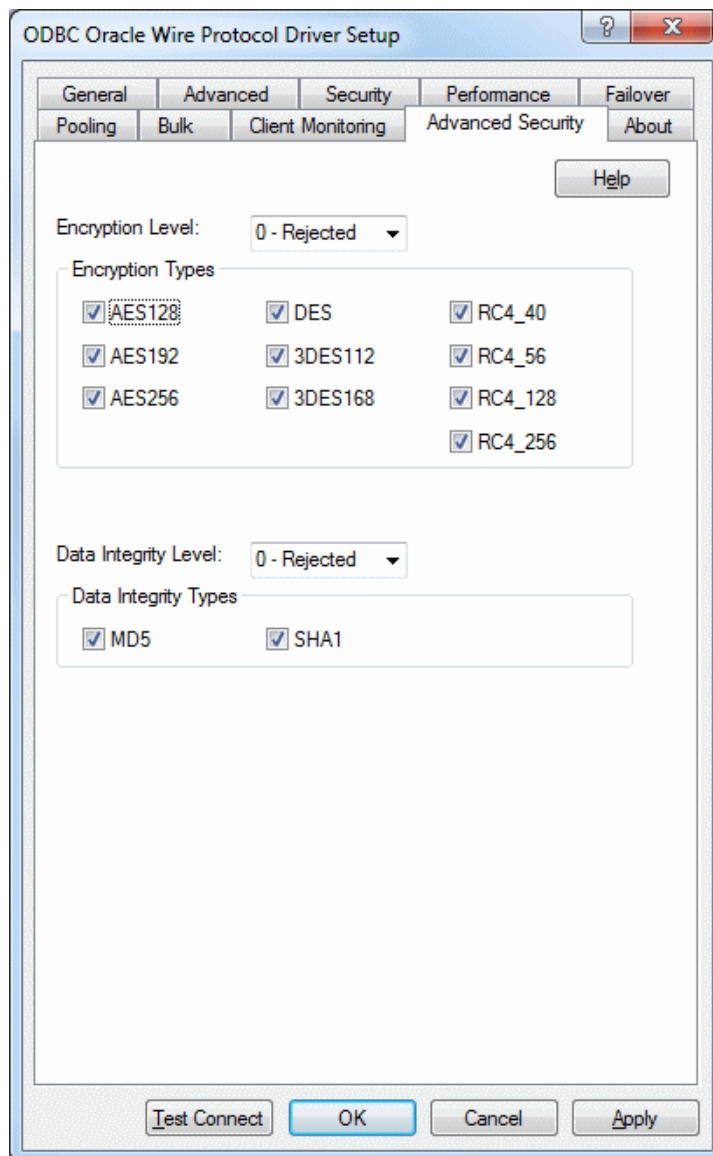
On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Security	Default
<a href="#">User Name (see page 299)</a>	None
<a href="#">Authentication Method (see page 269)</a>	1 - Encrypt Password

**Connection Options: Security**

GSS Client Library (see page 285)	native
Encryption Method (see page 281)	0 - No Encryption
Validate Server Certificate (see page 299)	Enabled
Truststore (see page 298)	None
Truststore Password (see page 298)	None
Keystore (see page 287)	None
Keystore Password (see page 288)	None
Key Password (see page 287)	None
Host Name In Certificate (see page 286)	None

- 6 Optionally, click the **Advanced Security** tab to specify settings for Oracle Advanced Security (OAS).





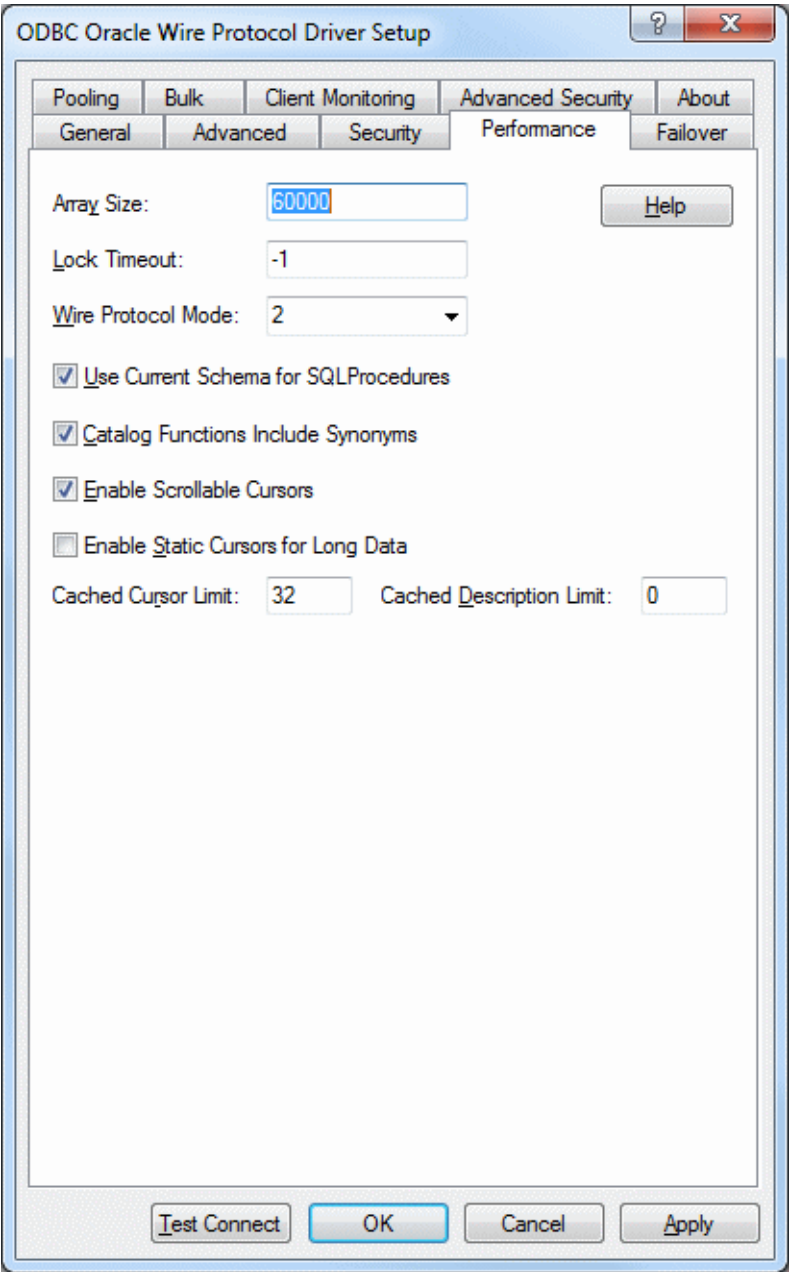
See ["Encryption and Data Integrity" on page 308](#) for a general description of encryption configuration.

Refer to your Oracle documentation for a discussion of Oracle Advanced Security.

On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Security	Default
<a href="#">Encryption Level (see page 281)</a>	0 - Rejected
<a href="#">Encryption Types (see page 282)</a>	All listed encryption algorithms are selected
<a href="#">Data Integrity Level (see page 275)</a>	0 - Rejected
<a href="#">Data Integrity Types (see page 276)</a>	SHA1 and MD5 are selected

- 7 Optionally, click the **Performance** tab to specify performance data source settings.



On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Performance	Default
<a href="#">Array Size (see page 268)</a>	60000
<a href="#">Lock Timeout (see page 289)</a>	-1
<a href="#">Wire Protocol Mode (see page 300)</a>	2
<a href="#">Use Current Schema for SQLProcedures (see page 298)</a>	Enabled
<a href="#">Catalog Functions Include Synonyms (see page 271)</a>	Enabled
<a href="#">Enable Scrollable Cursors (see page 279)</a>	Enabled

**Connection Options: Performance****Default**

Enable Static Cursors for Long Data (see page 280)

Disabled

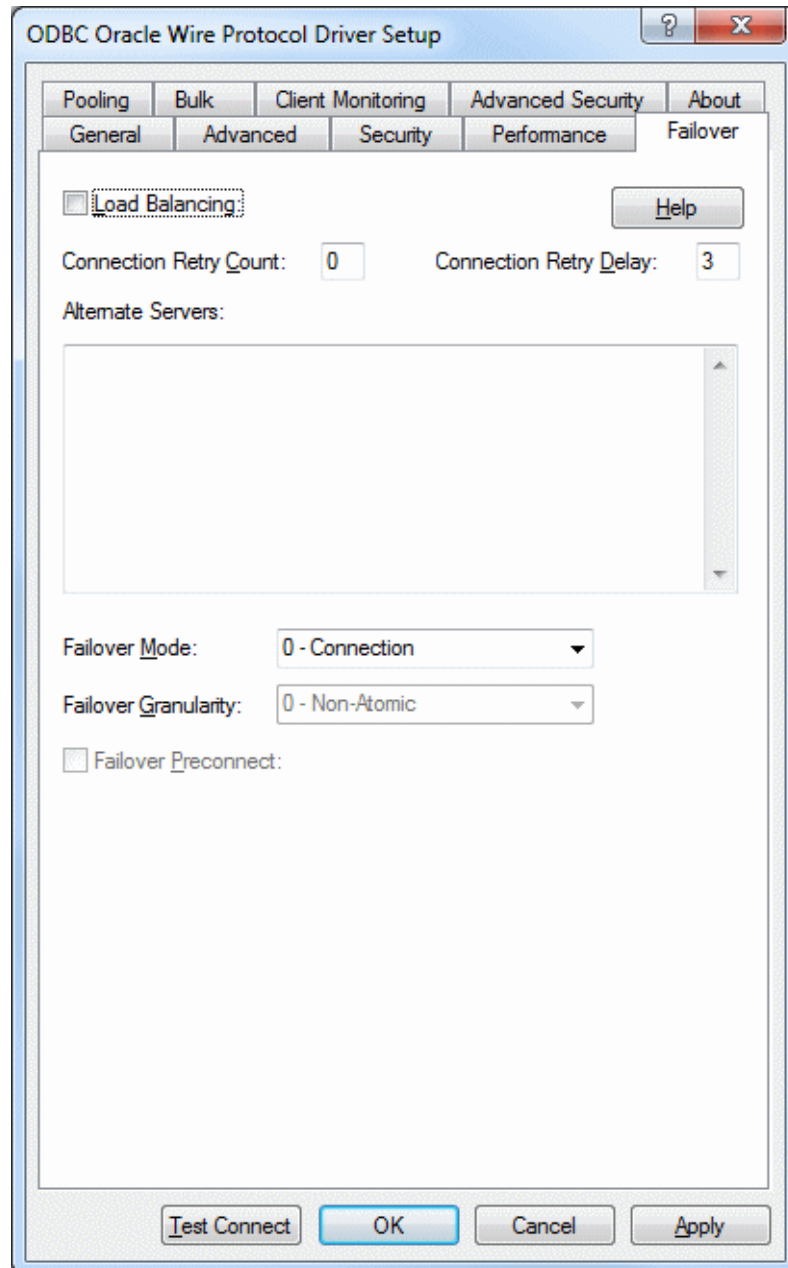
Cached Cursor Limit (see page 271)

32

Cached Description Limit (see page 271)

0

- 8 Optionally, click the **Failover** tab to specify failover data source settings.

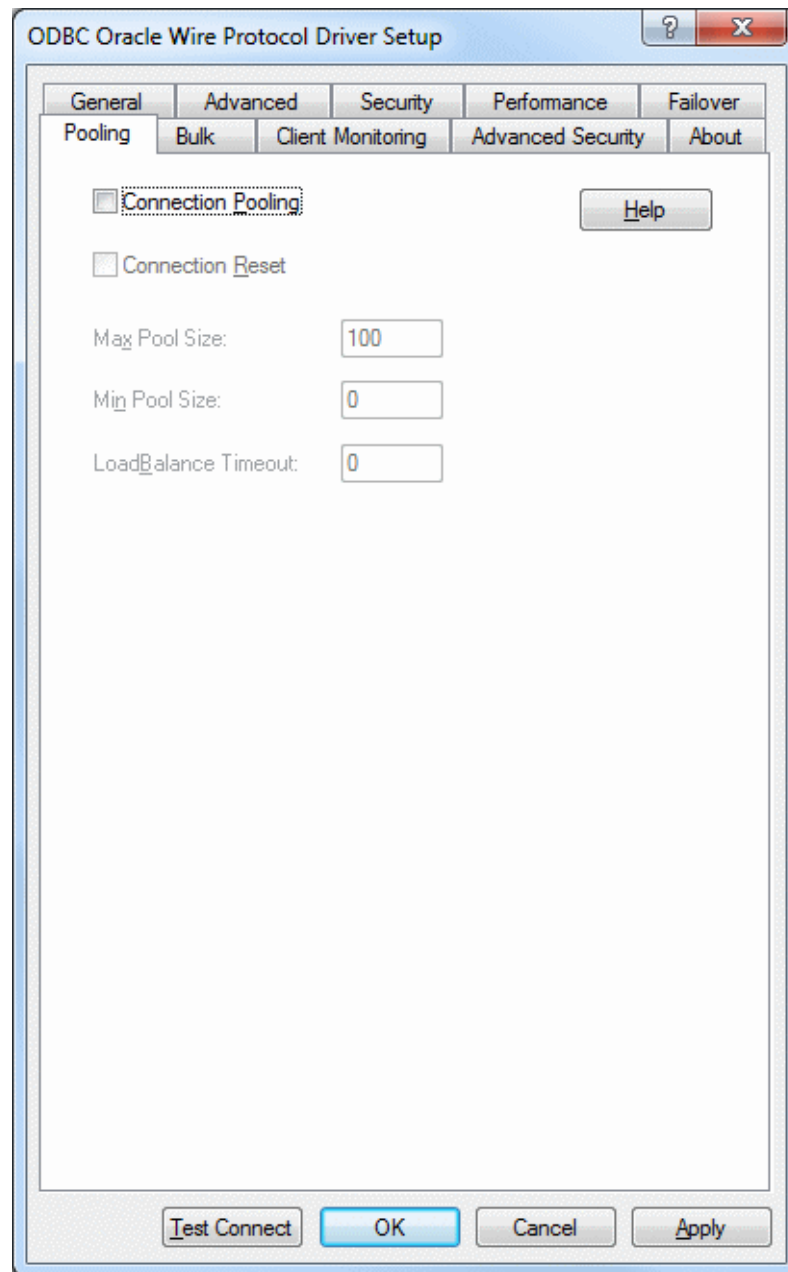


See ["Using Failover" on page 65](#) for a general description of failover and its related connection options.

On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

<b>Connection Options: Failover</b>	<b>Default</b>
<a href="#">Load Balancing (see page 289)</a>	Disabled
<a href="#">Connection Retry Count (see page 274)</a>	0
<a href="#">Connection Retry Delay (see page 275)</a>	3
<a href="#">Alternate Servers (see page 267)</a>	None
<a href="#">Failover Mode (see page 283)</a>	0 - Connection
<a href="#">Failover Granularity (see page 283)</a>	0 - Non-Atomic
<a href="#">Failover Preconnect (see page 284)</a>	Disabled

- 9 Optionally, click the **Pooling** tab to specify connection pooling data source settings.



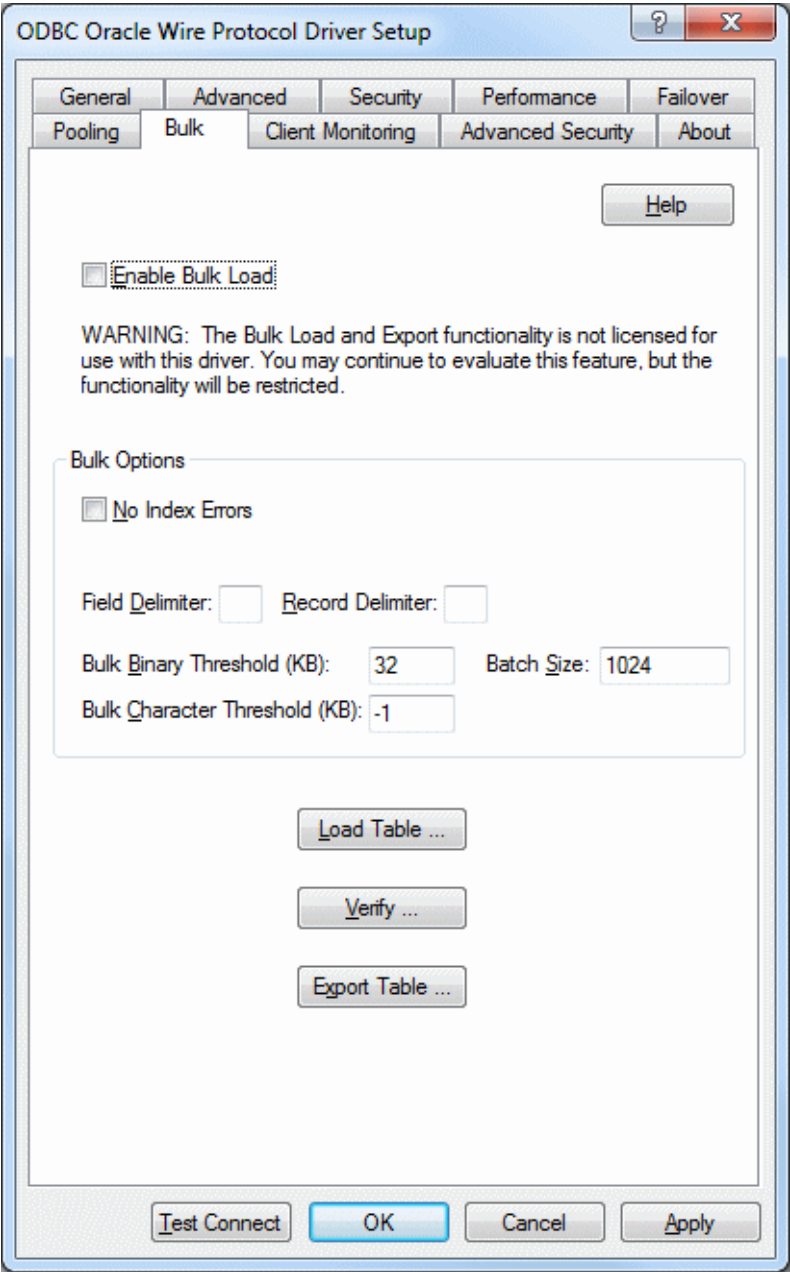
See ["Using DataDirect Connection Pooling" on page 82](#) for a general description of connection pooling.

On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Pooling	Default
<a href="#">Connection Pooling (see page 273)</a>	Disabled
<a href="#">Connection Reset (see page 274)</a>	Disabled
<a href="#">Max Pool Size (see page 290)</a>	100

Connection Options: Pooling	Default
<a href="#">Min Pool Size (see page 291)</a>	0
<a href="#">Load Balance Timeout (see page 288)</a>	0

10 Optionally, click the **Bulk** tab to specify DataDirect Bulk Load data source settings.



See ["Using DataDirect Bulk Load" on page 85](#) for a general description of DataDirect Bulk Load.

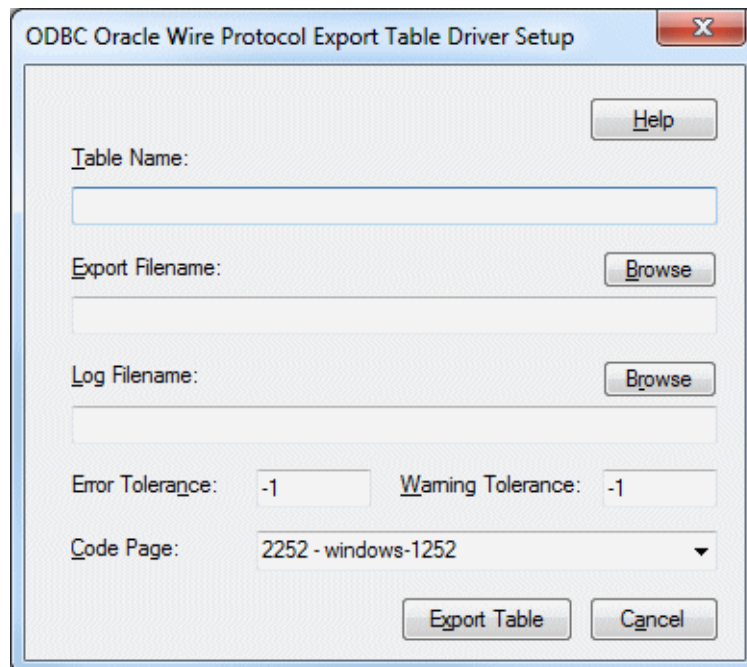
On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Bulk	Default
<a href="#">Enable Bulk Load (see page 278)</a>	Disabled
<a href="#">Bulk Options (see page 270)</a> (No Index Errors is described under the Bulk Options description.)	No Index Errors: disabled
<a href="#">Field Delimiter (see page 284)</a>	None
<a href="#">Record Delimiter (see page 293)</a>	None
<a href="#">Bulk Binary Threshold (see page 269)</a>	32
<a href="#">Bulk Character Threshold (see page 270)</a>	-1
<a href="#">Batch Size (see page 269)</a>	1024

If your application is already coded to use parameter array batch functionality, you can leverage DataDirect Bulk Load features through the Enable Bulk Load connection option. Enabling this option automatically converts the parameter array batch operation to use the database bulk load protocol.

If you are not using parameter array batch functionality, you can export data to a bulk load data file, verify the metadata of the bulk load configuration file against the structure of the target table, and bulk load data to a table. Use the following steps to accomplish these tasks.

- a To export data from a table to a bulk load data file, click **Export Table** from the Bulk tab. The Export Table dialog box appears.



Both a bulk data file and a bulk configuration file are produced by exporting a table. The configuration file has the same name as the data file, but with an XML extension. See ["Using DataDirect Bulk Load" on page 85](#) for details about these files.

The bulk export operation can create a log file and can also export to external files. See ["External Overflow Files" on page 95](#) for more information. The export operation can be configured such that if any errors or warnings occur:

- The operation always completes.
- The operation always terminates.
- The operation terminates after a certain threshold of warnings or errors is exceeded.

**Table Name:** A string that specifies the name of the source database table containing the data to be exported.

**Export Filename:** A string that specifies the path (relative or absolute) and file of the bulk load data file to which the data is to be exported. It also specifies the file name of the bulk configuration file. These files must not already exist; if one of both of them already exists, an error is returned.

**Log Filename:** A string that specifies the path (relative or absolute) and file name of the bulk log file. The log file is created if it does not exist. Events logged to this file are:

- Total number of rows fetched
- A message for each row that failed to export
- Total number of rows that failed to export
- Total number of rows successfully exported

Information about the load is written to this file, preceded by a header. Information about the next load is appended to the end of the file.

If you do not supply a value for Log Filename, no log file is created.

**Error Tolerance:** A value that specifies the number of errors to tolerate before an operation terminates. A value of 0 indicates that no errors are tolerated; the operation fails when the first error is encountered.

The default of -1 means that an infinite number of errors is tolerated.

**Warning Tolerance:** A value that specifies the number of warnings to tolerate before an operation terminates. A value of 0 indicates that no warnings are tolerated; the operation fails when the first warning is encountered.

The default of -1 means that an infinite number of warnings is tolerated.

**Code Page:** A value that specifies the code page value to which the driver must convert all data for storage in the bulk data file. See ["Character Set Conversions" on page 94](#) for more information.

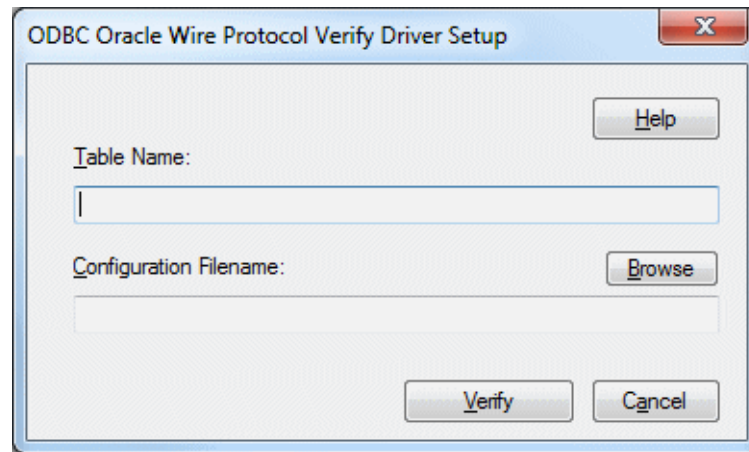
The default value on Windows is the current code page of the machine. On UNIX/Linux, the default value is 4 (ISO 8559-1 Latin-1).

Click **Export Table** to connect to the database and export data to the bulk data file or click **Cancel**.

Click **Export Table** to connect to the database and export data to the bulk data file or click **Cancel**.



- b To verify the metadata of the bulk load configuration file against the structure of the target database table, click **Verify** from the Bulk tab. See ["Verification of the Bulk Load Configuration File" on page 92](#) for details. The ODBC Oracle Wire Protocol Verify Driver Setup dialog box appears.



**Table Name:** A string that specifies the name of the target database table into which the data is to be loaded.

**Configuration Filename:** A string that specifies the path (relative or absolute) and file name of the bulk configuration file.

Click **Verify** to verify table structure or click **Cancel**.

- c To bulk load data from the bulk data file to a database table, click **Load Table** from the Bulk tab. The Load File dialog box appears.

The load operation can create a log file and can also create a discard file that contains rows rejected during the load. The discard file is in the same format as the bulk load data file. After fixing reported issues in the discard file, the bulk load can be reissued using the discard file as the bulk load data file.

The export operation can be configured such that if any errors or warnings occur:

- The operation always completes.
- The operation always terminates.
- The operation terminates after a certain threshold of warnings or errors is exceeded.

If a load fails, the Load Start and Load Count options can be used to control which rows are loaded when a load is restarted after a failure.

**Table Name:** A string that specifies the name of the target database table into which the data is loaded.

**Load Data Filename:** A string that specifies the path (relative or absolute) and file name of the bulk data file from which the data is loaded.

**Configuration Filename:** A string that specifies the path (relative or absolute) and file name of the bulk configuration file.

**Log Filename:** A string that specifies the path (relative or absolute) and file name of the bulk log file. Specifying a value for Log Filename creates the file if it does not already exist. Events logged to this file are:

- Total number of rows read
- Message for each row that failed to load
- Total number of rows that failed to load
- Total number of rows successfully loaded

Information about the load is written to this file, preceded by a header. Information about the next load is appended to the end of the file.

If you do not specify a value for Log Filename, no log file is created.

**Discard Filename:** A string that specifies the path (relative or absolute) and file name of the bulk discard file. Any row that cannot be inserted into database as result of bulk load is added to this file, with the last row rejected added to the end of the file.

Information about the load is written to this file, preceded by a header. Information about the next load is appended to the end of the file.

If you do not specify a value for Discard Filename, a discard file is not created.

**Error Tolerance:** A value that specifies the number of errors to tolerate before an operation terminates. A value of 0 indicates that no errors are tolerated; the operation fails when the first error is encountered.

The default of -1 means that an infinite number of errors is tolerated.

**Load Start:** A value that specifies the first row to be loaded from the data file. Rows are numbered starting with 1. For example, when Load Start is 10, the first 9 rows of the file are skipped and the first row loaded is row 10. This option can be used to restart a load after a failure.

The default value is 1.

**Read Buffer Size (KB):** A value that specifies the size, in KB, of the buffer that is used to read the bulk data file for a bulk load operation.

The default value is 2048.

**Warning Tolerance:** A value that specifies the number of warnings to tolerate before an operation terminates. A value of 0 indicates that no warnings are tolerated; the operation fails when the first warning is encountered.

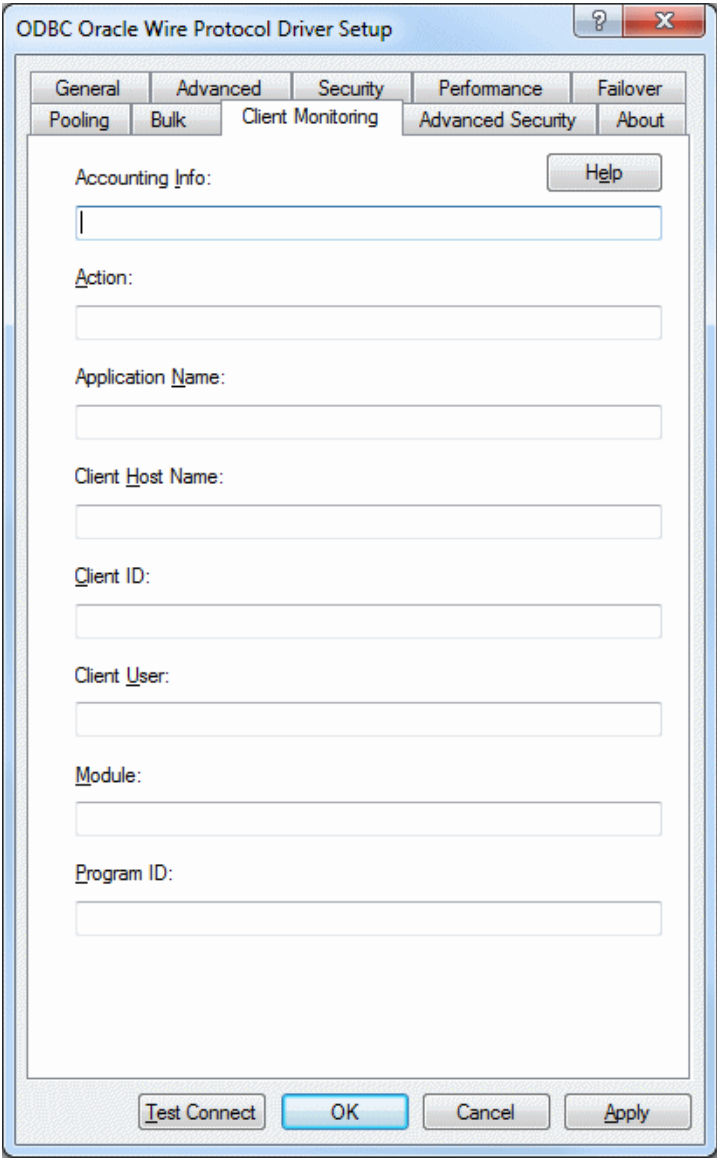
The default of -1 means that an infinite number of warnings is tolerated.

**Load Count:** A value that specifies the number of rows to be loaded from the data file. The bulk load operation loads rows up to the value of Load Count from the file to the database. It is valid for Load Count to specify more rows than exist in the data file. The bulk load operation completes successfully when either the number of rows specified by the Load Count value has been loaded or the end of the data file is reached. This option can be used in conjunction with Load Start to restart a load after a failure.

The default value is the maximum value for SQLLEN. If set to 0, no rows are loaded.

Click **Load Table** to connect to the database and load the table or click **Cancel**.

- 11 Optionally, click the **Client Monitoring** tab to specify additional data source settings.



See ["Using Client Information" on page 75](#) for additional information about client monitoring.

On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Client Monitoring	Default
<a href="#">Accounting Info (see page 266)</a>	None
<a href="#">Action (see page 267)</a>	None
<a href="#">Application Name (see page 267)</a>	None

Connection Options: Client Monitoring	Default
Client Host Name (see page 272)	None
Client ID (see page 273)	None
Client User (see page 273)	None
Module (see page 291)	None
Program ID (see page 293)	None

- 12** At any point during the configuration process, you can click **Test Connect** to attempt to connect to the data source using the connection options specified in the driver Setup dialog box. A logon dialog box appears (see [Using a Logon Dialog Box \(see page 262\)](#) for details). 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 incorrect environment or connection value, it displays an appropriate error message. Click **OK**.

NOTE: If you are configuring alternate servers for use with the connection failover feature, be aware that the Test Connect button tests only the primary server, not the alternate servers.

- 13** 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.

## Using a Connection String

If you want to use a connection string for connecting to a database, or if your application requires it, you must specify either a DSN (data source name), a File DSN, or a DSN-less connection in the string. The difference is whether you use the DSN=, FILEDSN=, or the DRIVER= keyword in the connection string, as described in the ODBC specification. A DSN or FILEDSN connection string tells the driver where to find the default connection information. Optionally, you may specify *attribute=value* pairs in the connection string to override the default values stored in the data source.

The DSN connection string has the form:

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

The FILEDSN connection string has the form:

```
FILEDSN=filename.dsn[;attribute=value[;attribute=value]...]
```

The DSN-less connection string specifies a driver instead of a data source. All connection information must be entered in the connection string because the information is not stored in a data source.

The DSN-less connection string has the form:

```
DRIVER={[driver_name]}[;attribute=value[;attribute=value]...]
```

[Table 8-1](#) lists the long and short names for each attribute, as well as the initial default value when the driver is first installed. You can specify either long or short names in the connection string.

An example of a DSN connection string with overriding attribute values for Oracle Wire Protocol is:

```
DSN=Accounting;ID=JOHN;PWD=XYZZY
```

A FILEDSN connection string is similar except for the initial keyword:

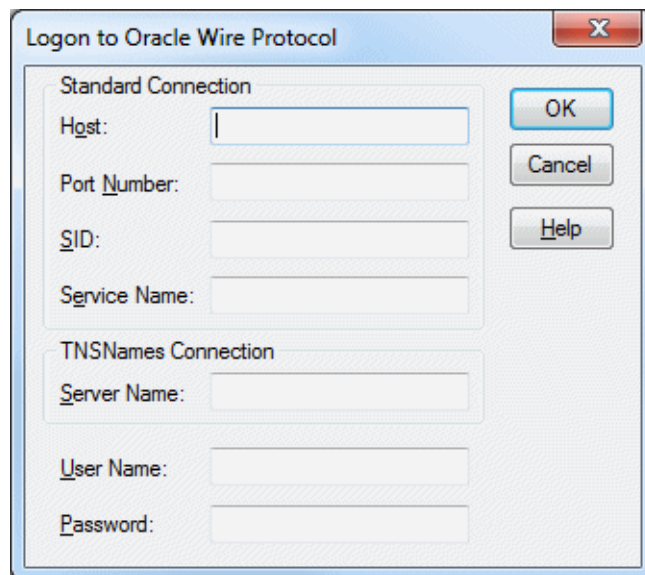
```
FILEDSN=OracleWP.dsn;ID=JOHN;PWD=XYZZY
```

A DSN-less connection string must provide all necessary connection information:

```
DRIVER=DataDirect 7.1 Oracle Wire Protocol;HOST=server1;PORT=1522;  
UID=JOHN;PWD=XYZZY;SERVICENAME=SALES.US.ACME.COM
```

## 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, provide the following information:

**NOTE:** To configure a standard connection, complete the first four fields and skip to Step 6.

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

The IP address can be specified in either IPv4 or IPv6 format, or a combination of the two. See ["Using IP Addresses" on page 54](#) for details concerning these formats.

If you enter a value for this field, the Server Name field is not available.

This field is not available if you enter a value for the Server Name field.

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

If you enter a value for this field, the Server Name field is not available.

This field is not available if you enter a value for the Server Name field.

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

If you enter a value for this field, the Server Name and Service Name fields are not available.

This field is not available if you enter a value for the Service Name or Server Name fields.

- 4 In the Service Name field, type the Oracle service name that specifies the database used for the connection.

See Service Name under [Step 3](#) in "[Configuring and Connecting to Data Sources](#)" on [page 241](#) for details.

If you enter a value for this field, the Server Name and SID fields are not available.

This field is not available if you enter a value for the SID or Server Name field.

NOTE: If you want to configure a TNSNames connection, complete only the following two fields.

- 5 In the Server Name field, type a net service name that exists in the TNSNAMES.ORA file. The corresponding entry in the TNSNAMES.ORA file is used to obtain Host, Port Number, and SID information.

If you enter a value for this field, the Host, Port Number, SID, and Service Name fields are not available.

If you enter a value for either the Host, Port Number, SID, or Service Name fields, this field is not available.

- 6 If required, type your Oracle user name.

- 7 If required, type your Oracle password.

- 8 Click **OK** to log on to the Oracle database installed on the server you specified and to update the values in the Registry.

NOTE: You can also use OS Authentication to connect to an Oracle database. See "[OS Authentication](#)" on [page 310](#) for details.

## Connection Option Descriptions

The following connection option descriptions are listed alphabetically by the GUI name that appears on the driver Setup dialog box. The connection string attribute name, along with its short name, is listed immediately underneath the GUI name. For example:

### Application Using Threads

Attribute    ApplicationUsingThreads (AUT)

In most cases, the GUI name and the attribute name are the same; however, some exceptions exist. If you need to look up an option by its connection string attribute name, please refer to the alphabetical table of connection string attribute names.

Also, a few connection string attributes, for example, Password, do not have equivalent options that appear on the GUI. They are in the list of descriptions alphabetically by their attribute names.

[Table 8-1](#) lists the connection string attributes supported by the Oracle Wire Protocol driver.

**Table 8-1. Oracle Wire Protocol Attribute Names**

Attribute (Short Name)	Default
AccountingInfo (AI)	None
Action (ACT)	None
AlternateServers (ASRV)	None
ApplicationName (AN)	None
ApplicationUsingThreads (AUT)	1 (Enabled)
ArraySize (AS)	60000
AuthenticationMethod (AM)	1 (Encrypt Password)
BulkBinaryThreshold (BBT)	32
BulkCharacterThreshold (BCT)	-1
BulkLoadBatchSize (BLBS))	1024
BulkLoadFieldDelimiter (BLFD)	None
BulkLoadOptions (BLO)	0
BulkLoadRecordDelimiter (BLRD)	None
CachedCursorLimit (CCL)	32
CachedDescriptionLimit (CDL)	0
CatalogIncludesSynonyms (CIS)	1 (Enabled)
CatalogOptions (CO)	0 (Disabled)
ClientHostName (CHN)	None
ClientID (CID)	None
ClientUser (CU)	None
ConnectionReset (CR)	0 (Disabled)
ConnectionRetryCount (CRC)	0



**Table 8-1. Oracle Wire Protocol Attribute Names** (cont.)

Attribute (Short Name)	Default
ConnectionRetryDelay (CRD)	3
DataIntegrityLevel (DIL)	0 (Disabled)
DataIntegrityTypes (DIT)	SHA1,MD5
DataSourceName (DSN)	None
DefaultLongDataBuffLen (DLDBL)	1024
DescribeAtPrepare (DAP)	0 (Disabled)
Description (n/a)	None
EditionName (EN)	None
EnableBulkLoad (EBL)	0 (Disabled)
EnableDescribeParam (EDP)	0 (Disabled)
EnableNcharSupport (ENS)	0 (Disabled)
EnableScrollableCursors (ESC)	1 (Enabled)
EnableServerResultCache (ESRC)	0 (Disabled)
EnableStaticCursorsForLongData (ESCLD)	0 (Disabled)
EnableTimestampwithTimezone (ETWT)	0 (Disabled)
EncryptionLevel (EL)	0 (Disabled)
EncryptionMethod (EM)	0 (No Encryption)
EncryptionTypes (ET)	No encryption methods are specified. The driver sends a list of all of the encryption methods to the Oracle server.
FailoverGranularity (FG)	0 (Non-Atomic)
FailoverMode (FM)	0 (Connection)
FailoverPreconnect (FP)	0 (Disabled)
FetchTSWTZasTimestamp (FTSWTZAT)	0 (Disabled)
GSSClient (GSSC)	native
HostName (HOST)	None
HostNameInCertificate (HNIC)	None
IANAAppCodePage (IACP)	4 (ISO 8559-1 Latin-1)
UNIX ONLY	
InitializationString (IS)	None
KeyPassword (KP)	None
Keystore (KS)	None
KeystorePassword (KSP)	None
LoadBalanceTimeout (LBT)	0
LoadBalancing (LB)	0 (Disabled)
LocalTimezoneOffset (LTZO)	"" (Empty String)
LockTimeout (LTO)	-1
LoginTimeout (LT)	15

**Table 8-1. Oracle Wire Protocol Attribute Names** (cont.)

Attribute (Short Name)	Default
LogonID (UID)	None
MaxPoolSize (MXPS)	100
MinPoolSize (MNPS)	0
Module (MOD)	None
Password (PWD)	None
Pooling (POOL)	0 (Disabled)
PortNumber (PORT)	None
ProcedureRetResults (PRR)	0 (Disabled)
ProgramID (PID)	None
QueryTimeout (QT)	0
ReportCodepageConversionErrors (RCCE)	0 (Ignore Errors)
ReportRecycleBin (RRB)	0 (Disabled)
ServerName (SRVR)	None
ServerType (ST)	0 (Server Default)
ServiceName (SN)	None
SID (SID)	None
TimestampEscapeMapping (TEM)	0 (Oracle Version Specific)
TNSNamesFile (TNF)	None
Truststore (TS)	None
TruststorePassword (TSP)	None
UseCurrentSchema (UCS)	1 (Enabled)
ValidateServerCertificate (VSC)	1 (Enabled)
WireProtocolMode (WPM)	2

## Accounting Info

Attribute	AccountingInfo (AI)
Description	Accounting information to be stored in the database. This value sets the CLIENT_INFO value of the V\$SESSION table on the server. This value is used by the client information feature.  This connection option can affect performance. See <a href="#">"Performance Considerations" on page 300</a> for details.
Valid Values	<i>string</i>  where <i>string</i> is the accounting information.
Default	None
GUI Tab	<a href="#">Client Monitoring tab</a> on page 260

## Action

Attribute	Action (ACT)
Description	<p>The current action (Select, Insert, Update, or Delete, for example) within the current module. This value sets the ACTION column of the V\$SESSION table on the server. This value is used by the client information feature.</p> <p>This option only applies to connections to Oracle 10g R2 and higher database servers.</p> <p>NOTE: You can also specify this information using the Oracle DBMS_APPLICATION_INFO.SET_ACTION procedure or the DBMS_APPLICATION_INFO.SET_MODULE procedure.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 300</a> for details.</p>
Valid Values	<p><i>string</i></p> <p>where <i>string</i> is the current action.</p>
Default	None
GUI Tab	<a href="#">Client Monitoring tab</a> on page 260

## Alternate Servers

Attribute	AlternateServers (ASRV)
Description	<p>A list of alternate database servers to which the driver tries to connect if the primary database server is unavailable. Specifying a value for this option enables connection failover for the driver. The value you specify must be in the form of a string that defines the physical location of each alternate server. All of the other required connection information for each alternate server is the same as what is defined for the primary server connection.</p>
Valid Values	<p>(HostName=hostvalue:PortNumber=portvalue:{SID=sidvalue   ServiceName=servicevalue}[ , . . .])</p> <p>You must specify the host name, port number, and either the SID or service name of each alternate server.</p>
Example	<p>The following Alternate Servers value defines two alternate database servers for connection failover:</p> <pre>(HostName=AccountingOracleServer:PortNumber=1521: SID=Accounting,HostName=255.201.11.24:PortNumber=1522: ServiceName=ABackup.NA.MyCompany)</pre>
Default	None
GUI tab	<a href="#">Failover tab</a> on page 251

## Application Name

Attribute	ApplicationName (AN)
Description	<p>The name of the application to be stored in the database. This value sets the dbms_session value in the database and the PROGRAM value of the V\$SESSION table on the server. This value is used by the client information feature.</p>

This connection option can affect performance. See ["Performance Considerations" on page 300](#) for details.

Valid Values *string*

where *string* is the name of the application.

If a value for this option is not specified, the driver uses the name of the current process.

Default None

GUI Tab [Client Monitoring tab](#) on page 260

## Application Using Threads

Attribute ApplicationUsingThreads (AUT)

Description Determines whether the driver works with applications using multiple ODBC threads.

This connection option can affect performance. See ["Performance Considerations" on page 300](#) for details.

Valid Values 0 | 1

If set to 1 (Enabled), the driver works with single-threaded and multi-threaded applications.

If set to 0 (Disabled), the driver does not work with multi-threaded applications. If using the driver with single-threaded applications, this value avoids additional processing required for ODBC thread-safety standards.

Default 1 (Enabled)

GUI tab [Advanced tab](#) on page 245

## Array Size

Attribute ArraySize (AS)

Description The number of bytes the driver can fetch in a single network round trip. 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.

This connection option can affect performance. See ["Performance Considerations" on page 300](#) for details.

Valid Values An integer from 1 to 4,294,967,296 (4 GB)

The value 1 does not define the number of bytes but, instead, causes the driver to allocate space for exactly one row of data.

Default 60000

GUI Tab [Performance tab](#) on page 250

## Authentication Method

Attribute	AuthenticationMethod (AM)
Description	Specifies the method the driver uses to authenticate the user to the server when a connection is established. If the specified authentication method is not supported by the database server, the connection fails and the driver generates an error.
Valid Values	1   3   4   5  If set to 1 (Encrypt Password), the driver sends the user ID in clear text and an encrypted password to the server for authentication.  If set to 3 (Client Authentication), the driver uses client authentication when establishing a connection. The database server relies on the client to authenticate the user and does not provide additional authentication.  If set to 4 (Kerberos Authentication), the driver uses Kerberos authentication. This method supports both Windows Active Directory Kerberos and MIT Kerberos environments.  When set to 5 (Kerberos with UID & PWD), the driver uses both Kerberos authentication and user ID and password authentication. The driver first authenticates the user using Kerberos. If a user ID and password are specified, the driver reauthenticates using the user name and password supplied. An error is generated if a user ID and password are not specified.
Default	1 (Encrypt Password)
GUI tab	<a href="#">Security tab</a> on page 247

## Batch Size

Attribute	BulkLoadBatchSize (BLBS)
Description	The number of rows that the driver sends to the database at a time during bulk operations. This value applies to all methods of bulk loading.
Valid Values	0   $x$  where $x$ is the number of rows to send during a bulk operation.
Default	1024
GUI Tab	<a href="#">Bulk tab</a> on page 254

## Bulk Binary Threshold

Attribute	BulkBinaryThreshold (BBT)
Description	The maximum size, in KB, of binary data that is exported to the bulk data file.
Valid Values	-1   0   $x$  where $x$ is an integer that specifies the number of KB.  If set to -1, all binary data, regardless of size, is written to the bulk data file, not to an external file.

If set to 0, all binary data, regardless of size, is written to an external file, not the bulk data file. A reference to the external file is written to the bulk data file.

If set to *x*, any binary data exceeding this specified number of KB is written to an external file, not the bulk data file. A reference to the external file is written to the bulk data file.

Default 32

GUI Tab [Bulk tab](#) on page 254

**Bulk Character Threshold**

Attribute BulkCharacterThreshold (BCT)

Description The maximum size, in KB, of character data that is exported to the bulk data file.

Valid Values -1 | 0 | *x*

where *x* is an integer that specifies the number of KB.

If set to -1, all character data, regardless of size, is written to the bulk data file, not to an external file.

If set to 0, all character data regardless of size, is written to an external file, not the bulk data file. A reference to the external file is written to the bulk data file.

If set to *x*, any character data exceeding this specified number of KB is written to an external file, not the bulk data file. A reference to the external file is written to the bulk data file.

Default -1

GUI Tab [Bulk tab](#) on page 254

**Bulk Options**

Attribute BulkLoadOptions (BLO)

Description Toggles options for the bulk load process.

This option only applies to connections to Oracle 11g R2 and higher database servers.

Valid Values 0 | *x*

where *x* is a positive integer representing the cumulative total of the Bulk Options values.

If set to 0, none of the options for bulk load are enabled.

If set to *x*, the values represented by *x* are enabled.

NOTE: The cumulative value of the options is only used in a connection string with the connection string attribute, BulkLoadOptions. On the Bulk tab of the driver Setup dialog, the individual options are enabled by selecting the appropriate check box.

Currently, the only bulk load option available is:

No Index Errors - The driver stops a bulk load operation when a value that would cause an index to be invalidated is loaded. For example, if a value is loaded that violates a unique or non-null constraint, the driver stops the bulk load operation and discards all data being loaded, including any data that was loaded prior to the problem value. If not enabled, the

bulk load operation continues even if a value that would cause an index to be invalidated is loaded. Value=128.

Default 0  
GUI Tab [Bulk tab](#) on page 254

## Cached Cursor Limit

Attribute CachedCursorLimit (CCL)

Description Specifies the number of Oracle Cursor Identifiers that the driver stores in cache. A Cursor Identifier is needed for each concurrent open Select statement. When a Select statement is closed, the driver stores the identifier in its cache, up to the limit specified, rather than closing the Cursor Identifier. When a new Cursor Identifier is needed, the driver takes one from its cache, if one is available. Cached Cursor Identifiers are closed when the connection is closed.

This connection option can affect performance. See ["Performance Considerations" on page 300](#) for details.

Valid Values An integer from 0 to 65535

Default 32

GUI Tab [Performance tab](#) on page 250

## Cached Description Limit

Attribute CachedDescriptionLimit (CDL)

Description Specifies the number of descriptions that the driver saves for Select statements. These descriptions include the number of columns, data type, length, and scale for each column. The matching is done by an exact-text match through the FROM clause.

NOTE: If the Select statement contains a Union or a nested Select, the description is not cached.

This connection option can affect performance. See ["Performance Considerations" on page 300](#) for details.

Valid Values An integer from 0 to 65535

Default 0

GUI Tab [Performance tab](#) on page 250

## Catalog Functions Include Synonyms

Attribute CatalogIncludesSynonyms (CIS)

Description Determines whether synonyms are included in calls to SQLProcedures, SQLStatistics, and SQLProcedureColumns.

This connection option can affect performance. See ["Performance Considerations" on page 300](#) for details.

Valid Values 0 | 1

If set to 1 (Enabled), synonyms are included in calls to `SQLProcedures`, `SQLStatistics`, and `SQLProcedureColumns`.

If set to 0 (Disabled), synonyms are excluded (a non-standard behavior) and performance is thereby improved.

Default 1 (Enabled)

GUI Tab [Performance tab](#) on page 250

## Catalog Options

Attribute `CatalogOptions (CO)`

Description Determines whether `SQL_NULL_DATA` is returned for the result columns `REMARKS` and `COLUMN_DEF`.

This connection option can affect performance. See ["Performance Considerations" on page 300](#) for details.

Valid Values 0 | 1

If set to 1 (Enabled), the result column `REMARKS` (for the catalog functions `SQLTables` and `SQLColumns`) and the result column `COLUMN_DEF` (for the catalog function `SQLColumns`) return actual values. Enabling this option reduces the performance of your catalog (`SQLColumns` and `SQLTables`) queries.

If set to 0 (Disabled), `SQL_NULL_DATA` is returned for the result columns `REMARKS` and `COLUMN_DEF`.

Default 0 (Disabled)

GUI Tab [Advanced tab](#) on page 245

## Client Host Name

Attribute `ClientHostName (CHN)`

Description The host name of the client machine to be stored in the database. This value sets the `MACHINE` value in the `V$SESSION` table on the server. This value is used by the client information feature.

This connection option can affect performance. See ["Performance Considerations" on page 300](#) for details.

Valid Values *string*

where *string* is the host name of the client machine.

If a value for this option is not specified, the driver uses the current machine name and IP address in the following format:

*machine\_name/IP\_address*

Default None

GUI Tab [Client Monitoring tab](#) on page 260



## Client ID

Attribute	ClientID (CID)
Description	<p>Additional information about the client to be stored in the database. This value sets the CLIENT_IDENTIFIER value in the V\$SESSION table on the server. This value is used by the client information feature.</p> <p>This option only applies to connections to Oracle 10g R2 and higher database servers.</p> <p>NOTE: You can also specify this information using the Oracle DBMS_SESSION.SETIDENTIFIER procedure or the DBMS_APPLICATION_INFO.SET_CLIENT_INFO procedure.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 300</a> for details.</p>
Valid Values	<p><i>string</i></p> <p>where <i>string</i> is additional information about the client.</p>
Default	None
GUI Tab	<a href="#">Client Monitoring tab</a> on page 260

## Client User

Attribute	ClientUser (CU)
Description	<p>The user ID to be stored in the database. This value sets the OSUSER value in the V\$SESSION table on the server. This value is used by the client information feature.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 300</a> for details.</p>
Valid Values	<p>-1   <i>string</i></p> <p>where <i>string</i> is a valid user ID.</p> <p>The value -1 means that the driver uses the userid of the user that is currently logged onto the client.</p> <p>If a value for this option is not specified, the driver uses name of the user that is currently logged into the OS.</p>
Default	None
GUI Tab	<a href="#">Client Monitoring tab</a> on page 260

## Connection Pooling

Attribute	Pooling (POOL)
Description	<p>Specifies whether to use the driver's connection pooling.</p> <p>NOTE: The application must be thread-enabled to use connection pooling.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 300</a> for details.</p>

Valid Values 0 | 1

If set to 1 (Enabled), the driver uses connection pooling.

If set to 0 (Disabled), the driver does not use connection pooling.

Default 0 (Disabled)

GUI Tab [Pooling tab](#) on page 253

## Connection Reset

Attribute ConnectionReset (CR)

Description Determines whether the state of connections that are removed from the connection pool for reuse by the application is reset to the initial configuration of the connection.

This connection option can affect performance. See ["Performance Considerations" on page 300](#) for details.

Valid Values 0 | 1

If set to 1 (Enabled), the state of connections removed from the connection pool for reuse by an application is reset to the initial configuration of the connection. Resetting the state can negatively impact performance because additional commands must be sent over the network to the server to reset the state of the connection.

If set to 0 (Disabled), the state of connections is not reset.

Default 0 (Disabled)

GUI Tab [Pooling tab](#) on page 253

## Connection Retry Count

Attribute ConnectionRetryCount (CRC)

Description The number of times the driver retries connection attempts to the primary database server, and if specified, alternate servers until a successful connection is established.

This option and the Connection Retry Delay connection option, which specifies the wait interval between attempts, can be used in conjunction with failover.

Valid Values 0 |  $x$

where  $x$  is a positive integer from 1 to 65535.

If set to 0, the driver does not try to connect after the initial unsuccessful attempt.

If set to  $x$ , the driver retries connection attempts the specified number of times. If a connection is not established during the retry attempts, the driver returns an error that is generated by the last server to which it tried to connect.

Default 0

GUI Tab [Failover tab](#) on page 251

## Connection Retry Delay

Attribute	ConnectionRetryDelay (CRD)
Description	<p>The number of seconds the driver waits between connection retry attempts when Connection Retry Count is set to a positive integer.</p> <p>This option and the Connection Retry Count connection option can be used in conjunction with failover.</p>
Valid Values	<p>0   <math>x</math></p> <p>where <math>x</math> is a positive integer from 1 to 65535.</p> <p>If set to 0, there is no delay between retries.</p> <p>If set to <math>x</math>, the driver waits the specified number of seconds between connection retry attempts.</p>
Default	3
GUI Tab	<a href="#">Failover tab</a> on page 251

## Data Integrity Level

Attribute	DataIntegrityLevel (DIL)
Description	<p>Specifies a preference for the data integrity to be used on data sent between the driver and the database server. The connection fails if the database server does not have a compatible integrity algorithm. See <a href="#">"Encryption and Data Integrity" on page 308</a> for more information.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 300</a> for details.</p>
Valid Values	<p>Rejected   Accepted   Requested   Required</p> <p>If set to 0 - Rejected, a data integrity check on data sent between the driver and the database server is refused. The connection fails if the database server specifies REQUIRED.</p> <p>If set to 1 - Accepted, a data integrity check can be made on data sent between the driver and the database server. Data integrity is used if the database server requests or requires it.</p> <p>If set to 2 - Requested, the driver enables a data integrity check on data sent between the driver and the database server if the database server permits it.</p> <p>If set to 3 - Required, a data integrity check must be performed on data sent between the driver and the database server. The connection fails if the database server specifies REJECTED.</p> <p>NOTE: Consult your database administrator concerning the data integrity settings of your Oracle server.</p>
Default	0 - Rejected
GUI Tab	<a href="#">Advanced Security tab</a> on page 248

## Data Integrity Types

Attribute	DataIntegrityTypes (DIT)
Description	Determines the method the driver uses to protect against attacks that intercept and modify data being transmitted between the client and server. You can enable data integrity protection without enabling encryption. See <a href="#">"Encryption and Data Integrity" on page 308</a> for more information.
	NOTE: This option has no effect if <a href="#">Data Integrity Level</a> is set to 0 - Rejected.
	This connection option can affect performance. See <a href="#">"Performance Considerations" on page 300</a> for details.
Valid Values	MD5   SHA1   SHA1,MD5
	If set to MD5, the driver uses the Message Digest 5 (MD5) algorithm.
	If set to SHA1, the driver uses the Secure Hash Algorithm (SHA-1).
	If multiple values are specified and Oracle Advanced Security data integrity is enabled using the Data Integrity Level option, the database server determines which algorithm is used based on how it is configured..
	NOTE: Consult your database administrator concerning the data integrity settings of your Oracle server.
Default	SHA1,MD5
GUI Tab	<a href="#">Advanced Security tab</a> on page 248

## Data Source Name

Attribute	DataSourceName (DSN)
Description	The name of a data source in your Windows Registry or odbc.ini file.
Valid Values	<i>string</i>
	where <i>string</i> is the name of a data source.
Default	None
GUI Tab	<a href="#">General tab</a> on page 244

## Default Buffer Size for Long/LOB Columns (in Kb)

Attribute	DefaultLongDataBuffLen (DLDBL)
Description	The maximum length of data (in KB) the driver can fetch from long columns in a single round trip and the maximum length of data that the driver can send using the SQL_DATA_AT_EXEC parameter.
	This connection option can affect performance. See <a href="#">"Performance Considerations" on page 300</a> for details.

Valid Values An integer in multiples of 1024

The value must be in multiples of 1024 (for example, 1024, 2048). You need to increase the default value if the total size of any Long data exceeds 1 MB. This value is multiplied by 1024 to determine the total maximum length of fetched data. For example, if you enter a value of 2048, the maximum length of data would be 1024 x 2048, or 2097152 (2 MB).

Default 1024

GUI Tab [Advanced tab](#) on page 245

## Describe at Prepare

Attribute DescribeAtPrepare (DAP)

Description Determines whether the driver describes the SQL statement at prepare time.

This connection option can affect performance. See ["Performance Considerations" on page 300](#) for details.

Valid Values 0 | 1

If set to 1 (Enabled), the driver describes the SQL statement at prepare time.

If set to 0 (Disabled), the driver does not describe the SQL statement at prepare time.

Default 0 (Disabled)

GUI Tab [Advanced tab](#) on page 245

## Description

Attribute Description (n/a)

Description An optional long description of a data source. This description is not used as a runtime connection attribute, but does appear in the ODBC.INI section of the Registry and in the `odbc.ini` file.

Valid Values *string*

where *string* is a description of a data source.

Default None

GUI Tab [General tab](#) on page 244

## Edition Name

Attribute EditionName (EN)

Description The name of the Oracle edition the driver uses when establishing a connection. Oracle 11g R2 and higher allows your database administrator to create multiple editions of schema objects so that your application can still use those objects while the database is being upgraded. This option is only valid for Oracle 11g R2 and higher databases and tells the driver which edition of the schema objects to use.

The driver uses the default edition in the following cases:

- When the specified edition is not a valid edition. The driver generates a warning indicating that it was unable to set the current edition to the specified edition.
- When the value for this option is not specified or is set to an empty string.

If failover is enabled using the Failover Mode connection option and a connection fails over to another database server, the driver connects to the alternate server using the same edition that was used for the failed connection. The driver does not track changes to the current edition made using the ALTER SESSION SQL statement.

Valid Values *string*

where *string* is the name of a valid Oracle edition.

Default None

GUI Tab [General tab](#) on page 244

## Enable Bulk Load

Attribute EnableBulkLoad (EBL)

Description Specifies the bulk load method.

Valid Values 0 | 1

If set to 1 (Enabled), the driver uses the database bulk load protocol when an application executes an INSERT with multiple rows of parameter data. If the protocol cannot be used, the driver returns a warning.

If set to 0 (Disabled), the driver uses standard parameter arrays.

See ["DataDirect Bulk Load" on page 309](#) for further details.

Default 0 (Disabled)

GUI Tab [Bulk tab](#) on page 254

## Enable N-CHAR Support

Attribute EnableNcharSupport (ENS)

Description Determines whether the driver provides support for the N-types NCHAR, NVARCHAR2, and NCLOB. These types are described as SQL\_WCHAR, SQL\_WVARCHAR, and SQL\_WLONGVARCHAR, and are returned as supported by SQLGetTypeInfo. In addition, the "normal" char types (char, varchar2, long, clob) are described as SQL\_CHAR, SQL\_VARCHAR, and SQL\_LONGVARCHAR regardless of the character set on the Oracle server.

See ["Unicode Support" on page 306](#) for details.

NOTE: Valid only on Oracle 9i and higher.

Valid Values 0 | 1

If set to 1 (Enabled), the driver provides support for the N-types NCHAR, NVARCHAR2, and NCLOB.

If set to 0 (Disabled), the driver does not provide support for the N-types NCHAR, NVARCHAR2, and NCLOB.

Default 0 (Disabled)

GUI Tab [Advanced tab](#) on page 245

## Enable Scrollable Cursors

Attribute EnableScrollableCursors (ESC)

Description Determines whether scrollable cursors, both Keyset and Static, are enabled for the data source.

This connection option can affect performance. See ["Performance Considerations" on page 300](#) for details.

Valid Values 0 | 1

If set to 1 (Enabled), scrollable cursors are enabled for the data source.

If set to 0 (Disabled), scrollable cursors are not enabled.

Default 1 (Enabled)

GUI Tab [Performance tab](#) on page 250

## Enable Server Result Cache

Attribute EnableServerResultCache (ESRC)

Description Determines whether the driver sets the RESULT\_CACHE\_MODE session parameter to FORCE.

This option only applies to connections to Oracle 11g database servers that support server-side result set caching.

This connection option can affect performance. See ["Performance Considerations" on page 300](#) for details.

Valid Values 0 | 1

If set to 1 (Enabled), the driver sets the RESULT\_CACHE\_MODE session parameter to FORCE.

If set to 0 (Disabled), the driver does not sets the RESULT\_CACHE\_MODE session parameter.

Default 0 (Disabled)

GUI Tab [Advanced tab](#) on page 245

## Enable SQLDescribeParam

Attribute EnableDescribeParam (EDP)

Description Determines whether the SQLDescribeParam function describes all parameters with a data type of SQL\_VARCHAR for Select statements. For Insert/Update/Delete statements and for

stored procedures, the parameters are described as the actual Oracle data types on the Oracle server. This option must be enabled to access data when using Microsoft Remote Data Objects (RDO).

Valid Values 0 | 1

If set to 1 (Enabled), the SQLDescribeParam function describes all parameters with a data type of SQL\_VARCHAR for Select statements.

If set to 0 (Disabled), the SQLDescribeParam function does not describe all parameters with a data type of SQL\_VARCHAR for Select statements.

Default 0 (Disabled)

GUI Tab [Advanced tab](#) on page 245

## Enable Static Cursors for Long Data

Attribute EnableStaticCursorsForLongData (ESCLD)

Description Determines whether the driver supports Long columns when using a static cursor. Enabling this option causes a performance penalty at the time of execution when reading Long data.

This connection option can affect performance. See ["Performance Considerations" on page 300](#) for details.

Valid Values 0 | 1

If set to 1 (Enabled), the driver supports Long columns when using a static cursor.

If set to 0 (Disabled), the driver does not support Long columns when using a static cursor.

NOTE: You must enable this option if you want to persist a result set that contains Long data into an XML data file.

Default 0 (Disabled)

GUI Tab [Performance tab](#) on page 248

## Enable Timestamp with Timezone

Attribute EnableTimestampwithTimezone (ETWT)

Description Determines whether the driver exposes timestamps with timezones to the application.

Valid Values 0 | 1

If set to 1 (Enabled), the driver exposes timestamps with timezones to the application. The driver issues an ALTER SESSION at connection time to modify NLS\_TIMESTAMP\_TZ\_FORMAT. NLS\_TIMESTAMP\_TZ\_FORMAT is changed to the ODBC definition of a timestamp literal with the addition of the timezone literal: 'YYYY-MM-DD HH24:MI:SSXFF TZR'.

If set to 0 (Disabled), timestamps with timezones are not exposed to the application.

Default 0 (Disabled)

GUI Tab [Advanced tab](#) on page 245



## Encryption Level

Attribute	EncryptionLevel (EL)
Description	<p>Specifies a preference on whether to use encryption on data being sent between the driver and the database server.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 300</a> for details.</p>
Valid Values	<p>Rejected   Accepted   Requested   Required</p> <p>If set to 0 - Rejected, or if no match is found between the driver and server encryption types, data sent between the driver and the database server is not encrypted or decrypted. The connection fails if the database server specifies REQUIRED.</p> <p>If set to 1 - Accepted, encryption is used on data sent between the driver and the database server if the database server requests or requires it.</p> <p>If set to 2 - Requested, data sent between the driver and the database server is encrypted and decrypted if the database server permits it.</p> <p>If set to 3 - Required, data sent between the driver and the database server must be encrypted and decrypted. The connection fails if the database server specifies REJECTED.</p> <p>NOTE: Consult your database administrator concerning the data encryption settings of your Oracle server.</p>
Default	0 - Rejected
GUI Tab	<a href="#">Advanced Security tab</a> on page 248

## Encryption Method

Attribute	EncryptionMethod (EM)
Description	<p>The method the driver uses to encrypt data sent between the driver and the database server. If the specified encryption method is not supported by the database server, the connection fails and the driver returns an error.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 300</a> for details.</p>
Valid Values	<p>0   1   3   4   5</p> <p>If set to 0 (No Encryption), data is not encrypted.</p> <p>If set to 1 (SSL), data is encrypted using SSL. If the server supports protocol negotiation, the driver and server negotiate the use of TLS v1, SSL v3, or SSL v2 in that order.</p> <p>If set to 3 (SSL3), the driver uses SSL3 data encryption.</p> <p>If set to 4 (SSL2), the driver uses SSL2 data encryption.</p> <p>If set to 5 (TLS1), the driver uses TLS1 data encryption.</p> <p>NOTE: Consult your database administrator concerning the SSL settings of your Oracle server.</p>

Default 0 (No Encryption)  
GUI Tab [Security tab](#) on page 247

Encryption Types

Attribute EncryptionTypes (ET)  
Description Specifies a comma-separated list of the encryption algorithms to use if Oracle Advanced Security encryption is enabled using the Encryption Level connection property.

NOTE: This option is ignored if [Encryption Level](#) is set to 0 - Rejected.

This connection option can affect performance. See ["Performance Considerations" on page 300](#) for details.

Valid Values `encryption_algorithm[,encryption_algorithm]...`

where `encryption_algorithm` is a encryption algorithm specifying an algorithm in the following table:

Encryption Algorithm	Description
3DES112	Two-key Triple-DES (with an effective key size of 112-bit).
AES128	AES with a 128-bit key size.
AES192	AES with a 192-bit key size.
AES256	AES with a 256-bit key size.
DES	DES (with an effective key size of 56-bit).
DES168	Three-key Triple-DES (with an effective key size of 168-bit).
RC4_128	RC4-128 with a 128-bit key size.
RC4_256	RC4 with a 256-bit key size.
RC4_40	RSA RC4 with a 40-bit key size.
RC4_56	RSA RC4 with a 56-bit key size.

NOTE: Consult your database administrator concerning the data encryption settings of your Oracle server.

Example Your security environments specifies that you can use RC4 with a 256-bit key size, AES with a 192-bit key size, or two-key Triple-DES with an effective key size of 112-bit. Use the following values:

`EncryptionTypes=RC4_256,AES192,3DES112`

Default On the GUI tab: all check boxes are selected.  
  
In the connection string: no encryption methods are specified. The driver sends a list of all of the encryption methods to the Oracle server.

GUI Tab [Advanced Security tab](#) on page 248

## Failover Granularity

Attribute	FailoverGranularity (FG)
Description	<p>Determines whether the driver fails the entire failover process or continues with the process if errors occur while trying to reestablish a lost connection.</p> <p>This option applies only when Failover Mode is set to 1 (Extended Connection) or 2 (Select).</p> <p>The Alternate Servers option specifies one or multiple alternate servers for failover and is required for all failover methods.</p>
Valid Values	<p>0   1   2   3</p> <p>If set to 0 (Non-Atomic), the driver continues with the failover process and posts any errors on the statement on which they occur.</p> <p>If set to 1 (Atomic) the driver fails the entire failover process if an error is generated as the result of anything other than executing and repositioning a Select statement. If an error is generated as a result of repositioning a result set to the last row position, the driver continues with the failover process, but generates a warning that the Select statement must be reissued.</p> <p>If set to 2 (Atomic Including Repositioning), the driver fails the entire failover process if any error is generated as the result of restoring the state of the connection or the state of work in progress.</p> <p>If set to 3 (Disable Integrity Check), the driver does not verify that the rows that were restored during the failover process match the original rows. This value applies only when Failover Mode is set to 2 (Select).</p>
Default	0 (Non-Atomic)
GUI Tab	<a href="#">Failover tab</a> on page 251

## Failover Mode

Attribute	FailoverMode (FM)
Description	<p>Specifies the type of failover method the driver uses.</p> <p>The Alternate Servers option specifies one or multiple alternate servers for failover and is required for all failover methods.</p>
Valid Values	<p>0   1   2</p> <p>If set to 0 (Connection), the driver provides failover protection for new connections only.</p> <p>If set to 1 (Extended Connection), the driver provides failover protection for new and lost connections, but not any work in progress.</p> <p>If set to 2 (Select), the driver provides failover protection for new and lost connections. In addition, it preserves the state of work performed by the last Select statement executed.</p>
Default	0 (Connection)
GUI Tab	<a href="#">Failover tab</a> on page 251

## Failover Preconnect

Attribute	FailoverPreconnect (FP)
Description	<p>Specifies whether the driver tries to connect to the primary and an alternate server at the same time.</p> <p>This attribute applies only when Failover Mode is set to 1 (Extended Connection) or 2 (Select) and at least one alternate server is specified.</p> <p>The Alternate Servers option specifies one or multiple alternate servers for failover and is required for all failover methods.</p>
Valid Values	<p>0   1</p> <p>If set to 0 (Disabled), the driver tries to connect to an alternate server only when failover is caused by an unsuccessful connection attempt or a lost connection. This value provides the best performance, but your application typically experiences a short wait while the failover connection is attempted.</p> <p>If set to 1 (Enabled), the driver tries to connect to the primary and an alternate server at the same time. This can be useful if your application is time-sensitive and cannot absorb the wait for the failover connection to succeed.</p>
Default	0 (Disabled)
GUI Tab	<a href="#">Failover tab</a> on page 251

## Fetch TSWTZ as Timestamp

Attribute	FetchTSWTZasTimestamp (FTSWTZAT)
Description	<p>Determines whether the driver returns column values with the timestamp with time zone data type as the ODBC data type SQL_TYPE_TIMESTAMP or SQL_VARCHAR.</p> <p>Valid on Oracle 10g R2 or higher.</p>
Valid Values	<p>0   1</p> <p>If set to 1 (Enabled), the driver returns column values with the timestamp with time zone data type as the ODBC type SQL_TYPE_TIMESTAMP. The time zone information in the fetched value is truncated. Use this value if your application needs to process values the same way as TIMESTAMP columns.</p> <p>If set to 0 (Disabled), the driver returns column values with the timestamp with time zone data type as the ODBC data type SQL_VARCHAR. Use this value if your application requires the time zone information in the fetched value.</p>
Default	0 (Disabled)
GUI Tab	<a href="#">Advanced tab</a> on page 245

## Field Delimiter

Attribute	BulkLoadFieldDelimiter (BLFD)
Description	Specifies the character that the driver will use to delimit the field entries in a bulk load data file.

Valid Values *x*

where *x* is any printable character.

For simplicity, avoid using a value that can be in the data, including all alphanumeric characters, the dash(-), the colon(:), the period (.), the forward slash (/), the space character, the single quote (') and the double quote ("). You can use some of these characters as delimiters if all of the data in the file is contained within double quotes.

NOTE: The Bulk Load Field Delimiter character must be different from the Bulk Load Record Delimiter.

Default None

GUI Tab [Bulk tab](#) on page 254

## GSS Client Library

Attribute GSSClient (GSSC)

Description The name of the GSS client library that the driver uses to communicate with the Key Distribution Center (KDC).

The driver uses the path defined by the PATH environment variable for loading the specified client library.

Valid Values *native* | *client\_library*

where *client\_library* is a GSS client library installed on the client.

If set to *client\_library*, the driver uses the specified GSS client library.

If set to *native*, the driver uses the GSS client shipped with the operating system.

Default *native*GUI Tab [Security tab](#) on page 247

## Host

Attribute HostName (HOST)

Description The name or the IP address of the server to which you want to connect.

Valid Values *server\_name* | *IP\_address*

where:

*server\_name* is the name of the server to which you want to connect.

*IP\_address* is the IP address of the server to which you want to connect.

The IP address can be specified in either IPv4 or IPv6 format, or a combination of the two. See ["Using IP Addresses" on page 54](#) for details about these formats.

NOTE: This option is mutually exclusive with the Server Name and TNSNames File options.

Default None

GUI Tab [General tab](#) on page 244

## Host Name In Certificate

Attribute	HostNameInCertificate (HNIC)
Description	A host name for certificate validation when SSL encryption is enabled (Encryption Method=1) and validation is enabled (Validate Server Certificate=1). This option provides additional security against man-in-the-middle (MITM) attacks by ensuring that the server the driver is connecting to is the server that was requested.
Valid Values	<i>host_name</i>   <i>#SERVERNAME#</i>  where the <i>host_name</i> is the host name specified in the certificate. Consult your SSL administrator for the correct value.  If set to a host name, the driver examines the subjectAltName values included in the certificate. If a dnsName value is present in the subjectAltName values, then the driver compares the value specified for Host Name In Certificate with the dnsName value. The connection succeeds if the values match. The connection fails if the Host Name In Certificate value does not match the dnsName value.  If no subjectAltName values exist or a dnsName value is not in the list of subjectAltName values, then the driver compares the value specified for Host Name In Certificate with the commonName part of the Subject name in the certificate. The commonName typically contains the host name of the machine for which the certificate was created. The connection succeeds if the values match. The connection fails if the Host Name In Certificate value does not match the commonName. If multiple commonName parts exist in the Subject name of the certificate, the connection succeeds if the Host Name In Certificate value matches any of the commonName parts.  If set to <i>#SERVERNAME#</i> , the driver compares the host server name specified as part of a data source or connection string to the dnsName or the commonName value.
Default	None
GUI Tab	<a href="#">Security tab</a> on page 247



## IANAAppCodePage

Attribute	IANAAppCodePage (IACP)
Description	An Internet Assigned Numbers Authority (IANA) value. You must specify a value for this option if your application is not Unicode-enabled or if your database character set is not Unicode. Refer to <a href="#">Chapter 4 “Internationalization, Localization, and Unicode”</a> in the <i>DataDirect Connect Series for ODBC Reference</i> for details.  The driver uses the specified IANA code page to convert "W" (wide) functions to ANSI.  The driver and Driver Manager both check for the value of IANAAppCodePage in the following order: <ul style="list-style-type: none"> <li>■ In the connection string</li> <li>■ In the Data Source section of the system information file (odbc.ini)</li> <li>■ In the ODBC section of the system information file (odbc.ini)</li> </ul> If the driver does not find an IANAAppCodePage value, the driver uses the default value of 4 (ISO 8859-1 Latin-1).

Valid Values *IANA\_code\_page*

where *IANA\_code\_page* is one of the valid values listed in [Chapter 1 “Values for the Attribute IANAAppCodePage”](#) in the *DataDirect Connect Series for ODBC Reference*. The value must match the database character encoding and the system locale.

Default 4 (ISO 8559-1 Latin-1)

GUI Tab [Advanced tab](#) on page 245

## Initialization String

Attribute InitializationString (IS)

Description A SQL command that is issued immediately after connecting to the database to manage session settings.

NOTE: If the statement fails to execute, the connection fails and the driver reports the error returned from the server.

Valid Values *SQL\_command*

where *SQL\_command* is a valid SQL command that is supported by the database.

Example To set the date format on every connection, specify:

```
Initialization String=ALTER SESSION SET DATE_FORMAT = 'DD/MM/YYYY'
```

Default None

GUI Tab [Advanced tab](#) on page 245

## Key Password

Attribute KeyPassword (KP)

Description The password used to access the individual keys in the keystore file when SSL is enabled (Encryption Method=1) and SSL client authentication is enabled on the database server. Keys stored in a keystore can be individually password-protected. To extract the key from the keystore, the driver must have the password of the key.

Valid Values *key\_password*

where *key\_password* is the password of a key in the keystore.

Default None

GUI Tab [Security tab](#) on page 247

## Keystore

Attribute Keystore (KS)

Description The name of the directory containing the keystore file to be used when SSL is enabled (Encryption Method=1) and SSL client authentication is enabled on the database server. The keystore file contains the certificates that the client sends to the server in response to the server's certificate request. If you do not specify a directory, the current directory is used.

NOTE: The keystore and truststore files may be the same file.

Valid Values *keystore\_directory*

where *keystore\_directory* is the location of the keystore file.

Default None

GUI Tab [Security tab](#) on page 247

## Keystore Password

Attribute KeystorePassword (KSP)

Description The password used to access the keystore file when SSL is enabled (Encryption Method=1) and SSL client authentication is enabled on the database server. The keystore file contains the certificates that the client sends to the server in response to the server's certificate request.

NOTE: The keystore and truststore files may be the same file; therefore, they may have the same password.

Valid Values *keystore\_password*

where *keystore\_password* is the password of the keystore file.

Default None

GUI Tab [Security tab](#) on page 247

## Load Balance Timeout

Attribute LoadBalanceTimeout (LBT)

Description The number of seconds to keep inactive connections open in a connection pool. An inactive connection is a database session that is not associated with an ODBC connection handle, that is, a connection in the pool that is not in use by an application.

NOTE: The Min Pool Size option may cause some connections to ignore this value.

This connection option can affect performance. See ["Performance Considerations" on page 300](#) for details.

Valid Values 0 | *x*

where *x* is a positive integer that specifies a number of seconds.

If set to 0, inactive connections are kept open.

If set to *x*, inactive connections are closed after the specified number of seconds passes.

Default 0

GUI Tab [Pooling tab](#) on page 253



## Load Balancing

Attribute	LoadBalancing (LB)
Description	Determines whether the driver uses client load balancing in its attempts to connect to the database servers (primary and alternate). You can specify one or multiple alternate servers by setting the Alternate Servers option.
Valid Values	0   1  If set to 1 (Enabled), the driver uses client load balancing and attempts to connect to the database servers (primary and alternate servers) in random order.  If set to 0 (Disabled), the driver does not use client load balancing and connects to each server based on their sequential order (primary server first, then, alternate servers in the order they are specified).  NOTE: This option has no effect unless alternate servers are defined for the Alternate Servers connection option.
Default	0 (Disabled)
GUI Tab	<a href="#">Failover tab</a> on page 251

## Local Timezone Offset

Attribute	LocalTimezoneOffset (LTZO)
Description	A value to alter local time zone information. The default is "" (empty string), which means that the driver determines local time zone information from the operating system. If it is not available from the operating system, the driver defaults to using the setting on the Oracle server.
Valid Values	Valid values are specified as offsets from GMT as follows: <i>(- )HH:MM</i> . For example, -08:00 equals GMT minus 8 hours.  The driver uses the value of this option to issue an ALTER SESSION for local time zone at connection time.
Default	"" (empty string)
GUI Tab	<a href="#">Advanced tab</a> on page 245

## Lock Timeout

Attribute	LockTimeout (LTO)
Description	Specifies the amount of time, in seconds, the Oracle server waits for a lock to be released before generating an error when processing a Select...For Update statement on an Oracle 9i or higher server.  This connection option can affect performance. See <a href="#">"Performance Considerations" on page 300</a> for details.
Valid Values	-1   0   <i>x</i>  where <i>x</i> is an integer that specifies a number of seconds.

If set to -1, the server waits indefinitely for the lock to be released.

If set to 0, the server generates an error immediately and does not wait for the lock to time out.

If set to  $x$ , the server waits for the specified number of seconds for the lock to be released.

NOTE: If you are connected to an Oracle 8i server, any value greater than 0 is equivalent to the value -1.

Default -1

GUI Tab [Performance tab](#) on page 250

## Login Timeout

Attribute LoginTimeout (LT)

Description The number of seconds the driver waits for a connection to be established before returning control to the application and generating a timeout error. To override the value that is set by this connection option for an individual connection, set a different value in the SQL\_ATTR\_LOGIN\_TIMEOUT connection attribute using the SQLSetConnectAttr() function.

Valid Values -1 | 0 |  $x$

where  $x$  is a positive integer that specifies a number of seconds.

If set to -1, the connection request does not time out. The driver silently ignores the SQL\_ATTR\_LOGIN\_TIMEOUT attribute.

If set to 0, the connection request does not time out, but the driver responds to the SQL\_ATTR\_LOGIN\_TIMEOUT attribute.

If set to  $x$ , the connection request times out after the specified number of seconds unless the application overrides this setting with the SQL\_ATTR\_LOGIN\_TIMEOUT attribute.

Default 15

GUI Tab [Advanced tab](#) on page 245

## Max Pool Size

Attribute MaxPoolSize (MXPS)

Description The maximum number of connections allowed within a single connection pool. When the maximum number of connections is reached, no additional connections can be created in the connection pool.

This connection option can affect performance. See ["Performance Considerations" on page 300](#) for details.

Valid Values An integer from 1 to 65535

For example, if set to 20, the maximum number of connections allowed in the pool is 20.

Default 100

GUI Tab [Pooling tab](#) on page 253

## Min Pool Size

Attribute	MinPoolSize (MNPS)
Description	<p>The minimum number of connections that are opened and placed in a connection pool, in addition to the active connection, when the pool is created. The connection pool retains this number of connections, even when some connections exceed their Load Balance Timeout value.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 300</a> for details.</p>
Valid Values	<p>0   <i>x</i></p> <p>where <i>x</i> is an integer from 1 to 65535.</p> <p>For example, if set to 5, the start-up number of connections in the pool is 5 in addition to the current existing connection.</p> <p>If set to 0, no connections are opened in addition to the current existing connection.</p>
Default	0
GUI Tab	<a href="#">Pooling tab</a> on page 253

## Module

Attribute	Module (MOD)
Description	<p>Additional information about the client to be stored in the database. This value sets the CLIENT_IDENTIFIER value in the V\$SESSION table on the server. This value is used by the client information feature.</p> <p>This option only applies to connections to Oracle 10g R2 and higher database servers.</p> <p>NOTE: You can also specify this information using the Oracle DBMS_SESSION.SETIDENTIFIER procedure or the DBMS_APPLICATION_INFO.SET_CLIENT_INFO procedure.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 300</a> for details.</p>
Valid Values	<p><i>string</i></p> <p>where <i>string</i> is a the name of a stored procedure or the name of the application.</p> <p>If a value is not specified for this option, the driver uses the PROGRAM value in the V\$SESSION table.</p>
Default	None
GUI Tab	<a href="#">Client Monitoring tab</a> on page 260

## Password

Attribute	Password (PWD)
Description	The password that the application uses to connect to your database. The Password option cannot be specified through the driver Setup dialog box and should not be stored in a data source. It is specified through the Logon dialog box or a connection string.
Valid Values	<i>pwd</i>  where <i>pwd</i> is a valid password.
Default	None
GUI Tab	n/a

## Port Number

Attribute	PortNumber (PORT)
Description	The port number of the server listener.  NOTE: This option is mutually exclusive with the Server Name and TNSNames File options.
Valid Values	<i>port_name</i>  where the <i>port_name</i> is the port number of the server listener. Check with your database administrator for the correct number.
Default	None
GUI Tab	<a href="#">General tab</a> on page 244

## Procedure Returns Results

Attribute	ProcedureRetResults (PRR)
Description	Determines whether the driver returns result sets from stored procedures/functions.  See <a href="#">"Support of Materialized Views" on page 310</a> for details.  This connection option can affect performance. See <a href="#">"Performance Considerations" on page 300</a> for details.
Valid Values	0   1  If set to 1 (Enabled), the driver returns result sets from stored procedures/functions. When set to 1 and you execute a stored procedure that does not return result sets, you will incur a small performance penalty.  If set to 0 (Disabled), the driver does not return result sets from stored procedures.
Default	0 (Disabled)
GUI Tab	<a href="#">Advanced tab</a> on page 245

## Program ID

Attribute	ProgramID (PID)
Description	<p>The product and version information of the driver on the client to be stored in the database. This value sets the PROCESS value in the V\$SESSION table on the server. This value is used by the client information feature.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 300</a> for details.</p>
Valid Values	<p><i>string</i></p> <p>where <i>string</i> is a value that identifies the product and version of the driver on the client.</p> <p>If a value for this option is not specified, the driver uses the process ID of the session.</p>
Default	None
GUI Tab	<a href="#">Client Monitoring tab</a> on page 260

## Query Timeout

Attribute	QueryTimeout (QT)
Description	<p>The number of seconds for the default query timeout for all statements that are created by a connection. To override the value set by this connection option for an individual statement, set a different value in the SQL_ATTR_QUERY_TIMEOUT statement attribute on the SQLSetStmtAttr() function.</p>
Valid Values	<p>-1   0   <i>x</i></p> <p>where <i>x</i> is a positive integer that specifies a number of seconds.</p> <p>If set to -1, the query does not time out. The driver silently ignores the SQL_ATTR_QUERY_TIMEOUT attribute.</p> <p>If set to 0, the query does not time out, but the driver responds to the SQL_ATTR_QUERY_TIMEOUT attribute.</p> <p>If set to <i>x</i>, all queries time out after the specified number of seconds unless the application overrides this value by setting the SQL_ATTR_QUERY_TIMEOUT attribute.</p>
Default	0
GUI Tab	<a href="#">Advanced tab</a> on page 245

## Record Delimiter

Attribute	BulkLoadRecordDelimiter (BLRD)
Description	<p>Specifies the character that the driver will use to delimit the record entries in a bulk load data file.</p>
Valid Values	<p><i>x</i></p> <p>where <i>x</i> is any printable character.</p>

For simplicity, avoid using a value that can be in the data, including all alphanumeric characters, the dash(-), the colon(:), the period (.), the forward slash (/), the space character, the single quote (') and the double quote ("). You can use some of these characters as delimiters if all of the data in the file is contained within double quotes.

NOTE: The Bulk Load Record Delimiter character must be different from the Bulk Load Field Delimiter.

Default None

GUI Tab [Bulk tab](#) on page 251

## Report Codepage Conversion Errors

Attribute ReportCodepageConversionErrors (RCCE)

Description Specifies how the driver handles code page conversion errors that occur when a character cannot be converted from one character set to another.

An error message or warning can occur if an ODBC call causes a conversion error, or if an error occurs during code page conversions to and from the database or to and from the application. The error or warning generated is `Code page conversion error encountered`. In the case of parameter data conversion errors, the driver adds the following sentence: `Error in parameter x`, where `x` is the parameter number. The standard rules for returning specific row and column errors for bulk operations apply.

Valid Values 0 | 1 | 2

If set to 0 (Ignore Errors), the driver substitutes 0x1A for each character that cannot be converted and does not return a warning or error.

If set to 1 (Return Error), the driver returns an error instead of substituting 0x1A for unconverted characters.

If set to 2 (Return Warning), the driver substitutes 0x1A for each character that cannot be converted and returns a warning.

Default 0 (Ignore Errors)

GUI Tab [Advanced tab](#) on page 245

## Report Recycle Bin

Attribute ReportRecycleBin (RRB)

Description Determines whether support is provided for reporting objects that are in the Oracle Recycle Bin.

On Oracle 10g R1 and higher, when a table is dropped, it is not actually removed from the database, but placed in the recycle bin instead.

Valid Values 0 | 1

If set to 1 (Enabled), support is provided for reporting objects that are in the Oracle Recycle Bin.

If set to 0 (Disabled), the driver does not return tables contained in the recycle bin in the result sets returned from `SQLTables` and `SQLColumns`. Functionally, this means that the driver filters out any results whose Table name begins with `BIN$`.

Default 0 (Disabled)  
 GUI Tab [Advanced tab](#) on page 245

## Server Name

Attribute `ServerName (SRVR)`  
 Description Specifies a net service name that exists in the `TNSNAMES.ORA` file. The corresponding net service name entry in the `TNSNAMES.ORA` file is used to obtain Host, Port Number, and Service Name or SID information.

NOTE: This option is mutually exclusive with the Host, Port Number, SID, and Service Name options.

Valid Values `server_name`

where `server_name` is a net service name in the `TNSNAMES.ORA` file.

Default None  
 GUI Tab [General tab](#) on page 244

## Server Process Type

Attribute `ServerType (ST)`  
 Description Determines whether the connection is established using a shared or dedicated server process (dedicated thread on Windows).

This connection option can affect performance. See ["Performance Considerations" on page 300](#) for details.

Valid Values 0 | 1 | 2

If set to 0 (Server Default), the driver uses the default server process set on the server.

NOTE: The server must be configured for shared connections (the `SHARED_SERVERS` initialization parameter on the server has a value greater than 0) for the driver to be able to specify the shared server process type.

If set to 1 (Shared), the server process used is retrieved from a pool. The socket connection between the application and server is made to a dispatcher process on the server. This setting allows there to be fewer processes than the number of connections, reducing the need for server resources. Use this value when a server must handle a large number of connections.

If set to 2 (Dedicated), a server process is created to service only that connection. When that connection ends, so does the process (UNIX and Linux) or thread (Windows). The socket connection is made directly between the application and the dedicated server process or thread. When connecting to UNIX and Linux servers, a dedicated server process can provide significant performance improvement, but uses more resources on the server.

When connecting to Windows servers, the server resource penalty is insignificant. Use this value if you have a batch environment with a low number of connections.

Default 0 (Server Default)

GUI Tab [Advanced tab](#) on page 245

## Service Name

Attribute ServiceName (SN)

Description The Oracle service name that specifies the database used for the connection. The service name is a string that is the global database name—a name that is comprised of the database name and domain name, for example:

`sales.us.acme.com`

The service name is included as part of the Oracle connect descriptor, which is a description of the destination for a network connection. The service name is specified in the `CONNECT_DATA` parameter of the connect descriptor, for example:

`(CONNECT_DATA=(SERVICE_NAME=sales.us.acme.com))`

In this example, you would specify `sales.us.acme.com` as the value for the Service Name connection option.

This option is mutually exclusive with the SID, Server Name, and TNSNames File options.

Valid Values *service\_name*

where *service\_name* is the description of the destination for a network connection.

Default None

GUI Tab [General tab](#) on page 244

## SID

Attribute SID (SID)

Description The Oracle System Identifier that refers to the instance of Oracle running on the server.

NOTE: This option is mutually exclusive with the Service Name, Server Name, and TNSNames File options.

Valid Values *sid*

where *sid* is the name of the Oracle System Identifier.

Default None

GUI Tab [General tab](#) on page 244

## Timestamp Escape Mapping

Attribute TimestampEscapeMapping (TEM)

Description Determines how the driver maps Date, Time, and Timestamp literals.



Valid Values 0 | 1

If set to 0 (Oracle Version Specific), the driver determines whether to use the TO\_DATE or TO\_TIMESTAMP function based on the version of the Oracle server to which it is connected. If the driver is connected to an 8.x server, it maps the Date, Time, and Timestamp literals to the TO\_DATE function. If the driver is connected to a 9.x or higher server, it maps these escapes to the TO\_TIMESTAMP function.

If set to 1 (Oracle 8x Compatible), the driver always uses the Oracle 8.x TO\_DATE function as if connected to an Oracle 8.x server.

Default 0 (Oracle Version Specific)

GUI Tab [Advanced tab](#) on page 245

## TNSNames File

Attribute TNSNamesFile (TNF)

Description Specifies the name of the TNSNAMES.ORA file. In a TNSNAMES.ORA file, connection information for Oracle services is associated with an Oracle net service name. The entry in the TNSNAMES.ORA file specifies Host, Port Number, and Service Name or SID.

TNSNames File is ignored if no value is specified in the Server Name option. If the Server Name option is specified but the TNSNames File option is left blank, the TNS\_ADMIN environment setting is used for the TNSNAMES.ORA file path. If there is no TNS\_ADMIN setting, the ORACLE\_HOME environment setting is used. On Windows, if ORACLE\_HOME is not set, the path is taken from the Oracle section of the Registry.

Using an Oracle TNSNAMES.ORA file to centralize connection information in your Oracle environment simplifies maintenance when changes occur. If, however, the TNSNAMES.ORA file is unavailable, then it is useful to be able to open a backup version of the TNSNAMES.ORA file (TNSNames file failover). You can specify one or more backup, or alternate, TNSNAMES.ORA files.

NOTE: This option is mutually exclusive with the Host, Port Number, SID, and Service Name options.

Valid Values *path\_filename*

where *path\_filename* is the entire path, including the file name, to the TNSNAMES.ORA file.

To specify multiple TNSNAMES.ORA file locations, separate the names with a comma and enclose the locations in parentheses (you do not need parentheses for a single entry). For example:

```
(F:\server2\oracle\tnsnames.ora,
C:\oracle\product\10.1\db_1\network\admin\tnsnames.ora)
```

The driver tries to open the first file in the list. If that file is not available, then it tries to open the second file in the list, and so on.

["Connection Retry Count" on page 274](#) and ["Connection Retry Delay" on page 275](#) are also valid with TNSNames failover. The driver makes at least one attempt to open the files, and, if Connection Retry Count is enabled, more than one. If Connection Retry Delay is enabled,

the driver waits the specified number of seconds between attempts. Load Balancing is not available for TNSNames failover.

Default None

GUI Tab [General tab](#) on page 244

## Truststore

Attribute Truststore (TS)

Description The directory that contains the truststore file and the truststore file name to be used when SSL is enabled (Encryption Method=1) and server authentication is used. The truststore file contains a list of the valid Certificate Authorities (CAs) that are trusted by the client machine for SSL server authentication. If you do not specify a directory, the current directory is used.

NOTE: The truststore and keystore files may be the same file.

Valid Values *truststore\_directory\filename*

where *truststore\_directory* is the directory where the truststore file is located and *filename* is the file name of the truststore file.

Default None

GUI Tab [Security tab](#) on page 247

## Truststore Password

Attribute TruststorePassword (TSP)

Description The password that is used to access the truststore file when SSL is enabled (Encryption Method=1) and server authentication is used. The truststore file contains a list of the Certificate Authorities (CAs) that the client trusts.

NOTE: The truststore and keystore files may be the same file; therefore, they may have the same password.

Valid Values *truststore\_password*

where *truststore\_password* is a valid password for the truststore file.

Default None

GUI Tab [Security tab](#) on page 247

## Use Current Schema for SQLProcedures

Attribute UseCurrentSchema (UCS)

Description Determines whether the driver returns only procedures owned by the current user when executing SQLProcedures.

This connection option can affect performance. See ["Performance Considerations" on page 300](#) for details.

Valid Values 0 | 1

When set to 1 (Enabled), the call for SQLProcedures is optimized, but only procedures owned by the user are returned.

When set to 0 (Disabled), the driver does not specify only the current user.

Default 1 (Enabled)

GUI Tab [Performance tab](#) on page 250

## User Name

Attribute LogonID (UID)

Description The default user ID that is used to connect to your database. Your ODBC application may override this value or you may override it in the logon dialog box or connection string.

You can also use OS Authentication to connect to your Oracle database. See ["OS Authentication" on page 310](#) for details.

Valid Values *userid*

where *userid* is a valid user ID with permissions to access the database.

Default None

GUI Tab [Security tab](#) on page 247

## Validate Server Certificate

Attribute ValidateServerCertificate (VSC)

Description Determines whether the driver validates the certificate that is sent by the database server when SSL encryption is enabled (Encryption Method=1). When using SSL server authentication, any certificate sent by the server must be issued by a trusted Certificate Authority (CA). Allowing the driver to trust any certificate returned from the server even if the issuer is not a trusted CA is useful in test environments because it eliminates the need to specify truststore information on each client in the test environment.

Truststore information is specified using the Trust Store and Trust Store Password options.

Valid Values 0 | 1

If set to 1 (Enabled), the driver validates the certificate that is sent by the database server. Any certificate from the server must be issued by a trusted CA in the truststore file. If the Host Name In Certificate option is specified, the driver also validates the certificate using a host name. The Host Name In Certificate option provides additional security against man-in-the-middle (MITM) attacks by ensuring that the server the driver is connecting to is the server that was requested.

If set to 0 (Disabled), the driver does not validate the certificate that is sent by the database server. The driver ignores any truststore information specified by the Trust Store and Trust Store Password options.

Default 1 (Enabled)

GUI Tab [Security tab](#) on page 247

## Wire Protocol Mode

Attribute	WireProtocolMode (WPM)
Description	Specifies whether the driver optimizes network traffic to the Oracle server.  This connection option can affect performance. See <a href="#">"Performance Considerations" on page 300</a> for details.
Valid Values	1   2  If set to 1, the driver operates in normal wire protocol mode without optimizing network traffic.  If set to 2, the driver optimizes network traffic to the Oracle server for result sets that contain repeating data in some or all of the columns, and the repeating data is in consecutive rows. It also optimizes network traffic if the application is updating or inserting images, pictures, or long text or binary data.
Default	2
GUI Tab	<a href="#">Performance tab</a> on page 250

---

## Performance Considerations

The following connection options can enhance driver performance. You can also enhance performance through efficient application design. Refer to [Chapter 5 "Designing ODBC Applications for Performance Optimization"](#) in the *DataDirect Connect Series for ODBC Reference* for details.

**Application Using Threads (ApplicationUsingThreads):** The driver coordinates concurrent database operations (operations from different threads) by acquiring locks. Although locking prevents errors in the driver, it also decreases performance. If your application does not make ODBC calls from different threads, the driver has no reason to coordinate operations. In this case, the ApplicationUsingThreads attribute should be disabled (set to 0).

NOTE: If you are using a multi-threaded application, you must enable the Application Using Threads option.

**Array Size (ArraySize):** If this connection string attribute is set appropriately, the driver can improve performance of your application by reducing the number of round trips on the network. For example, if your application normally retrieves 200 rows, it is more efficient for the driver to retrieve 200 rows at one time over the network than to retrieve 50 rows at a time during four round trips over the network.

**Cached Cursor Limit (CachedCursorLimit):** To improve performance when your application executes concurrent Select statements, Cursor Identifiers can be cached. In this case, the Cursor Identifier is retrieved from a cache rather than being created for each connection. When an Identifier is needed, the driver takes one from its cache, if one is available, rather than creating a new one. Cached Cursor Identifiers are closed when the connection is closed. To cache Cursor Identifiers, the CachedCursorLimit attribute must be set to the appropriate number of concurrent open Select statements.

**Cached Description Limit (CachedDescLimit):** The driver can cache descriptions of Select statements and improve the performance of your ODBC application; therefore, if your application issues a fixed set of SQL queries throughout the life of the application, the description of the query should be cached. If a description is not cached, the description must be retrieved from the server, which reduces performance. The descriptions include the number of columns and the data type, length, and scale for each column. The matching is done by an exact-text match through the From clause. If the statement contains a Union or a subquery, the driver cannot cache the description.

**Catalog Functions Include Synonyms (CatalogIncludesSynonyms):** Standard ODBC behavior is to include synonyms in the result set of calls to the following catalog functions: SQLProcedures, SQLStatistics and SQLProcedureColumns. Retrieving this synonym information degrades performance. If your ODBC application does not need to return synonyms when using these catalog functions, the driver can improve performance if the CatalogIncludesSynonyms attribute is disabled (set to 0).

**Catalog Options (CatalogOptions):** If your application does not need to access the comments/remarks for database tables, performance of your application can be improved. In this case, the CatalogOptions attribute should be disabled (set to 0) because retrieving comments/remarks degrades performance. If this attribute is enabled (set to 1), result column REMARKS (for the catalog functions SQLTables and SQLColumns) and the result column COLUMN\_DEF (for the catalog function SQLColumns) return actual values.

**Connection Pooling (ConnectionPooling):** If you enable the driver to use connection pooling, you can set additional options that affect performance:

- **Load Balance Timeout:** You can define how long to keep connections in the pool. The time that a connection was last used is compared to the current time and, if the timespan exceeds the value of the Load Balance Timeout option, the connection is destroyed. The Min Pool Size option can cause some connections to ignore this value.
- **Connection Reset:** Resetting a re-used connection to the initial configuration settings impacts performance negatively because the connection must issue additional commands to the server.
- **Max Pool Size:** Setting the maximum number of connections that the pool can contain too low might cause delays while waiting for a connection to become available. Setting the number too high wastes resources.
- **Min Pool Size:** A connection pool is created when the first connection with a unique connection string connects to the database. The pool is populated with connections up to the minimum pool size, if one has been specified. The connection pool retains this number of connections, even when some connections exceed their Load Balance Timeout value.

**Default Buffer Size for Long/LOB Columns (DefaultLongDataBuffLen):** To improve performance when your application fetches images, pictures, or long text or binary data, a buffer size can be set to accommodate the maximum size of the data. The buffer size should only be large enough to accommodate the maximum amount of data retrieved; otherwise, performance is reduced by transferring large amounts of data into an oversized buffer. If your application retrieves more than 1 MB of data, the buffer size should be increased accordingly.

**Describe At Prepare (DescribeAtPrepare):** When enabled, this option requires extra network traffic. If your application does not require result set information at prepare time (for instance, you request information about the result set using SQLColAttribute(s), SQLDescribeCol, SQLNumResultCols, and so forth, before calling SQLExecute on a prepared statement), you can increase performance by disabling this option.

**Enable Bulk Load (EnableBulkLoad):** If your application performs bulk loading of data, you can improve performance by configuring the driver to use the database system's bulk load functionality instead of database array binding. The trade-off to consider for improved performance is that using the bulk load functionality can bypass data integrity constraints.

**EnableServerResultCache:** If your application connects to Oracle 11g and executes the same query multiple times, you can improve performance by using the Oracle feature server-side resultset caching. When enabled, Oracle stores the result set in database memory. On subsequent executions of the same query, the result set is returned from database memory if the underlying tables have not been modified. Without result set caching, the server would process the query and formulate a new result set.

**Enable Scrollable Cursors (EnableScrollableCursors) and Enable Static Cursors for Long Data (EnableStaticCursorsForLongData):** When your application uses Static or Keyset (Scrollable) cursors, the EnableScrollableCursors attribute must be enabled (set to 1). Also, if your application retrieves images, pictures, long text or binary data while using Static cursors, the EnableStaticCursorsForLongData attribute must be enabled (set to 1). However, this can degrade performance when retrieving long data with Static cursors as the entire result set is stored on the client. To improve performance, you might consider designing your application to retrieve long data through forward-only cursors.

#### **Data Integrity Level (DataIntegrityLevel) and Data Integrity Types**

**(DataIntegrityTypes):** Checking data integrity may adversely reduce performance because of the additional overhead (mainly CPU usage) that is required to perform the check.

**Encryption Method (EncryptionMethod), Encryption Level (EncryptionLevel), and Encryption Types (EncryptionTypes):** Data encryption may adversely affect performance because of the additional overhead (mainly CPU usage) required to encrypt and decrypt data. Using data encryption can degrade performance more than performing data integrity checks.

**Failover Mode (FailoverMode):** Although high availability that replays queries after a failure provides increased levels of protection, it can adversely affect performance because of increased overhead.

**Lock Timeout (LockTimeOut):** Sometimes users attempt to select data that is locked by another user. Oracle provides three options when accessing locked data with SELECT ... FOR UPDATE statements:

- Wait indefinitely for the lock to be released (-1)
- Return an error immediately (0)
- Return an error if the lock has not been released within a specific number of seconds (*n* seconds)

NOTE: This option is not available with Oracle 8.

Some applications may benefit by not waiting indefinitely and continuing execution; this keeps the application from hanging. The application, however, needs to handle lock timeouts properly with an appropriate timeout value; otherwise, processing time could be wasted handling lock timeouts, and deadlocks could go undetected.

To improve performance, either enter a number of seconds or enter 0 as the value for this option.

**Procedure Returns Results (ProcedureRetResults):** The driver can be tuned for improved performance if your application's stored procedures do not return results. In this case, the ProcedureRetResults attribute should be disabled (set to 0).

**Client Information:** The client information feature automatically adjusts server resources, such as CPU and memory, based on the service class associated with a workload. Therefore, an application's performance is tied to the workload to which it is assigned and, ultimately, to the service class associated with that workload. The Oracle Wire Protocol driver allows your application to set client information in the Oracle database that can be used by the client information feature to classify work. If you know that your database environment can use client information, coordinate with your database administrator to determine how setting the following options affects performance.

- **Accounting Info:** Sets the CLIENT\_INFO value of the V\$SESSION table on the server.
- **Action:** Sets ACTION column of the V\$SESSION table on the server.
- **Application Name:** Sets the dbms\_session value in the database and the PROGRAM value of the V\$SESSION table on the server.
- **Client Host Name:** Sets the MACHINE value in the V\$SESSION table on the server.
- **Client ID:** Sets the CLIENT\_IDENTIFIER value in the V\$SESSION table on the server.
- **Client User:** Sets the OSUSER value in the V\$SESSION table on the server.
- **Module:** Sets the CLIENT\_IDENTIFIER value in the V\$SESSION table on the server.
- **Program ID:** Sets the PROCESS value in the V\$SESSION table on the server.
- **Server Process Type (ServerType):** When using a dedicated server connection, a server process on UNIX (a thread on Windows) is created to serve only your application connection. When you disconnect, the process goes away. The socket connection is made directly between your application and this dedicated server process. This can provide tremendous performance improvements, but will use significantly more resources on UNIX servers. Because this is a thread on Oracle servers running on Windows platforms, the additional resource usage on the server is significantly less. This option should be set to 2 (dedicated) when you have a batch environment with lower numbers of connections, your Oracle server has excess processing capacity and memory available when at maximum load, or if you have a performance-sensitive application that would be degraded by sharing Oracle resources with other applications.

**Use Current Schema for SQLProcedures (UseCurrentSchema):** If your application needs to access database objects owned only by the current user, performance of your application can be improved. In this case, the UseCurrentSchema attribute should be enabled (set to 1). When this attribute is enabled, the driver returns only database objects owned by the current user when executing catalog functions. Calls to catalog functions are

optimized by grouping queries. Enabling this attribute is equivalent to passing the Logon ID used on the connection as the SchemaName argument to the catalog functions.

## Data Types

Table 8-2 shows how the Oracle data types are mapped to the standard ODBC data types. "Unicode Support" on page 306 lists Oracle to Unicode data type mappings.

**Table 8-2. Oracle Data Types**

Oracle	ODBC
BFILE <sup>1</sup>	SQL_LONGVARBINARY
BINARY DOUBLE <sup>2</sup>	SQL_REAL
BINARY FLOAT <sup>2</sup>	SQL_DOUBLE
BLOB <sup>2</sup>	SQL_LONGVARBINARY
CHAR	SQL_CHAR
CLOB <sup>2</sup>	SQL_LONGVARCHAR
DATE	SQL_TYPE_TIMESTAMP
LONG	SQL_LONGVARCHAR
LONG RAW	SQL_LONGVARBINARY
NUMBER	SQL_DOUBLE
NUMBER (p,s)	SQL_DECIMAL
RAW	SQL_VARBINARY
TIMESTAMP <sup>3</sup>	SQL_TIMESTAMP
TIMESTAMP WITH LOCAL TIMEZONE <sup>3</sup>	SQL_TIMESTAMP
TIMESTAMP WITH TIMEZONE <sup>3 4</sup>	SQL_VARCHAR
VARCHAR2	SQL_VARCHAR
XMLType <sup>5</sup>	SQL_LONGVARCHAR

1. Read-Only

2. Supported only on Oracle 10g and higher.

3. Supported only on Oracle 9i and higher.

4. Timestamp with timezone mapping changes based on the setting of the Fetch TSWTZ as Timestamp option only on Oracle 10g R2 and higher.

5. Supported only on Oracle 9i R2 and higher.

The Oracle Wire Protocol driver does not support any object types (also known as abstract data types). When the driver encounters an object type during data retrieval, it returns an Unknown Data Type error (SQL State HY000).

See "Retrieving Data Type Information" on page 59 for more information about data types.



## XMLType

Oracle 9i R2 and higher supports the XMLType data type. The driver supports tables containing columns whose data type is specified as XMLType.

When inserting or updating XMLType columns, the data to be inserted or updated must be in the form of an XMLType data type. The database provides functions to construct XMLType data. The `xmlData` argument to `xmltype()` may be specified as a string literal.

### Examples

If the XMLType column is created with the CLOB storage type, then the driver returns it without use of the special `getClobVal` function, that is, you can use:

```
SELECT XML_col FROM table_name...
```

instead of

```
SELECT XML_col.getClobVal()...
```

The following example illustrates using the CLOB storage type:

```
CREATE TABLE po_xml_tab(
  poId NUMBER(10),
  poDoc XMLTYPE
)
XMLType COLUMN poDoc
STORE AS CLOB (
  TABLESPACE lob_seg_ts
  STORAGE (INITIAL 4096 NEXT 4096)
  CHUNK 4096 NOCACHE LOGGING
);
```

The next example illustrates how to create a table, insert data, and retrieve data when not using the CLOB storage type:

```
CREATE TABLE PURCHASEORDER (PODOCUMENT sys.XMLTYPE);
```

The PURCHASEORDER table contains one column—PODOCUMENT—with a data type of XMLType (`sys.XMLTYPE`). The next step is to insert one purchase order, created by the static function `sys.XMLTYPE.createXML`:

```
INSERT INTO PURCHASEORDER (PODOCUMENT) values (
  sys.XMLTYPE.createXML(
    '
    <PurchaseOrder>
      <Reference>BLAKE-2001062514034298PDT</Reference>

      <Actions>
        <Action>
          <User>KING</User>
          <Date/>
        </Action>
      </Actions>
      <Reject/>

      <Requester>David E. Blake</Requester>
    '
  )
);
```

```

<User>BLAKE</User>
<CostCenter>S30</CostCenter>
<ShippingInstructions>
  <name>David E. Blake</name>
  <address>400 Oracle Parkway Redwood Shores, CA, 94065 USA</address>
  <telephone>650 999 9999</telephone>
</ShippingInstructions>

<SpecialInstructions>Air Mail</SpecialInstructions>
<LineItems>
  <LineItem ItemNumber="1">
    <Description>The Birth of a Nation</Description>
    <Part Id="EE888" UnitPrice="65.39" Quantity="31"/>
  </LineItem>
</LineItems>
</PurchaseOrder>
');

```

Use the getClobVal function to retrieve the data:

```
SELECT p.podocument.getClobVal() FROM PURCHASEORDER p;
```

---

## Unicode Support

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

If the database character set is set to UTF-8, the Oracle driver maps the Oracle data types to Unicode data types as shown in the following table:

Oracle Data Type	Mapped to . . .
CHAR	SQL_WCHAR
CLOB	SQL_WLONGVARCHAR
VARCHAR2	SQL_WVARCHAR
LONG	SQL_WLONGVARCHAR

The driver also continues to map these Oracle data types to the normal character data types. See ["Data Types" on page 304](#) for these mappings. The only exception to this is that when the Enable N-CHAR Support option is enabled, the N-CHAR types are mapped to the Unicode types SQL\_WCHAR, SQL\_WVARCHAR, and SQL\_WLONGVARCHAR, and the normal character types are mapped to the data types SQL\_CHAR, SQL\_LONGVARCHAR, and SQL\_VARCHAR, regardless of the character set on the Oracle server.

---

## Advanced Features

The driver supports the following advanced features:

- Failover
- Client Information
- Security
- Connection Pooling
- DataDirect Bulk Load

### Failover

The driver supports failover and its related connection options. Failover connection options are located on the [Failover tab](#) of the driver Setup dialog box. See ["Using Failover" on page 65](#) for a general description of failover and its implementation.

### Client Information

Oracle provides a client information feature that allows an administrator to define different work load classifications and store client information associated with a connection. These workload classifications can be assigned different priorities and resource allocations. To enable applications to leverage these work load classifications fully, the Oracle Wire Protocol driver provides connection options for setting the session properties that are used in identifying a work load. These options are located on the [Client Monitoring tab](#) of the driver Setup dialog box. See ["Using Client Information" on page 75](#) for a general description of client information and its implementation.

### Security

The driver supports authentication in addition to encryption and data integrity checks. Security connection options are located on the [Security tab](#) and [Advanced Security tab](#) of the driver Setup dialog box. See ["Using Security" on page 77](#) for a general description of security and its implementation. The following security information is specific to the Oracle Wire Protocol Driver.

### Authentication

If you are using Kerberos, verify that your environment meets the requirements listed in [Table 8-3](#) before you configure the driver for Kerberos authentication.

**Table 8-3. Kerberos Authentication Requirements for the Oracle Wire Protocol Driver**

Component	Requirements
Database server	<p>The database server must be administered by the same domain controller that administers the client and must be running one of the following databases:</p> <ul style="list-style-type: none"> <li>■ Oracle 11g (R1 and R2)</li> <li>■ Oracle 10g (R1 and R2)</li> <li>■ Oracle 9i (R2)</li> </ul> <p>In addition, Oracle Advanced Security is required.</p>
Kerberos server	<p>The Kerberos server is the machine where the user IDs for authentication are administered. The Kerberos server is also the location of the Kerberos KDC.</p> <p>Network authentication must be provided by one of the following methods:</p> <ul style="list-style-type: none"> <li>■ Windows Active Directory on one of the following operating systems: Windows Server 2003 or Windows 2000 Server Service Pack 3 or higher</li> <li>■ MIT Kerberos 1.4.2 or higher</li> </ul>
Client	<p>The client must be administered by the same domain controller that administers the database server.</p>

## Encryption and Data Integrity

To enable support for SSL connections to Oracle, the Oracle database must be configured with the Oracle Advanced Security bundle. This is an option available from Oracle as an add-on to Oracle Enterprise Edition Servers.

The driver also supports encryption and data integrity checks through Oracle Advanced Security. Oracle Advanced Security provides the Advanced Encryption Standard (AES), DES, 3DES, and RC4 symmetric cryptosystems for protecting the confidentiality of network traffic.

Encrypting network data provides data privacy so that unauthorized parties cannot view and alter clear text data as it passes over the network. Attacks on intercepted data include data modification and replay attacks.

- In a data modification attack, an unauthorized party intercepts transmitted data, alters it, and retransmits it. For example, suppose a customer order for 5 widgets for delivery to an office in San Francisco is intercepted. A data modification attack might change the quantity to 500 and the delivery address to a warehouse in Los Angeles, and then retransmit the order.
- In a replay attack, a set of valid data is retransmitted a number of times. For example, an order for 100 widgets is intercepted and then retransmitted ten times so the final order quantity equals 1,000 widgets.

Because data integrity protection operates independently from the encryption process, you can enable data integrity with or without enabling encryption.

## Connection Pooling

The driver supports connection pooling and its related connection options. Connection pooling connection options are located on the [Pooling tab](#) of the driver Setup dialog box. See ["Using DataDirect Connection Pooling" on page 82](#) for a general description of connection pooling and its implementation.

## DataDirect Bulk Load

The driver supports DataDirect bulk load and its related connection options. Bulk load connection options are located on the [Bulk tab](#) of the driver Setup dialog box. See ["Using DataDirect Bulk Load" on page 85](#) for a general description of DataDirect bulk load and its implementation.

The Oracle Wire Protocol driver uses array binding instead of DataDirect Bulk Load when the Oracle server version is older than Oracle 9i R1 (9.0.1)

## Limitations

- A bulk operation is not allowed in a manual transaction if it is not the first event.
- Once a bulk operation is started, any non-bulk operation is disallowed until the transaction is committed.
- The Oracle Wire Protocol driver currently does not support the use of LONG and LONG RAW data types with array binding.
- Because of Oracle limitations, issuing a SELECT statement to determine a row count may return different results before and after a bulk load operation.
- Oracle does not support literal values in a bulk load operation. You must use parameter markers for all columns being loaded.

---

## MTS Support



On Windows, the driver can take advantage of Microsoft Transaction Server (MTS) capabilities, specifically, the Distributed Transaction Coordinator (DTC) using the XA Protocol. For a general discussion of MTS and DTC, refer to the help file of the Microsoft Transaction Server SDK.

NOTE: The DataDirect Connect *for* ODBC 32-bit drivers can operate in a 64-bit Windows environment; however, they do not support DTC in this environment. Only the DataDirect Connect64 *for* ODBC 64-bit drivers support DTC in a 64-bit Windows environment.

To enable DTC support, you must be connected to an Oracle 8.1.7 or higher server.

---

## OS Authentication

On Windows, UNIX, and Linux, Oracle has a feature called OS Authentication that allows you to connect to an Oracle database via the operating system user name and password. To connect, use a forward slash ( / ) for the user name and leave the password blank. To configure the Oracle server, refer to the Oracle server documentation. This feature is valid when connecting from a data source, a connection string, or a logon dialog box.

---

## Support for Oracle RAC

Oracle introduced Real Application Clusters (RAC) with Oracle 9i, and RAC is also a key feature of Oracle 10g. Oracle RAC allows a single physical Oracle database to be accessed by concurrent instances of Oracle running across several different CPUs.

An Oracle RAC is composed of a group of independent servers, or nodes, that cooperate as a single system. A cluster architecture such as this provides applications access to more computing power when needed, while allowing computing resources to be used for other applications when database resources are not as heavily required. For example, in the event of a sudden increase in network traffic, an Oracle RAC can distribute the load over many nodes, a feature referred to as *server load balancing*. Oracle RAC features are available to you simply by connecting to an Oracle RAC system with a DataDirect Connect Series *for* ODBC driver. There is no additional configuration required.

*Connection failover* and *client load balancing* can be used in conjunction with an Oracle RAC system, but they are not specifically part of Oracle RAC. See ["Using Failover" on page 65](#) for details about how these features work in DataDirect Connect Series *for* ODBC drivers.

---

## Support of Materialized Views

When connected to an Oracle 9i or higher server, the Oracle Wire Protocol driver supports the creation of materialized views. Materialized views are like any other database view with the following additions: the results are stored as a database object and the results can be updated on a schedule determined by the Create View statement.

Materialized views improve performance for data warehousing and replication. Refer to the Oracle documentation for more information about materialized views.

## Stored Procedure Results

When you enable the Procedure Returns Results connection option, the driver returns result sets from stored procedures/functions. In addition, `SQLGetInfo(SQL_MULT_RESULTS_SETS)` returns Y and `SQLGetInfo(SQL_BATCH_SUPPORT)` returns `SQL_BS_SELECT_PROC`. If this option is enabled and you execute a stored procedure that does not return result sets, you 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 G1%rowtype;
end GEN_PACKAGE;
Create or replace procedure GEN_PROCEDURE1 (
  rset IN OUT GEN_PACKAGE.GTABLE2CHARCOL, icol INTEGER) as
begin
  open rset for select CHARCOL from GTABLE2
    where INTEGERCOL <= icol order by INTEGERCOL;
end;
```

When executing the stored procedures with result sets, do not include the result set arguments (Oracle ref cursors) in the list of procedure parameters. The result set returned through the ref cursor is returned as a normal ODBC result set.

```
{call GEN_PROCEDURE1 (?)}
```

where ? is the parameter for the icol argument.

For more information, refer to your Oracle SQL documentation.

## Unexpected Characters

Users are sometimes surprised when they insert a character into a database, only to have a different character displayed when they fetch it from the database. There are many reasons this can happen, but it most often involves code page issues, not driver errors.

Client and server machines in a database system each use code pages, which can be identified by a name or a number, such as `Shift_JIS` (Japanese) or `cp1252` (Windows English). A code page is a mapping that associates a sequence of bits, called a code point, with a specific character. Code pages include the characters and symbols of one or more languages. Regardless of geographical location, a machine can be configured to use a specific code page. Most of the time, a client and database server would use similar, if not identical, code pages. For example, a client and server might use two different Japanese code pages, such as `Shift_JIS` and `EUC_JP`, but they would still share many Japanese characters in common. These characters might, however, be represented by different code points in each code page. This introduces the need to convert between code pages to

maintain data integrity. In some cases, no one-to-one character correspondence exists between the two code points. This causes a substitution character to be used, which can result in displaying an unexpected character on a fetch.

When the driver on the client machine opens a connection with the database server, the driver determines the code pages being used on the client and the server. This is determined from the Active Code Page on a Windows-based machine. If the client machine is UNIX-based, the driver checks the IANAAppCodePage option. If it does not find a specific setting for IACP, it defaults to a value of ISO\_8859\_1.

If the client and server code pages are compatible, the driver transmits data in the code page of the server. Even though the pages are compatible, a one-to-one correspondence for every character may not exist. If the client and server code pages are completely dissimilar, for example, Russian and Japanese, then many substitutions occur because very few, if any, of the characters are mapped between the two code pages.

The following is a specific example of an unexpected character:

- The Windows client machine is running code page cp1252.
- The Oracle server is running code page ISO-8859-P1.
- When you insert a Euro character (€) from the Windows client and then fetch it back, an upside down question mark (¿) is displayed on the client instead of the Euro symbol.

This substitution occurs because the Euro character does not exist within the characters defined by the ISO-8859-P1 character set on the Oracle server. The Oracle server records the code point for its substitution character in the table instead of the code point for the Euro. This code point is an upside down question mark in the Windows cp1252 code page.

This is not a driver error. The code page of the Oracle database could not recognize the Euro code point and used its substitution character in the table. The best way to avoid these problems is to use the same code page on both the client and server machines.

You can check the native code point stored in the Oracle database using SQL\*Plus with a SQL statement similar to the following:

```
SELECT dump(columnname, 1016) FROM yourtable;
```

Check the returned hexadecimal values to verify whether the data you intended to reside in the table is there. If it appears that Oracle substituted a different code point, then check the Oracle database code page to see if your intended character exists. If your character does not exist in the code page, then no error is involved; Oracle simply does not recognize the original character, and uses its substitution character instead.

---

## Persisting a Result Set as an XML Data File

The driver allows you to persist a result as an XML data file with embedded schema. See ["Persisting a Result Set as an XML Data File" on page 60](#) for details about implementation.

NOTE: If you are persisting a result set that contains Long data, you must enable the EnableStaticCursorsforLongData connection string attribute.



---

## Isolation and Lock Levels Supported

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

Refer to [Chapter 7 “Locking and Isolation Levels”](#) in the *DataDirect Connect Series for ODBC Reference* for details.

---

## SQL Support

The driver supports the core SQL grammar.

---

## ODBC Conformance Level

The driver is Level 1 compliant, that is, it supports all ODBC Core and Level 1 functions.

In addition, the following functions are supported:

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

Refer to [Chapter 2 “ODBC API and Scalar Functions”](#) in the *DataDirect Connect Series for ODBC Reference* for a list of the API functions supported by the Oracle Wire Protocol driver.

---

## Number of Connections and Statements Supported

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

---

## Using Parameter Arrays

Oracle 8i and higher databases natively support parameter arrays, and the Oracle Wire Protocol driver, in turn, supports them. When designing an application for performance, using native parameter arrays for bulk inserts or updates, for example, can improve performance. Refer to [Chapter 5 “Designing ODBC Applications for Performance Optimization”](#) in the *DataDirect Connect Series for ODBC Reference* for more information about using arrays of parameters to improve performance.

## 9 The PostgreSQL Wire Protocol Driver

The DataDirect Connect *for* ODBC and DataDirect Connect64 *for* ODBC PostgreSQL Wire Protocol driver (the PostgreSQL Wire Protocol driver) each support the following PostgreSQL database servers:

- PostgreSQL 9.0, 9.1
- PostgreSQL 8.2, 8.3, 8.4

The PostgreSQL Wire Protocol driver is supported in the Windows, UNIX, and Linux environments. See ["Environment-Specific Information" on page 45](#) for detailed information about the Windows, UNIX, and Linux environments supported by this driver.

Refer to the readme file shipped with your DataDirect product for the file name of the PostgreSQL Wire Protocol driver.

---

### Driver Requirements

The driver has no client requirements.

---

### Configuring and Connecting to Data Sources

After you install the driver, you configure data sources to connect to the database. See [Chapter 1 "Quick Start Connect" on page 35](#) for an explanation of different types of data sources. The data source contains connection options that allow you to tune the driver for specific performance. If you want to use a data source but need to change some of its values, you can either modify the data source or override its values at connection time through a connection string.

If you choose to use a connection string, you must use specific connection string attributes. See ["Using a Connection String" on page 323](#) and [Table 9-1 on page 325](#) for an alphabetical list of driver connection string attributes and their initial default values.

#### Data Source Configuration in the UNIX/Linux odbc.ini File

On UNIX and Linux, you must set up the proper ODBC environment before configuring data sources. See ["Environment Configuration" on page 37](#) for basic setup information and ["Environment Variables" on page 97](#) for more detail about this procedure.

Data sources for UNIX and Linux are stored in the system information file (by default, odbc.ini). If you have a Motif GUI environment on Linux, you can configure and modify data sources through the DataDirect ODBC Data Source Administrator for Linux (the Linux ODBC Administrator) using a driver Setup dialog box. (See ["Configuration Through the Administrator" on page 100](#) for a detailed explanation of the Administrator.)

If you do not have a GUI environment, you can configure and modify data sources directly by editing the `odbc.ini` file and storing default connection values there. See ["Configuration Through the System Information \(odbc.ini\) File" on page 103](#) for detailed information about the specific steps necessary to configure a data source.

[Table 9-1 on page 325](#) lists driver connection string attributes that must be used in the `odbc.ini` file to set the value of the attributes. Note that only the long name of the attribute can be used in the file. The default listed in the table is the initial default value when the driver is installed.

## Data Source Configuration through a GUI

On Windows, data sources are stored in the Windows Registry. You can configure and modify data sources through the ODBC Administrator using a driver Setup dialog box, as described in this section.

On UNIX and Linux, data sources are stored in the `odbc.ini` file. On Linux, you can configure and modify data sources through the Linux ODBC Administrator using a driver Setup dialog box, as described in this section.

**NOTE:** This book shows dialog box images that are specific to Windows. If you are using the drivers in the Linux environment, the dialog box that you see may differ slightly from the Windows version. Windows-only and UNIX-only connection options are specifically noted by icons in the Setup dialog box descriptions.

When the driver is first installed, the values of its connection options are set by default. These values appear on the driver Setup dialog box tabs when you create a new data source. You can change these default values by modifying the data source. In the following procedure, the description of each tab is followed by a table that lists the connection options for that tab and their initial default values. This table links you to a complete description of the options and their connection string attribute equivalents. The connection string attributes are used to override the default values of the data source if you want to change these values at connection time.

### To configure a PostgreSQL data source:

- 1 Start the ODBC Administrator:
  - On Windows, start the ODBC Administrator by selecting its icon from the DataDirect Connect program group.
  - On Linux, change to the `install_dir/tools` directory and, at a command prompt, enter:
 

```
odbcadmin
```

 where `install_dir` is the path to the product installation directory.
- 2 Select a tab:
  - **User DSN:** If you are configuring an existing user data source, select the data source name and click **Configure** to display the driver Setup dialog box.
 

If you are configuring a new user data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.

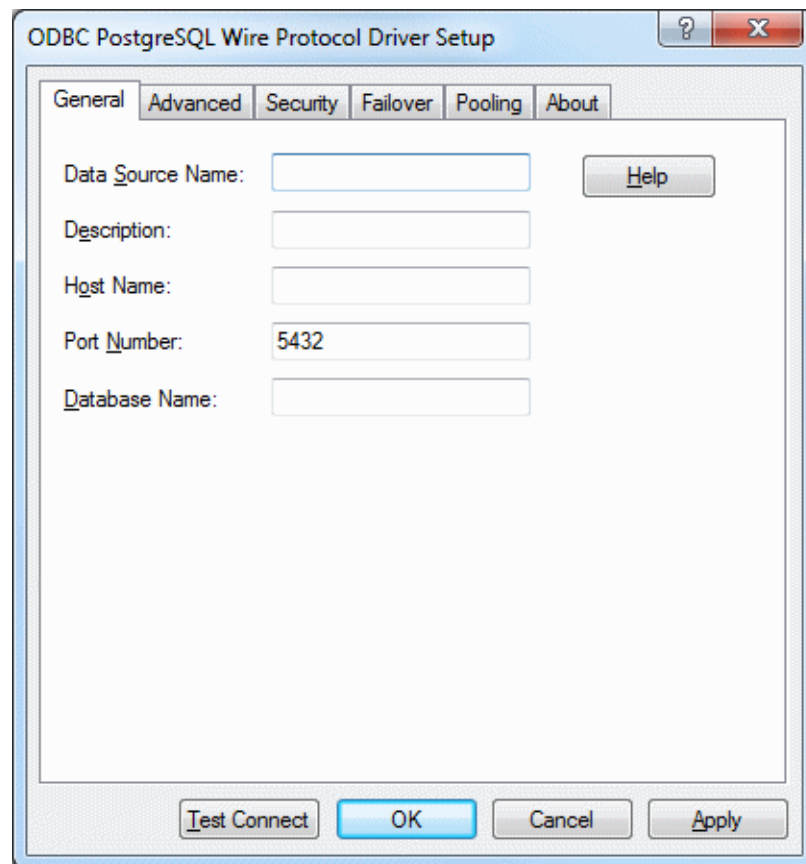
- **System DSN:** If you are configuring an existing system data source, select the data source name and click **Configure** to display the driver Setup dialog box.

If you are configuring a new system data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.

- **File DSN:** If you are configuring an existing file data source, select the data source file and click **Configure** to display the driver Setup dialog box.

If you are configuring a new file data source, click **Add** to display a list of installed drivers; then, select a driver. Click **Advanced** if you want to specify attributes; otherwise, click **Next** to proceed. Specify a name for the data source and click **Next**. Verify the data source information; then, click **Finish** to display the driver Setup dialog box.

The General tab of the Setup dialog box appears.



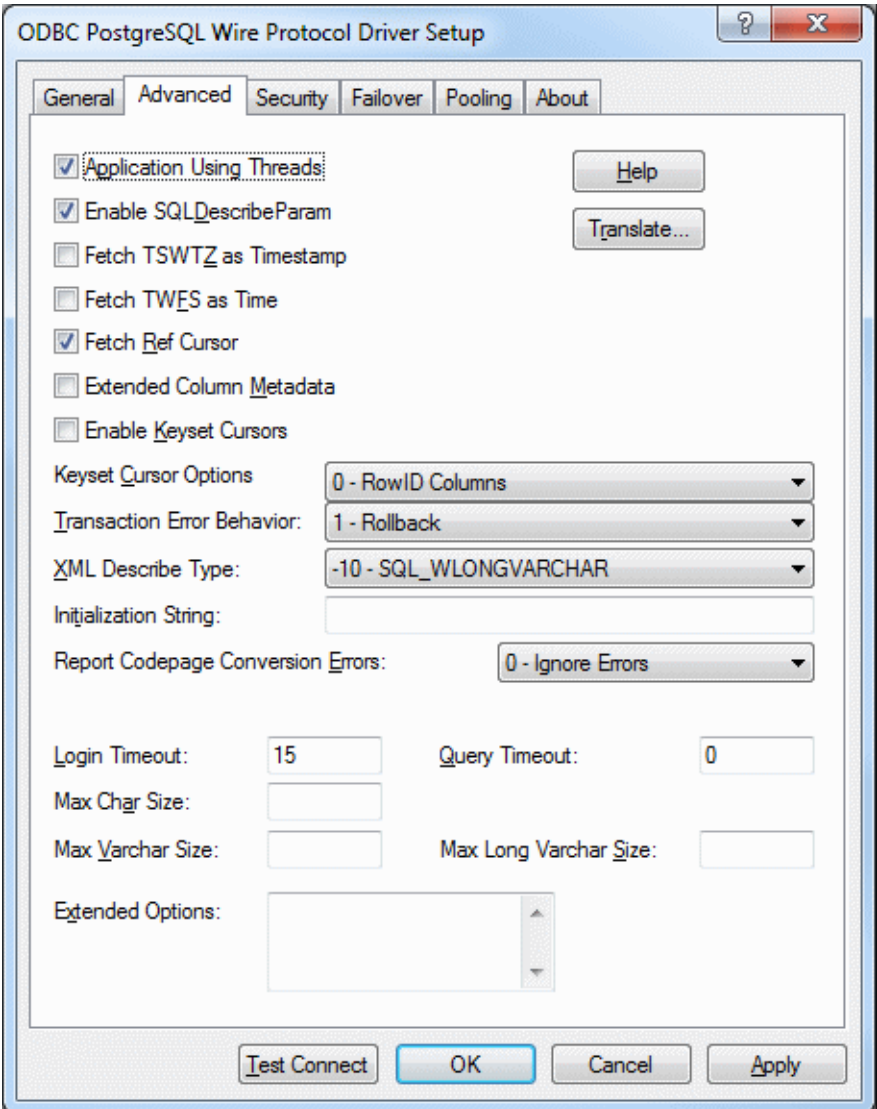
NOTE: The General tab displays only fields that are required for creating a data source. The fields on all other tabs are optional, unless noted otherwise.

- 3 On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: General	Default
<a href="#">Data Source Name (see page 328)</a>	None
<a href="#">Description (see page 329)</a>	None

Connection Options: General	Default
<a href="#">Host Name (see page 334)</a>	None
<a href="#">Port Number (see page 340)</a>	5432
<a href="#">Database Name (see page 329)</a>	None

- Optionally, click the **Advanced** tab to specify additional data source settings.



On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Advanced	Default
<a href="#">Application Using Threads (see page 327)</a>	Enabled
<a href="#">Enable SQLDescribeParam (see page 330)</a>	Enabled
<a href="#">Fetch TSWTZ as Timestamp (see page 333)</a>	Disabled
<a href="#">Fetch TWFS as Time (see page 333)</a>	Disabled
<a href="#">Fetch Ref Cursors (see page 332)</a>	Enabled

Connection Options: Advanced	Default
<a href="#">Extended Column Metadata (see page 330)</a>	Disabled
<a href="#">Enable Keyset Cursors (see page 329)</a>	Disabled
<a href="#">Keyset Cursor Options (see page 337)</a>	0 - RowID Columns
<a href="#">Transaction Error Behavior (see page 342)</a>	1 - Rollback
<a href="#">XML Describe Type (see page 344)</a>	-10 - SQL_WLONGVARCHAR
<a href="#">Report Codepage Conversion Errors (see page 341)</a>	0 - Ignore Errors
<a href="#">Login Timeout (see page 338)</a>	15
<a href="#">Query Timeout (see page 341)</a>	0
<a href="#">Initialization String (see page 335)</a>	None
<a href="#">Max Char Size (see page 338)</a>	None
<a href="#">Max Varchar Size (see page 339)</a>	None
<a href="#">Max Long Varchar Size (see page 339)</a>	None
<a href="#">IANAAppCodePage (see page 335)</a>	4 (ISO 8559-1 Latin-1)
UNIX ONLY	

**Extended Options:** Type a semi-colon separated list of connection options and their values. Use this configuration option to set the value of undocumented connection options that are provided by Progress DataDirect customer support. You can include any valid connection option in the Extended Options string, for example:

```
Database=Server1;UndocumentedOption1=value[;UndocumentedOption2=value;]
```

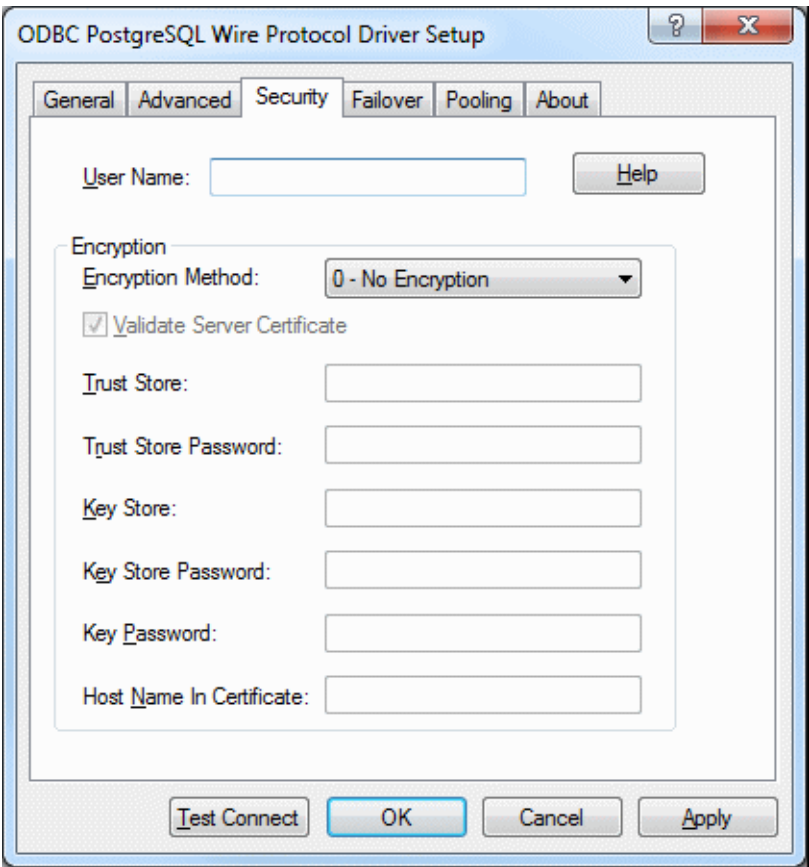
If the Extended Options string contains option values that are also set in the setup dialog or data source, the values of the options specified in the Extended Options string take precedence. However, connection options that are specified on a connection string override any option value specified in the Extended Options string.

**NOTE:** Do not specify the Extended Options configuration option in a connection string, or the driver will return an error. Instead, applications should specify the individual undocumented connection options in the connection string.

**Translate:** Click **Translate** to display the Select Translator dialog box, which lists the translators specified in the ODBC Translators section of the Registry. Progress 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.

- 5 Optionally, click the **Security** tab to specify security data source settings.



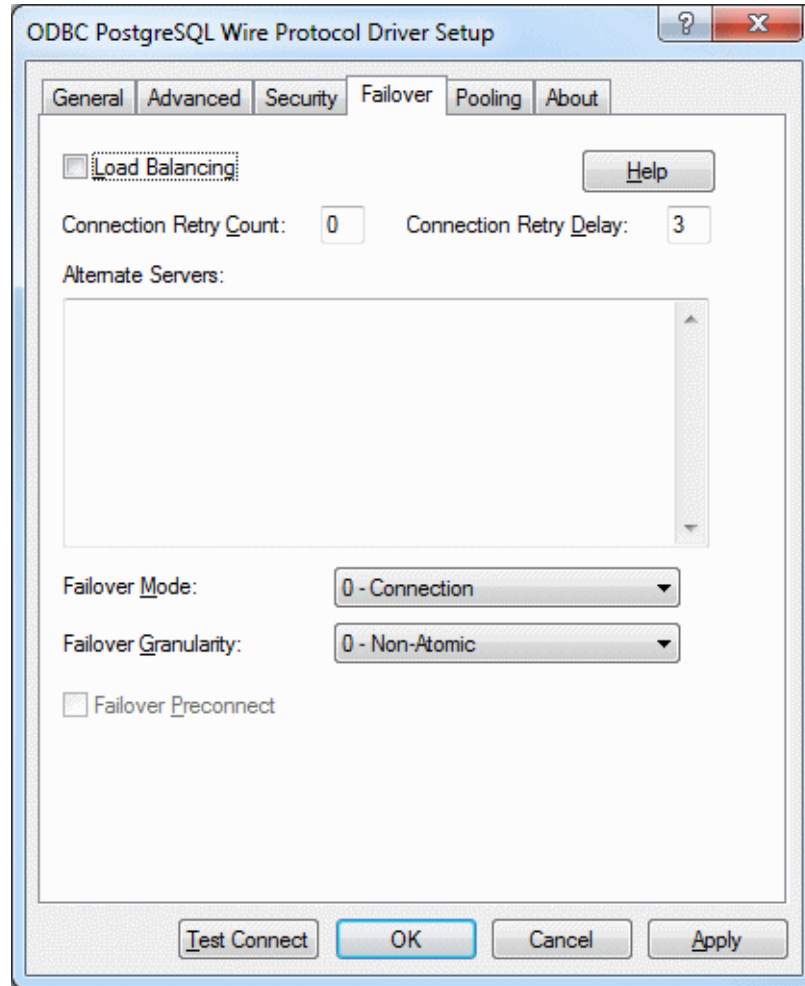
See ["Using Security" on page 77](#) for a general description of encryption and its configuration requirements.

On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Security	Default
<a href="#">User Name (see page 343)</a>	None
<a href="#">Encryption Method (see page 330)</a>	0 - No Encryption
<a href="#">Validate Server Certificate (see page 343)</a>	Enabled
<a href="#">TrustStore (see page 342)</a>	None
<a href="#">TrustStore Password (see page 342)</a>	None
<a href="#">Key Store (see page 336)</a>	None
<a href="#">Key Store Password (see page 336)</a>	None
<a href="#">Key Password (see page 336)</a>	None
<a href="#">Host Name In Certificate (see page 334)</a>	None

- 6 Optionally, click the **Failover** tab to specify failover data source settings.



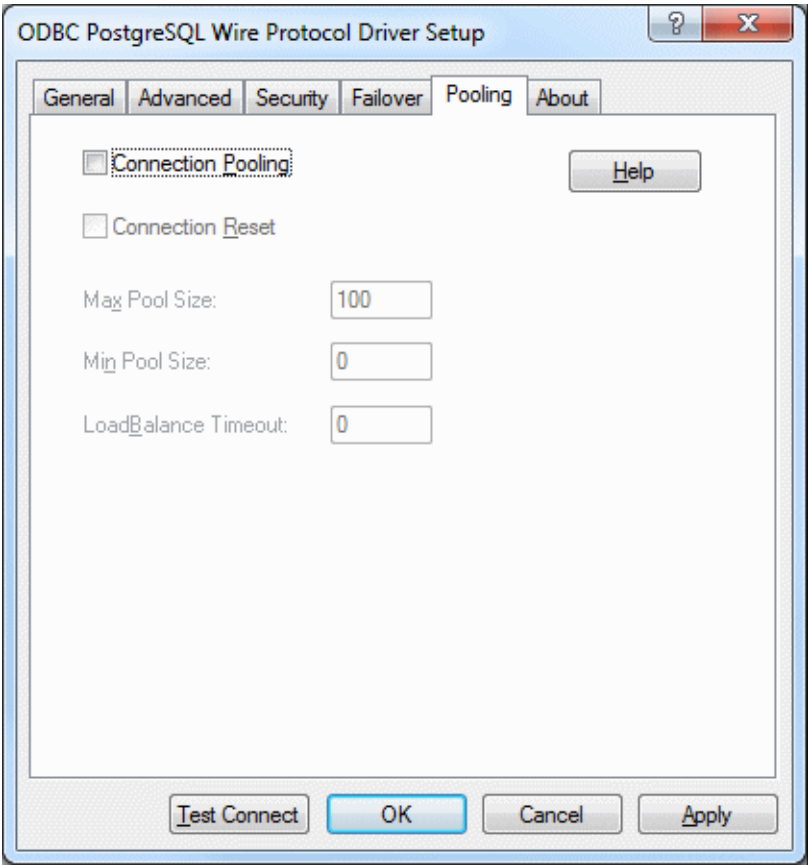


See ["Using Failover" on page 65](#) for a general description of failover and its related connection options.

On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Failover	Default
<a href="#">Load Balancing (see page 338)</a>	Disabled
<a href="#">Connection Retry Count (see page 328)</a>	0
<a href="#">Connection Retry Delay (see page 328)</a>	3
<a href="#">Alternate Servers (see page 326)</a>	None
<a href="#">Failover Mode (see page 332)</a>	0 - Connection
<a href="#">Failover Granularity (see page 331)</a>	0 - Non-Atomic
<a href="#">Failover Preconnect (see page 332)</a>	Disabled

- 7 Optionally, click the **Pooling** tab to specify pooling data source settings.



See ["Using DataDirect Connection Pooling" on page 82](#) for a general description of connection pooling.

On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Pooling	Default
<a href="#">Connection Pooling (see page 327)</a>	Disabled
<a href="#">Connection Reset (see page 327)</a>	Disabled
<a href="#">Max Pool Size (see page 339)</a>	100
<a href="#">Min Pool Size (see page 340)</a>	0
<a href="#">Load Balance Timeout (see page 337)</a>	0

- 8 At any point during the configuration process, you can click **Test Connect** to attempt to connect to the data source using the connection options specified in the driver Setup dialog box. A logon dialog box appears (see ["Using a Logon Dialog Box" on page 324](#) for details). 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 incorrect environment or connection value, it displays an appropriate error message. Click **OK**.

NOTE: If you are configuring alternate servers for use with the connection failover feature, be aware that the Test Connect button tests only the primary server, not the alternate servers.

- 9 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.

## Using a Connection String

If you want to use a connection string for connecting to a database, or if your application requires it, you must specify either a DSN (data source name), a File DSN, or a DSN-less connection in the string. The difference is whether you use the DSN=, FILEDSN=, or the DRIVER= keyword in the connection string, as described in the ODBC specification. A DSN or FILEDSN connection string tells the driver where to find the default connection information. Optionally, you may specify *attribute=value* pairs in the connection string to override the default values stored in the data source.

The DSN connection string has the form:

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

The FILEDSN connection string has the form:

```
FILEDSN=filename.dsn[;attribute=value[;attribute=value]...]
```

The DSN-less connection string specifies a driver instead of a data source. All connection information must be entered in the connection string because the information is not stored in a data source.

The DSN-less connection string has the form:

```
DRIVER=[{driver_name}][;attribute=value[;attribute=value]...]
```

[Table 9-1](#) lists the long and short names for each attribute, as well as the initial default value when the driver is first installed. You can specify either long or short names in the connection string.

An example of a DSN connection string with overriding attribute values for PostgreSQL Wire Protocol is:

```
DSN=Accounting;UID=JOHN;PWD=XYZZY
```

A FILEDSN connection string is similar except for the initial keyword:

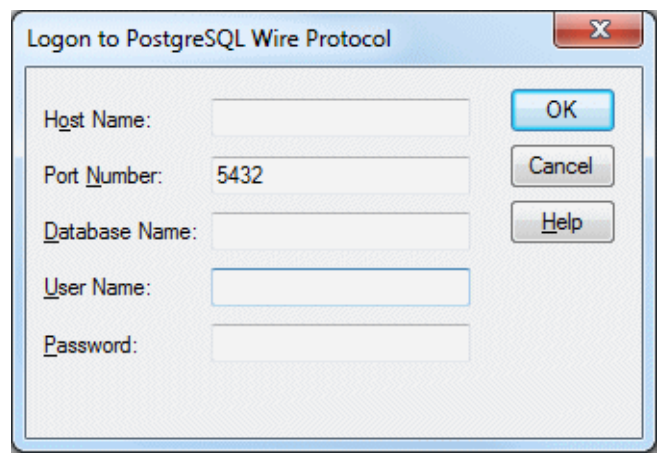
```
FILEDSN=PostgreSQLLWP.dsn;UID=JOHN;PWD=XYZZY
```

A DSN-less connection string must provide all necessary connection information:

```
DRIVER=DataDirect 7.1 PostgreSQL Wire Protocol;  
HOST=PostgreSQLServer;PORT=5432;UID=JOHN;PWD=XYZZY;DB=Pgredb1
```

## 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, provide the following information:

- 1 In the Host Name field, type either the name or the IP address of the server to which you want to connect.  
  
The IP address can be specified in either IPv4 or IPv6 format, or a combination of the two. See ["Using IP Addresses" on page 54](#) for details concerning these formats.
- 2 In the Port Number field, type the number of your PostgreSQL listener. Check with your database administrator for the correct number.
- 3 In the Database Name field, type the name of the database to which you want to connect.
- 4 If required, type your PostgreSQL user name.
- 5 If required, type your PostgreSQL password.
- 6 Click **OK** to log on to the PostgreSQL database installed on the server you specified and to update the values in the Registry.

---

## Connection Option Descriptions

The following connection option descriptions are listed alphabetically by the GUI name that appears on the driver Setup dialog box. The connection string attribute name, along with its short name, is listed immediately underneath the GUI name. For example:

### Application Using Threads

Attribute    ApplicationUsingThreads (AUT)

In most cases, the GUI name and the attribute name are the same; however, some exceptions exist. If you need to look up an option by its connection string attribute name, please refer to the alphabetical table of connection string attribute names.

Also, a few connection string attributes, for example, Password, do not have equivalent options that appear on the GUI. They are in the list of descriptions alphabetically by their attribute names.

Table 9-1 lists the connection string attributes supported by the PostgreSQL Wire Protocol driver.

**Table 9-1. PostgreSQL Wire Protocol Attribute Names**

Attribute (Short Name)	Default
AlternateServers (ASRV)	None
ApplicationUsingThreads (AUT)	1 (Enabled)
ConnectionReset (CR)	0 (Disabled)
ConnectionRetryCount (CRC)	0
ConnectionRetryDelay (CRD)	3
Database (DB)	None
DataSourceName (DSN)	None
Description (n/a)	None
EnableDescribeParam (EDP)	1 (Enabled)
EnableKeysetCursors (EKC)	0 (Disabled)
EncryptionMethod (EM)	0 (No Encryption)
ExtendedColumnMetaData (ECMD)	0 (Disabled)
FailoverGranularity (FG)	0 (Non-Atomic)
FailoverMode (FM)	0 (Connection)
FailoverPreconnect (FP)	0 (Disabled)
FetchRefCursors (FRC)	1 (Enabled)
FetchTSWTZasTimestamp (FTSWTZAT)	0 (Disabled)
FetchTWFSasTime (FTWFSAT)	0 (Disabled)
HostName (HOST)	None
HostNameInCertificate (HNIC)	None
IANAAppCodePage (IACP)	4 (ISO 8559-1 Latin-1)
UNIX ONLY	
InitializationString (IS)	None
KeyPassword (KP)	None
Keyset Cursor Options	0 - RowID Columns
Keystore (KS)	None
KeystorePassword (KSP)	None
LoadBalanceTimeout (LBT)	0
LoadBalancing (LB)	0 (Disabled)
LoginTimeout (LT)	15
LogonID (UID)	None
MaxLongVarcharSize (MLVS)	None
MaxPoolSize (MXPS)	100

**Table 9-1. PostgreSQL Wire Protocol Attribute Names** (cont.)

Attribute (Short Name)	Default
Max Char Size	None
MaxVarcharSize (MVS)	None
MinPoolSize (MNPS)	0
Password (PWD)	None
Pooling (POOL)	0 (Disabled)
PortNumber (PORT)	5432
QueryTimeout (QT)	0
ReportCodepageConversionErrors (RCCE)	0 (Ignore Errors)
TransactionErrorBehavior (TEB)	1 (Rollback Transaction)
Truststore (TS)	None
TruststorePassword (TSP)	None
ValidateServerCertificate (VSC)	1 (Enabled)
XMLDescribeType (XDT)	-10

## Alternate Servers

Attribute	AlternateServers (ASRV)
Description	A list of alternate database servers to which the driver tries to connect if the primary database server is unavailable. Specifying a value for this option enables connection failover for the driver. The value you specify must be in the form of a string that defines the physical location of each alternate server. All of the other required connection information for each alternate server is the same as what is defined for the primary server connection.
Valid Values	<p>(HostName=hostvalue:PortNumber=portvalue:Database=databasevalue[, . . .])</p> <p>You must specify the host name, port number, and database name of each alternate server.</p> <p>NOTE: An alternate server address in IPv6 format must be enclosed in double quotation marks.</p>
Example	<p>The following Alternate Servers value defines two alternate database servers for connection failover:</p> <pre>AlternateServers=(HostName=PostgreSQLServer: PortNumber=5431:Database=Pgredb1, HostName=255.201.11.24:PortNumber=5432:Database=Pgredb2)</pre>
Default	None
GUI Tab	<a href="#">Failover tab</a> on page 320

## Application Using Threads

Attribute	ApplicationUsingThreads (AUT)
Description	<p>Determines whether the driver works with applications using multiple ODBC threads.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 344</a> for details.</p>
Valid Values	<p>0   1</p> <p>If set to 1 (Enabled), the driver works with single-threaded and multi-threaded applications.</p> <p>If set to 0 (Disabled), the driver does not work with multi-threaded applications. If using the driver with single-threaded applications, this value avoids additional processing required for ODBC thread-safety standards.</p>
Default	1 (Enabled)
GUI Tab	<a href="#">Advanced tab</a> on page 318

## Connection Pooling

Attribute	Pooling (POOL)
Description	<p>Specifies whether to use the driver's connection pooling.</p> <p>NOTE: The application must be thread-enabled to use connection pooling.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 344</a> for details.</p>
Valid Values	<p>0   1</p> <p>If set to 1 (Enabled), the driver uses connection pooling.</p> <p>If set to 0 (Disabled), the driver does not use connection pooling.</p>
Default	0 (Disabled)
GUI Tab	<a href="#">Pooling tab</a> on page 321

## Connection Reset

Attribute	ConnectionReset (CR)
Description	<p>Determines whether the state of connections that are removed from the connection pool for reuse by the application is reset to the initial configuration of the connection.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 344</a> for details.</p>
Valid Values	<p>0   1</p> <p>If set to 1 (Enabled), the state of connections removed from the connection pool for reuse by an application is reset to the initial configuration of the connection. Resetting the state can negatively impact performance because additional commands must be sent over the network to the server to reset the state of the connection.</p>

If set to 0 (Disabled), the state of connections is not reset.

Default 0 (Disabled)

GUI Tab [Pooling tab](#) on page 321

## Connection Retry Count

Attribute ConnectionRetryCount (CRC)

Description The number of times the driver retries connection attempts to the primary database server, and if specified, alternate servers until a successful connection is established.

This option and the Connection Retry Delay connection option, which specifies the wait interval between attempts, can be used in conjunction with failover.

Valid Values 0 |  $x$

where  $x$  is a positive integer from 1 to 65535.

If set to 0, the driver does not try to connect after the initial unsuccessful attempt.

If set to  $x$ , the driver retries connection attempts the specified number of times. If a connection is not established during the retry attempts, the driver returns an error that is generated by the last server to which it tried to connect.

Default 0

GUI Tab [Failover tab](#) on page 320

## Connection Retry Delay

Attribute ConnectionRetryDelay (CRD)

Description The number of seconds the driver waits between connection retry attempts when Connection Retry Count is set to a positive integer.

This option and the Connection Retry Count connection option can be used in conjunction with failover.

Valid Values 0 |  $x$

where  $x$  is a positive integer from 1 to 65535.

If set to 0, there is no delay between retries.

If set to  $x$ , the driver waits the specified number of seconds between connection retry attempts.

Default 3

GUI Tab [Failover tab](#) on page 320

## Data Source Name

Attribute DataSourceName (DSN)

Description The name of a data source in your Windows Registry or odbc.ini file.



Valid Values	<i>string</i> where <i>string</i> is the name of a data source.
Default	None
GUI Tab	<a href="#">General tab</a> on page 317

## Database Name

Attribute	Database (DB)
Description	The name of the database to which you want to connect.
Valid Values	<i>database_name</i> where <i>database_name</i> is the name of a valid database.
Default	None
GUI Tab	<a href="#">General tab</a> on page 317

## Description

Attribute	Description (n/a)
Description	An optional long description of a data source. This description is not used as a runtime connection attribute, but does appear in the ODBC.INI section of the Registry and in the <code>odbc.ini</code> file.
Valid Values	<i>string</i> where <i>string</i> is a description of a data source.
Default	None
GUI Tab	<a href="#">General tab</a> on page 317

## Enable Keyset Cursors

Attribute	EnableKeysetCursors (EKC)
Description	Determines whether the driver emulates keyset cursors to provide scrollable keyset cursors to an ODBC application.
Valid Values	0   1  If set to 1 (Enabled), the driver emulates keyset cursors.  If set to 0 (Disabled), the driver does not emulate keyset cursors. If an application requests a keyset cursor and this option is set to 0, the driver uses a static cursor and returns a message that a different value was used.
Default	0 (Disabled)
GUI Tab	<a href="#">Advanced tab</a> on page 318

## Enable SQLDescribeParam

Attribute	EnableDescribeParam (EDP)
Description	Determines whether SQLDescribeParam returns the Datatype, ParameterSize, DecimalDigits, and Nullable information for parameters in a prepared statement.
Valid Values	0   1  If set to 1 (enabled), SQLDescribeParam returns the Datatype, ParameterSize, DecimalDigits, and Nullable information for parameters in a prepared statement.  If set to 0 (disabled), the driver does not support SQLDescribeParam and returns the message: <code>Driver does not support this function.</code>
Default	1 (Enabled)
GUI Tab	<a href="#">Advanced tab</a> on page 318

## Encryption Method

Attribute	EncryptionMethod (EM)
Description	The method the driver uses to encrypt data sent between the driver and the database server. If the specified encryption method is not supported by the database server, the connection fails and the driver returns an error.  This connection option can affect performance. See <a href="#">"Performance Considerations" on page 344</a> for details.
Valid Values	0   1  If set to 0 (No Encryption), data is not encrypted.  If set to 1 (SSL), data is encrypted using SSL. If the server supports protocol negotiation, the driver and server negotiate the use of TLS v1, SSL v3, or SSL v2 in that order.
Default	0 (No Encryption)
GUI Tab	<a href="#">Security tab</a> on page 320

## Extended Column Metadata

Attribute	ExtendedColumnMetaData (ECMD)
Description	Determines how the driver returns column metadata when using SQLDescribeCol and SQLColAttribute.
Valid Values	0   1  If set to 1 (Enabled), SQLDescribeCol returns the actual values for Data Type, Column Size, Decimal Digits, and Nullable. SQLColAttribute returns the actual values for: <ul style="list-style-type: none"> <li>■ SQL_DESC_CATALOG_NAME: <i>catalog_name</i></li> <li>■ SQL_DESC_TABLE_NAME: <i>table_name</i></li> <li>■ SQL_DESC_BASE_COLUMN_NAME: <i>base_column_name</i></li> <li>■ SQL_DESC_LOCAL_TYPE_NAME: <i>local_type_name</i></li> </ul>

- SQL\_DESC\_NULLABLE: *nullable*
- SQL\_DESC\_AUTO\_UNIQUE\_VALUE: *auto\_unique\_value*

If set to 0 (Disabled), SQLDescribeCol returns the Data Type, Column Size, and Decimal Digits for the column. The value SQL\_NULLABLE\_UNKNOWN is returned for Nullable. SQLColAttribute returns the following attribute values:

- SQL\_DESC\_CATALOG\_NAME: empty string
- SQL\_DESC\_TABLE\_NAME: empty string
- SQL\_DESC\_BASE\_COLUMN\_NAME: empty string
- SQL\_DESC\_LOCAL\_TYPE\_NAME: empty string
- SQL\_DESC\_NULLABLE: SQL\_NULLABLE\_UNKNOWN
- SQL\_DESC\_AUTO\_UNIQUE\_VALUE: SQL\_FALSE

Default 0 (Disabled)

GUI Tab [Advanced tab](#) on page 318

## Failover Granularity

Attribute FailoverGranularity (FG)

Description Determines whether the driver fails the entire failover process or continues with the process if errors occur while trying to reestablish a lost connection.

This option applies only when Failover Mode is set to 1 (Extended Connection) or 2 (Select).

The Alternate Servers option specifies one or multiple alternate servers for failover and is required for all failover methods.

Valid Values 0 | 1 | 2 | 3

If set to 0 (Non-Atomic), the driver continues with the failover process and posts any errors on the statement on which they occur.

If set to 1 (Atomic) the driver fails the entire failover process if an error is generated as the result of anything other than executing and repositioning a Select statement. If an error is generated as a result of repositioning a result set to the last row position, the driver continues with the failover process, but generates a warning that the Select statement must be reissued.

If set to 2 (Atomic Including Repositioning), the driver fails the entire failover process if any error is generated as the result of restoring the state of the connection or the state of work in progress.

If set to 3 (Disable Integrity Check), the driver does not verify that the rows that were restored during the failover process match the original rows. This value applies only when Failover Mode is set to 2 (Select).

Default 0 (Non-Atomic)

GUI Tab [Failover tab](#) on page 320

## Failover Mode

Attribute	FailoverMode (FM)
Description	<p>Specifies the type of failover method the driver uses.</p> <p>The Alternate Servers option specifies one or multiple alternate servers for failover and is required for all failover methods.</p>
Valid Values	<p>0   1   2</p> <p>If set to 0 (Connection), the driver provides failover protection for new connections only.</p> <p>If set to 1 (Extended Connection), the driver provides failover protection for new and lost connections, but not any work in progress.</p> <p>If set to 2 (Select), the driver provides failover protection for new and lost connections. In addition, it preserves the state of work performed by the last Select statement executed.</p>
Default	0 (Connection)
GUI Tab	<a href="#">Failover tab</a> on page 320

## Failover Preconnect

Attribute	FailoverPreconnect (FP)
Description	<p>Specifies whether the driver tries to connect to the primary and an alternate server at the same time.</p> <p>This attribute applies only when Failover Mode is set to 1 (Extended Connection) or 2 (Select) and at least one alternate server is specified.</p> <p>The Alternate Servers option specifies one or multiple alternate servers for failover and is required for all failover methods.</p>
Valid Values	<p>0   1</p> <p>If set to 0 (Disabled), the driver tries to connect to an alternate server only when failover is caused by an unsuccessful connection attempt or a lost connection. This value provides the best performance, but your application typically experiences a short wait while the failover connection is attempted.</p> <p>If set to 1 (Enabled), the driver tries to connect to the primary and an alternate server at the same time. This can be useful if your application is time-sensitive and cannot absorb the wait for the failover connection to succeed.</p>
Default	0 (Disabled)
GUI Tab	<a href="#">Failover tab</a> on page 320

## Fetch Ref Cursors

Attribute	FetchRefCursors (FRC)
Description	Determines whether the driver returns refcursors from stored procedures as results sets.
Valid Values	0   1

If set to 1 (Enabled), the driver returns refcursors from stored procedures as result sets. The driver fetches all the data from the refcursor and then closes the refcursor. If a stored procedure returns multiple refcursors, the driver generates multiple result sets, one for each refcursor returned.

If set to 0 (Disabled), the driver returns the cursor name for refcursors. The application must fetch the actual data from the refcursor using the cursor name and must close the cursor before additional processing can be done on the statement. The application must close the cursor regardless of whether it actually fetches data from the cursor.

Default 1 (Enabled)

GUI Tab [Advanced tab](#) on page 318

## Fetch TSWTZ as Timestamp

Attribute FetchTSWTZasTimestamp (FTSWTZAT)

Description Determines whether the driver returns column values with the timestamp with time zone data type as the ODBC data type SQL\_TYPE\_TIMESTAMP or SQL\_VARCHAR.

Valid Values 0 | 1

If set to 1 (Enabled), the driver returns column values with the timestamp with time zone data type as the ODBC type SQL\_TYPE\_TIMESTAMP. The time zone information in the fetched value is truncated. Use this value if your application needs to process values the same way as TIMESTAMP columns.

If set to 0 (Disabled), the driver returns column values with the timestamp with time zone data type as the ODBC data type SQL\_VARCHAR. Use this value if your application requires the time zone information in the fetched value.

Default 0 (Disabled)

GUI Tab [Advanced tab](#) on page 318

## Fetch TWFS as Time

Attribute FetchTWFSasTime (FTWFSAT)

Description Determines whether the driver returns column values with the time data type as the ODBC data type SQL\_TYPE\_TIME or SQL\_TYPE\_TIMESTAMP.

Valid Values 0 | 1

If set to 1 (Enabled), the driver returns column values with the time data type as the ODBC data type SQL\_TYPE\_TIME. The fractional seconds portion of the value is truncated.

If set to 0 (Disabled), the driver returns column values with the time data type as the ODBC data type SQL\_TYPE\_TIMESTAMP. The fractional seconds portion of the value is preserved. Time columns are not searchable when they are described and fetched as timestamp

NOTE: When returning time with fractional seconds data as SQL\_TYPE\_TIMESTAMP, the Year, Month and Day parts of the timestamp must be set to zero.

Default 0 (Disabled)  
 GUI Tab [Advanced tab](#) on page 318

## Host Name

Attribute HostName (HOST)  
 Description The name or the IP address of the server to which you want to connect.  
 Valid Values *server\_name* | *IP\_address*

where:

*server\_name* is the name of the server to which you want to connect.

*IP\_address* is the IP address of the server to which you want to connect.

The IP address can be specified in either IPv4 or IPv6 format, or a combination of the two. See ["Using IP Addresses" on page 54](#) for details about these formats.

Default None  
 GUI Tab [General tab](#) on page 317

## Host Name In Certificate

Attribute HostNameInCertificate (HNIC)  
 Description A host name for certificate validation when SSL encryption is enabled (Encryption Method=1) and validation is enabled (Validate Server Certificate=1). This option provides additional security against man-in-the-middle (MITM) attacks by ensuring that the server the driver is connecting to is the server that was requested.  
 Valid Values *host\_name* | *#SERVERNAME#*

where the *host\_name* is the host name specified in the certificate. Consult your SSL administrator for the correct value.

If set to a host name, the driver examines the subjectAltName values included in the certificate. If a dnsName value is present in the subjectAltName values, then the driver compares the value specified for Host Name In Certificate with the dnsName value. The connection succeeds if the values match. The connection fails if the Host Name In Certificate value does not match the dnsName value.

If no subjectAltName values exist or a dnsName value is not in the list of subjectAltName values, then the driver compares the value specified for Host Name In Certificate with the commonName part of the Subject name in the certificate. The commonName typically contains the host name of the machine for which the certificate was created. The connection succeeds if the values match. The connection fails if the Host Name In Certificate value does not match the commonName. If multiple commonName parts exist in the Subject name of the certificate, the connection succeeds if the Host Name In Certificate value matches any of the commonName parts.

If set to *#SERVERNAME#*, the driver compares the host server name specified as part of a data source or connection string to the dnsName or the commonName value.

Default None  
 GUI Tab [Security tab](#) on page 320

## IANAAppCodePage

Attribute IANAAppCodePage (IACP)

Description An Internet Assigned Numbers Authority (IANA) value. You must specify a value for this option if your application is not Unicode-enabled or if your database character set is not Unicode. Refer to [Chapter 4 “Internationalization, Localization, and Unicode”](#) in the *DataDirect Connect Series for ODBC Reference* for details.

The driver uses the specified IANA code page to convert "W" (wide) functions to ANSI.

The driver and Driver Manager both check for the value of IANAAppCodePage in the following order:

- In the connection string
- In the Data Source section of the system information file (odbc.ini)
- In the ODBC section of the system information file (odbc.ini)

If the driver does not find an IANAAppCodePage value, the driver uses the default value of 4 (ISO 8859-1 Latin-1).

Valid Values *IANA\_code\_page*

where *IANA\_code\_page* is one of the valid values listed in [Chapter 1 “Values for the Attribute IANAAppCodePage”](#) in the *DataDirect Connect Series for ODBC Reference*. The value must match the database character encoding and the system locale.

Default 4 (ISO 8559-1 Latin-1)

GUI Tab [Advanced tab](#) on page 318

## Initialization String

Attribute InitializationString (IS)

Description A SQL command that is issued immediately after connecting to the database to manage session settings.

NOTE: If the statement fails to execute, the connection fails and the driver reports the error returned from the server.

Valid Values *SQL\_command*

where *SQL\_command* is a valid SQL command that is supported by the database.

Example To set the date format on every connection, specify:

```
Set DateStyle='ISO, MDY'
```

Default None

GUI Tab [Advanced tab](#) on page 318

## Key Password

Attribute	KeyPassword (KP)
Description	The password used to access the individual keys in the keystore file when SSL is enabled (Encryption Method=1) and SSL client authentication is enabled on the database server. Keys stored in a keystore can be individually password-protected. To extract the key from the keystore, the driver must have the password of the key.
Valid Values	<i>key_password</i>  where <i>key_password</i> is the password of a key in the keystore.
Default	None
GUI Tab	<a href="#">Security tab</a> on page 320

## Key Store

Attribute	Keystore (KS)
Description	The name of the directory containing the keystore file to be used when SSL is enabled (Encryption Method=1) and SSL client authentication is enabled on the database server. The keystore file contains the certificates that the client sends to the server in response to the server's certificate request. If you do not specify a directory, the current directory is used.  NOTE: The keystore and truststore files may be the same file.
Valid Values	<i>keystore_directory</i>  where <i>keystore_directory</i> is the location of the keystore file.
Default	None
GUI Tab	<a href="#">Security tab</a> on page 320

## Key Store Password

Attribute	KeystorePassword (KSP)
Description	The password used to access the keystore file when SSL is enabled (Encryption Method=1) and SSL client authentication is enabled on the database server. The keystore file contains the certificates that the client sends to the server in response to the server's certificate request.  NOTE: The keystore and truststore files may be the same file; therefore, they may have the same password.
Valid Values	<i>keystore_password</i>  where <i>keystore_password</i> is the password of the keystore file.
Default	None
GUI Tab	<a href="#">Security tab</a> on page 320



## Keyset Cursor Options

Attribute	KeysetCursorOptions (KCO)
Description	<p>Determines which columns are used to comprise the keyset that the driver uses to create the initial keyset on which cursor operations are based. PostgreSQL does not offer a true row identifier column; the driver instead uses a hidden system column provided by the PostgreSQL database, ctid. Because the database might reassign the ID following a Vacuum operation, the driver can be configured to also include other columns to help ensure that data integrity is maintained.</p> <p>NOTE: This option has no effect unless the <a href="#">Enable Keyset Cursors</a> connection option is enabled.</p>
Valid Values	<p>0   1</p> <p>If set to 1 - RowID and Searchable Columns (Enabled), the driver uses a combination of every non-LOB column in the Select list and the ctid hidden column to build the keyset. By adding other Select list fields to the keyset, the driver is able to indicate the row cannot be found if the IDs change following a Vacuum operation.</p> <p>If set to 0 - RowID Columns (Disabled), the driver uses the ctid hidden system column.</p>
Default	0 - RowID Columns
GUI Tab	<a href="#">Advanced tab</a> on page 318

## Load Balance Timeout

Attribute	LoadBalanceTimeout (LBT)
Description	<p>The number of seconds to keep inactive connections open in a connection pool. An inactive connection is a database session that is not associated with an ODBC connection handle, that is, a connection in the pool that is not in use by an application.</p> <p>NOTE: The Min Pool Size option may cause some connections to ignore this value.</p> <p>This connection option can affect performance. See "<a href="#">Performance Considerations</a>" on <a href="#">page 344</a> for details.</p>
Valid Values	<p>0   <i>x</i></p> <p>where <i>x</i> is a positive integer that specifies a number of seconds.</p> <p>If set to 0, inactive connections are kept open.</p> <p>If set to <i>x</i>, inactive connections are closed after the specified number of seconds passes.</p>
Default	0
GUI Tab	<a href="#">Pooling tab</a> on page 321

## Load Balancing

Attribute	LoadBalancing (LB)
Description	Determines whether the driver uses client load balancing in its attempts to connect to the database servers (primary and alternate). You can specify one or multiple alternate servers by setting the Alternate Servers option.
Valid Values	0   1  If set to 1 (Enabled), the driver uses client load balancing and attempts to connect to the database servers (primary and alternate servers) in random order.  If set to 0 (Disabled), the driver does not use client load balancing and connects to each server based on their sequential order (primary server first, then, alternate servers in the order they are specified).  NOTE: This option has no effect unless alternate servers are defined for the Alternate Servers connection option.
Default	0 (Disabled)
GUI Tab	<a href="#">Failover tab</a> on page 320

## Login Timeout

Attribute	LoginTimeout (LT)
Description	The number of seconds the driver waits for a connection to be established before returning control to the application and generating a timeout error. To override the value that is set by this connection option for an individual connection, set a different value in the SQL_ATTR_LOGIN_TIMEOUT connection attribute using the SQLSetConnectAttr() function.
Valid Values	-1   0   <i>x</i>  where <i>x</i> is a positive integer that specifies a number of seconds.  If set to -1, the connection request does not time out. The driver silently ignores the SQL_ATTR_LOGIN_TIMEOUT attribute.  If set to 0, the connection request does not time out, but the driver responds to the SQL_ATTR_LOGIN_TIMEOUT attribute.  If set to <i>x</i> , the connection request times out after the specified number of seconds unless the application overrides this setting with the SQL_ATTR_LOGIN_TIMEOUT attribute.
Default	15
GUI Tab	<a href="#">Advanced tab</a> on page 318

## Max Char Size

Attribute	MaxCharSize (MCS)
Description	Specifies the maximum size of columns of type SQL_CHAR that the driver describes through result set descriptions and catalog functions.

Valid Values	A positive integer from 1 to 10485760
	When not specified, the actual size of the columns from the database is persisted to the application.
	If you specify a value that is not in the specified range, the driver uses the maximum value of the SQL_CHAR data type, 10485760.
Default	None. The actual size of the columns from the database is persisted to the application.
GUI Tab	Advanced

## Max Long Varchar Size

Attribute	MaxLongVarcharSize (MLVS)
Description	Specifies the maximum size of columns of type SQL_LONGVARCHAR that the driver describes through result set descriptions and catalog functions.
Valid Values	A positive integer from 1 to x
	where x is maximum size of the SQL_LONGVARCHAR data type.
Default	None. The actual size of the columns from the database is persisted to the application.
GUI Tab	Advanced

## Max Pool Size

Attribute	MaxPoolSize (MXPS)
Description	The maximum number of connections allowed within a single connection pool. When the maximum number of connections is reached, no additional connections can be created in the connection pool.
	This connection option can affect performance. See <a href="#">"Performance Considerations" on page 344</a> for details.
Valid Values	An integer from 1 to 65535
	For example, if set to 20, the maximum number of connections allowed in the pool is 20.
Default	100
GUI Tab	<a href="#">Pooling tab</a> on page 321

## Max Varchar Size

Attribute	MaxVarcharSize (MVS)
Description	Specifies the maximum size of columns of type SQL_VARCHAR that the driver describes through result set descriptions and catalog functions.
Valid Values	A positive integer from 1 to x
	where x is maximum size of the SQL_VARCHAR data type.

Default	None. The actual size of the columns from the database is persisted to the application.
GUI Tab	Advanced

## Min Pool Size

Attribute	MinPoolSize (MNPS)
Description	<p>The minimum number of connections that are opened and placed in a connection pool, in addition to the active connection, when the pool is created. The connection pool retains this number of connections, even when some connections exceed their Load Balance Timeout value.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 344</a> for details.</p>
Valid Values	<p>0   <i>x</i></p> <p>where <i>x</i> is an integer from 1 to 65535.</p> <p>For example, if set to 5, the start-up number of connections in the pool is 5 in addition to the current existing connection.</p> <p>If set to 0, no connections are opened in addition to the current existing connection.</p>
Default	0
GUI Tab	<a href="#">Pooling tab</a> on page 321

## Password

Attribute	Password (PWD)
Description	The password that the application uses to connect to your database. The Password option cannot be specified through the driver Setup dialog box and should not be stored in a data source. It is specified through the Logon dialog box or a connection string.
Valid Values	<p><i>pwd</i></p> <p>where <i>pwd</i> is a valid password.</p>
Default	None
GUI Tab	n/a

## Port Number

Attribute	PortNumber (PORT)
Description	The port number of the server listener.
Valid Values	<p><i>port_name</i></p> <p>where the <i>port_name</i> is the port number of the server listener. Check with your database administrator for the correct number.</p>
Default	5432
GUI Tab	<a href="#">General tab</a> on page 317

## Query Timeout

Attribute	QueryTimeout (QT)
Description	The number of seconds for the default query timeout for all statements that are created by a connection. To override the value set by this connection option for an individual statement, set a different value in the SQL_ATTR_QUERY_TIMEOUT statement attribute on the SQLSetStmtAttr() function.
Valid Values	-1   0   <i>x</i>  where <i>x</i> is a positive integer that specifies a number of seconds.  If set to -1, the query does not time out. The driver silently ignores the SQL_ATTR_QUERY_TIMEOUT attribute.  If set to 0, the query does not time out, but the driver responds to the SQL_ATTR_QUERY_TIMEOUT attribute.  If set to <i>x</i> , all queries time out after the specified number of seconds unless the application overrides this value by setting the SQL_ATTR_QUERY_TIMEOUT attribute.
Default	0
GUI Tab	<a href="#">Advanced tab</a> on page 318

## Report Codepage Conversion Errors

Attribute	ReportCodepageConversionErrors (RCCE)
Description	Specifies how the driver handles code page conversion errors that occur when a character cannot be converted from one character set to another.  An error message or warning can occur if an ODBC call causes a conversion error, or if an error occurs during code page conversions to and from the database or to and from the application. The error or warning generated is <code>Code page conversion error encountered</code> . In the case of parameter data conversion errors, the driver adds the following sentence: <code>Error in parameter <i>x</i></code> , where <i>x</i> is the parameter number. The standard rules for returning specific row and column errors for bulk operations apply.
Valid Values	0   1   2  If set to 0 (Ignore Errors), the driver substitutes 0x1A for each character that cannot be converted and does not return a warning or error.  If set to 1 (Return Error), the driver returns an error instead of substituting 0x1A for unconverted characters.  If set to 2 (Return Warning), the driver substitutes 0x1A for each character that cannot be converted and returns a warning.
Default	0 (Ignore Errors)
GUI Tab	<a href="#">Advanced tab</a> on page 318

## Transaction Error Behavior

Attribute	TransactionErrorBehavior (TEB)
Description	Determines how the driver handles errors that occur within a transaction. When an error occurs in a transaction, the PostgreSQL server does not allow any operations on the connection except for rolling back the transaction.
Valid Values	0   1   2  If set to 0 (None), the driver does not roll back the transaction when an error occurs. The application must handle the error and roll back the transaction. Any operation on the statement other than a rollback results in an error.  If set to 1 (Rollback Transaction), the driver rolls back the transaction when an error occurs. In addition to the original error message, the driver posts an error message indicating that the transaction has been rolled back.  If set to 2 (Rollback Savepoint), the driver rolls back the transaction to the last savepoint when an error is detected. In manual commit mode, the driver automatically sets a savepoint after each statement issued. This value makes transaction behavior resemble that of most other database system types, but uses more resources on the database server and may incur a slight performance penalty.
Default	1 (Rollback Transaction)
GUI Tab	<a href="#">Advanced tab</a> on page 318

## TrustStore

Attribute	Truststore (TS)
Description	The directory that contains the truststore file and the truststore file name to be used when SSL is enabled (Encryption Method=1) and server authentication is used. The truststore file contains a list of the valid Certificate Authorities (CAs) that are trusted by the client machine for SSL server authentication. If you do not specify a directory, the current directory is used.  NOTE: The truststore and keystore files may be the same file.
Valid Values	<i>truststore_directory\filename</i>  where <i>truststore_directory</i> is the directory where the truststore file is located and <i>filename</i> is the file name of the truststore file.
Default	None
GUI Tab	<a href="#">Security tab</a> on page 320

## TrustStore Password

Attribute	TruststorePassword (TSP)
Description	The password that is used to access the truststore file when SSL is enabled (Encryption Method=1) and server authentication is used. The truststore file contains a list of the Certificate Authorities (CAs) that the client trusts.

NOTE: The truststore and keystore files may be the same file; therefore, they may have the same password.

Valid Values *truststore\_password*

where *truststore\_password* is a valid password for the truststore file.

Default None

GUI Tab [Security tab](#) on page 320

## User Name

Attribute LogonID (UID)

Description The default user ID that is used to connect to your database. Your ODBC application may override this value or you may override it in the logon dialog box or connection string.

Valid Values *userid*

where *userid* is a valid user ID with permissions to access the database.

Default None

GUI Tab [Security tab](#) on page 320

## Validate Server Certificate

Attribute ValidateServerCertificate (VSC)

Description Determines whether the driver validates the certificate that is sent by the database server when SSL encryption is enabled (Encryption Method=1). When using SSL server authentication, any certificate sent by the server must be issued by a trusted Certificate Authority (CA). Allowing the driver to trust any certificate returned from the server even if the issuer is not a trusted CA is useful in test environments because it eliminates the need to specify truststore information on each client in the test environment.

Truststore information is specified using the Trust Store and Trust Store Password options.

Valid Values 0 | 1

If set to 1 (Enabled), the driver validates the certificate that is sent by the database server. Any certificate from the server must be issued by a trusted CA in the truststore file. If the Host Name In Certificate option is specified, the driver also validates the certificate using a host name. The Host Name In Certificate option provides additional security against man-in-the-middle (MITM) attacks by ensuring that the server the driver is connecting to is the server that was requested.

If set to 0 (Disabled), the driver does not validate the certificate that is sent by the database server. The driver ignores any truststore information specified by the Trust Store and Trust Store Password options.

Default 1 (Enabled)

GUI Tab [Security tab](#) on page 320

## XML Describe Type

Attribute	XMLDescribeType (XDT)
Description	The SQL data type that is returned by SQLGetTypeInfo for the XML data type.  See <a href="#">"Using the XML Data Type" on page 346</a> for further information about the XML data type.
Valid Values	-4   -10  If set to -4 (SQL_LONGVARBINARY), the driver uses the description SQL_LONGVARBINARY for columns that are defined as the XML data type.  If set to -10 (SQL_WLONGVARCHAR), the driver uses the description SQL_WLONGVARCHAR for columns that are defined as the XML data type.
Default	-10
GUI Tab	<a href="#">Advanced tab</a> on page 318

---

## Performance Considerations

The following connection options can enhance driver performance. You can also enhance performance through efficient application design. Refer to [Chapter 5 "Designing ODBC Applications for Performance Optimization"](#) in the *DataDirect Connect Series for ODBC Reference* for details.

**Application Using Threads (ApplicationUsingThreads):** The driver coordinates concurrent database operations (operations from different threads) by acquiring locks. Although locking prevents errors in the driver, it also decreases performance. If your application does not make ODBC calls from different threads, the driver has no reason to coordinate operations. In this case, the ApplicationUsingThreads attribute should be disabled (set to 0).

**NOTE:** If you are using a multi-threaded application, you must enable the Application Using Threads option.

**Connection Pooling (ConnectionPooling):** If you enable the driver to use connection pooling, you can set additional options that affect performance:

- **Load Balance Timeout:** You can define how long to keep connections in the pool. The time that a connection was last used is compared to the current time and, if the timespan exceeds the value of the Load Balance Timeout option, the connection is destroyed. The Min Pool Size option can cause some connections to ignore this value.
- **Connection Reset:** Resetting a re-used connection to the initial configuration settings impacts performance negatively because the connection must issue additional commands to the server.



- **Max Pool Size:** Setting the maximum number of connections that the pool can contain too low might cause delays while waiting for a connection to become available. Setting the number too high wastes resources.
- **Min Pool Size:** A connection pool is created when the first connection with a unique connection string connects to the database. The pool is populated with connections up to the minimum pool size, if one has been specified. The connection pool retains this number of connections, even when some connections exceed their Load Balance Timeout value.

**Encryption Method (EncryptionMethod):** Data encryption may adversely affect performance because of the additional overhead (mainly CPU usage) that is required to encrypt and decrypt data.

**Failover Mode (FailoverMode):** Although high availability that replays queries after a failure provides increased levels of protection, it can adversely affect performance because of increased overhead.

---

## Data Types

Table 9-2 shows how the PostgreSQL data types are mapped to the standard ODBC data types. "Using the XML Data Type" on page 346 describes PostgreSQL to Unicode data type mappings.

---

**Table 9-2. PostgreSQL Data Types**

---

PostgreSQL	ODBC
Bigint	SQL_BIGINT
Bigserial	SQL_BIGINT
Bit <sup>1</sup>	SQL_BIT
Bit varying	SQL_VARBINARY
Boolean	SQL_BIT
Bytea	SQL_VARBINARY
Character	SQL_CHAR
Character varying	SQL_VARCHAR
Date	SQL_TYPE_DATE
Decimal	SQL_DECIMAL
Double Precision	SQL_DOUBLE
Integer	SQL_INTEGER
Money	SQL_DOUBLE
Name	SQL_VARCHAR
Numeric <sup>2</sup>	SQL_NUMERIC
Real	SQL_REAL
Serial	SQL_INTEGER

**Table 9-2. PostgreSQL Data Types** (cont.)

PostgreSQL	ODBC
Smallint	SQL_SMALLINT
Text	SQL_LONGVARCHAR
Time <sup>3</sup>	SQL_TYPE_TIME
Timestamp	SQL_TYPE_TIMESTAMP
Timestamp with timezone <sup>4</sup>	SQL_VARCHAR
XML	SQL_WLONGVARCHAR

1. Bit maps to SQL\_BIT when the length for the bit is 1. If the length is greater than 1, the driver maps the column to SQL\_BINARY.
2. Numeric maps to SQL\_NUMERIC if the precision of the Numeric is less than or equal to 38. If the precision is greater than 38, the driver maps the column to SQL\_VARCHAR.
3. Time mapping changes based on the setting of the Fetch TWFS as Time option
4. Timestamp with timezone mapping changes based on the setting of the Fetch TSWTZ as Timestamp option.

See ["Retrieving Data Type Information" on page 59](#) for more information about data types.

## Using the XML Data Type

By default, PostgreSQL returns XML data to the driver encoded as UTF-8. To avoid data loss, an application must bind XML data as SQL\_C\_WCHAR. The driver then returns the data as either UTF-8 or UTF-16, depending on platform and application settings. If the application binds XML data as SQL\_C\_CHAR, the driver converts it to the client character encoding, possibly causing data loss or corruption. To prevent any conversion of XML data, the application must set the option [XML Describe Type](#) to SQL\_LONGVARBINARY (-4) and bind the data as SQL\_C\_BINARY.

## Unicode Support

The PostgreSQL Wire Protocol driver automatically determines whether the PostgreSQL database is a Unicode database.

## Advanced Features

The driver supports the following advanced features:

- Failover
- Security
- Connection Pooling

## Failover

The driver supports failover and its related connection options. Failover connection options are located on the [Failover tab](#) of the driver Setup dialog box. See ["Using Failover" on page 65](#) for a general description of failover and its implementation.

## Security

The driver supports authentication and encryption. Security connection options are located on the [Security tab](#) of the driver Setup dialog box. See ["Using Security" on page 77](#) for a general description of security and its implementation.

## Connection Pooling

The driver supports connection pooling and its related connection options. Connection pooling connection options are located on the [Pooling tab](#) of the driver Setup dialog box. See ["Using DataDirect Connection Pooling" on page 82](#) for a general description of connection pooling and its implementation.

---

## Stored Procedure Results

PostgreSQL provides functionality to create user-defined functions. PostgreSQL does not make a distinction between user-defined functions and stored procedures. To PostgreSQL, everything is a user-defined function. PostgreSQL does not define a call mechanism for invoking a user-defined function. User-defined functions must be invoked via a SQL statement. For example, given a function defined as:

```
CREATE table foo (intcol int, varcharcol varchar(123))
CREATE or REPLACE FUNCTION insertFoo
  (IN idVal int, IN nameVal varchar) RETURNS void
  AS $$
    insert into foo values ($1, $2);
  $$
LANGUAGE SQL;
```

must be invoked natively as:

```
SELECT * FROM insertFoo(100, 'Mark')
```

even though the function does not return a value or results. The Select SQL statement returns a result set that has one column named insertFoo and no row data.

The PostgreSQL Wire Protocol driver supports invoking user-defined functions using the ODBC call Escape. The previously described function can be invoked using:

```
{call insertFoo(100, 'Mark')}
```

PostgreSQL functions return data from functions as a result set. If multiple output parameters are specified, the values for the output parameters are returned as columns in the result set. For example, the function defined as:

```
CREATE or REPLACE FUNCTION addValues(in v1 int, in v2 int)
  RETURNS int
  AS $$
    SELECT $1 + $2;
  $$
  LANGUAGE SQL;
```

returns a result set with a single column of type SQL\_INTEGER, whereas the function defined as:

```
CREATE or REPLACE FUNCTION selectFooRow2
  (IN idVal int, OUT id int, OUT name varchar)
  AS $$
    select intcol, varcharcol from foo where intcol = $1;
  $$
  LANGUAGE SQL
```

returns a result set that contains two columns, a SQL\_INTEGER id column and a SQL\_VARCHAR name column.

In addition, when calling PostgreSQL functions that contain output parameters, the native syntax requires that the output parameter values be omitted from the function call. This, in addition to output parameter values being returned as a result set, makes the PostgreSQL behavior of calling functions different from most other databases.

The PostgreSQL Wire Protocol driver provides a mechanism that makes the invoking of functions more consistent with how other databases behave. In particular, the PostgreSQL Wire Protocol driver allows parameter markers for output parameters to be specified in the function argument list when the Escape call is used. The driver allows buffers to be bound to these output parameters. When the function is executed, the output parameters are removed from the argument list sent to the server. The driver extracts the output parameter values from the result set returned by the server and updates the bound output parameter buffers with those values. For example, the function `selectFooRow2` described previously can be invoked as:

```
sql = L"{call selectFooRow2(?, ?, ?)}";
retVal = SQLPrepare(hPrepStmt, sql, SQL_NTS);
retVal = SQLBindParameter(
    hPrepStmt, 1, SQL_PARAM_INPUT, SQL_C_LONG,
    SQL_INTEGER, 0, 0, &idBuf, 0, &idInd);
retVal = SQLBindParameter(
    hPrepStmt, 2, SQL_PARAM_OUTPUT, SQL_C_LONG,
    SQL_INTEGER, 0, 0, &idBuf2, 4, &idInd2);
retVal = SQLBindParameter(
    hPrepStmt, 3, SQL_PARAM_OUTPUT, SQL_C_WCHAR,
    SQL_VARCHAR, 30, 0, &nameBuf, 123, &nameInd);
retVal = SQLExecute(hPrepStmt);
```

The values of the id and name output parameters are returned in the `idBuf2` and `nameBuf` buffers.

If output parameters are bound to a function call, the driver returns the output parameters in the bound buffers. An error is returned if the number of output parameters bound when the function is executed is less than the number of output parameters defined in the function. If no output parameters are bound to a function call, the driver returns the output parameters as a result set.

PostgreSQL can also return results from a function as a refcursor. There can be, at most, one refcursor per result; however, a function can return multiple results where each result is a refcursor. A connection option defines how the driver handles refcursors. See ["Fetch Ref Cursors" on page 332](#) for details about this option.

---

## Persisting a Result Set as an XML Data File

The driver allows you to persist a result as an XML data file with embedded schema. See ["Persisting a Result Set as an XML Data File" on page 60](#) for details about implementation.

---

## Isolation and Lock Levels Supported

PostgreSQL supports isolation level 0 (read uncommitted), level 1 (read committed), 2 (Repeatable read), and level 3 (serializable). PostgreSQL supports record-level locking.

Refer to [Chapter 7 "Locking and Isolation Levels"](#) in the *DataDirect Connect Series for ODBC Reference* for details.

---

## SQL Support

The driver supports the core SQL grammar.

---

## ODBC Conformance Level

The driver is Level 1 compliant, that is, it supports all ODBC Core and Level 1 functions.

In addition, the following functions are supported:

- SQLColumnPrivileges
- SQLDescribeParam (if EnableDescribeParam=1)
- SQLForeignKeys
- SQLTablePrivileges

Refer to [Chapter 2 “ODBC API and Scalar Functions”](#) in the *DataDirect Connect Series for ODBC Reference* for a list of the API functions supported by the PostgreSQL Wire Protocol driver.

---

## Number of Connections and Statements Supported

The PostgreSQL Wire Protocol driver supports multiple connections and multiple statements per connection.

---

## Using Arrays of Parameters

PostgreSQL supports returning a set of output parameters or return values, but no ODBC standard method exists for returning arrays of output parameters or return values. If the call Escape is used to invoke a function that returns a set of output parameters and buffers are bound for those output parameters, the PostgreSQL Wire Protocol driver places the first set of output parameters in the bound buffers. If no output parameters are bound for functions that return a set of results or output parameters, the driver returns a result set with a row for each set of output parameters.

# 10 The Progress OpenEdge® Wire Protocol Driver

The DataDirect Connect *for* ODBC Progress OpenEdge Wire Protocol driver (the Progress OpenEdge Wire Protocol driver) supports:

- Progress OpenEdge 11
- Progress OpenEdge 10.1.x, 10.2.x

The Progress OpenEdge Wire Protocol driver is supported in the Windows, UNIX, and Linux environments. See ["Environment-Specific Information" on page 45](#) for detailed information about the Windows, UNIX, and Linux environments supported by this driver.

See the README file shipped with your DataDirect product for the file name of the Progress OpenEdge Wire Protocol driver.

---

## Driver Requirements

There are no client requirements for the Progress OpenEdge Wire Protocol driver.

---

## Configuring and Connecting to Data Sources

After you install the driver, you configure data sources to connect to the database. See [Chapter 1 "Quick Start Connect" on page 35](#) for an explanation of different types of data sources. The data source contains connection options that allow you to tune the driver for specific performance. If you want to use a data source but need to change some of its values, you can either modify the data source or override its values at connection time through a connection string.

If you choose to use a connection string, you must use specific connection string attributes. See ["Using a Connection String" on page 357](#) and [Table 10-1 on page 359](#) for an alphabetical list of driver connection string attributes and their initial default values.



### Data Source Configuration in the UNIX odbc.ini File

On UNIX and Linux, data sources are configured and modified by editing the system information file (by default, odbc.ini) and storing default connection values there. See [Chapter 4 "Configuring the Product on UNIX/Linux" on page 97](#) for detailed information about the specific steps needed to set up the UNIX and Linux environments and to configure a data source.

Table 10-1 on page 359 lists driver connection string attributes that must be used in the `odbc.ini` file to set the value of connection options. Note that only the long name of the attribute can be used in the file. The default listed in the table is the initial default value when the driver is installed.

## Data Source Configuration through a GUI



On Windows, data sources are stored in the Windows Registry. You can configure and modify data sources through the ODBC Administrator using a driver Setup dialog box, as described in this section.

When the driver is first installed, the values of its connection options are set by default. These values appear on the driver Setup dialog box tabs when you create a new data source. You can change these default values by modifying the data source. In the following procedure, the description of each tab is followed by a table that lists the connection options for that tab and their initial default values. This table links you to a complete description of the options and their connection string attribute equivalents. The connection string attributes are used to override the default values of the data source if you want to change these values at connection time.

### To configure a Progress OpenEdge data source:

- 1 Start the ODBC Administrator. On Windows, start the ODBC Administrator by selecting its icon from the DataDirect Connect program group.
- 2 Select a tab:

- **User DSN:** If you are configuring an existing user data source, select the data source name and click **Configure** to display the driver Setup dialog box.

If you are configuring a new user data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.



- **System DSN:** If you are configuring an existing system data source, select the data source name and click **Configure** to display the driver Setup dialog box.

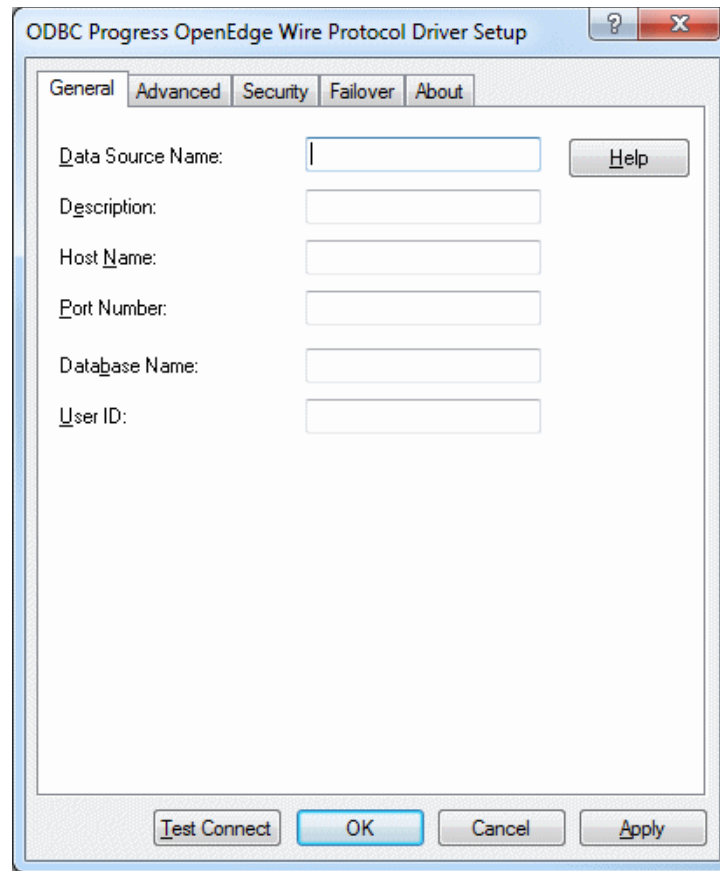
If you are configuring a new system data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.

- **File DSN:** If you are configuring an existing file data source, select the data source file and click **Configure** to display the driver Setup dialog box.

If you are configuring a new file data source, click **Add** to display a list of installed drivers; then, select a driver. Click **Advanced** if you want to specify attributes; otherwise, click **Next** to proceed. Specify a name for the data source and click **Next**. Verify the data source information; then, click **Finish** to display the driver Setup dialog box.



The General tab of the Setup dialog box appears by default.

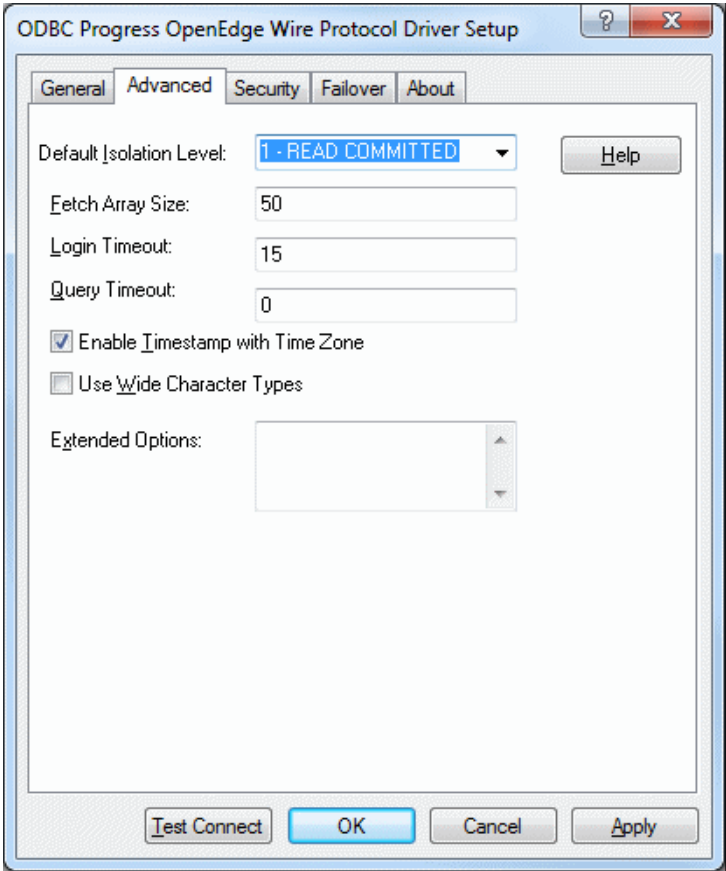


NOTE: The General tab displays only fields that are required for creating a data source. The fields on all other tabs are optional, unless noted otherwise.

- 3 Provide values for the options on this tab in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: General	Default
<a href="#">Data Source Name (see page 361)</a>	None
<a href="#">Description (see page 362)</a>	None
<a href="#">Host Name (see page 365)</a>	None
<a href="#">Port Number (see page 368)</a>	None
<a href="#">Database Name (see page 361)</a>	None
<a href="#">User ID (see page 369)</a>	None

- 4 Optionally, click the **Advanced** tab to specify additional data source settings.



On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Advanced	Default
<a href="#">Default Isolation Level (see page 361)</a>	1- READ COMMITTED
<a href="#">Fetch Array Size (see page 365)</a>	50
<a href="#">Login Timeout (see page 367)</a>	15
<a href="#">Query Timeout (see page 368)</a>	0
<a href="#">Enable Timestamp with Timezone (see page 362)</a>	Enabled
<a href="#">Use Wide Character Types (see page 369)</a>	Disabled
<a href="#">IANAAppCodePage (see page 366)</a>	4 (ISO 8559-1 Latin 1)
UNIX ONLY	

**Extended Options:** Type a semi-colon separated list of connection options and their values. Use this configuration option to set the value of undocumented connection options that are provided by Progress DataDirect customer support. You can include any valid connection option in the Extended Options string, for example:

```
Database=Server1;UndocumentedOption1=value[;UndocumentedOption2=value;]
```

If the Extended Options string contains option values that are also set in the setup dialog or data source, the values of the options specified in the Extended Options string

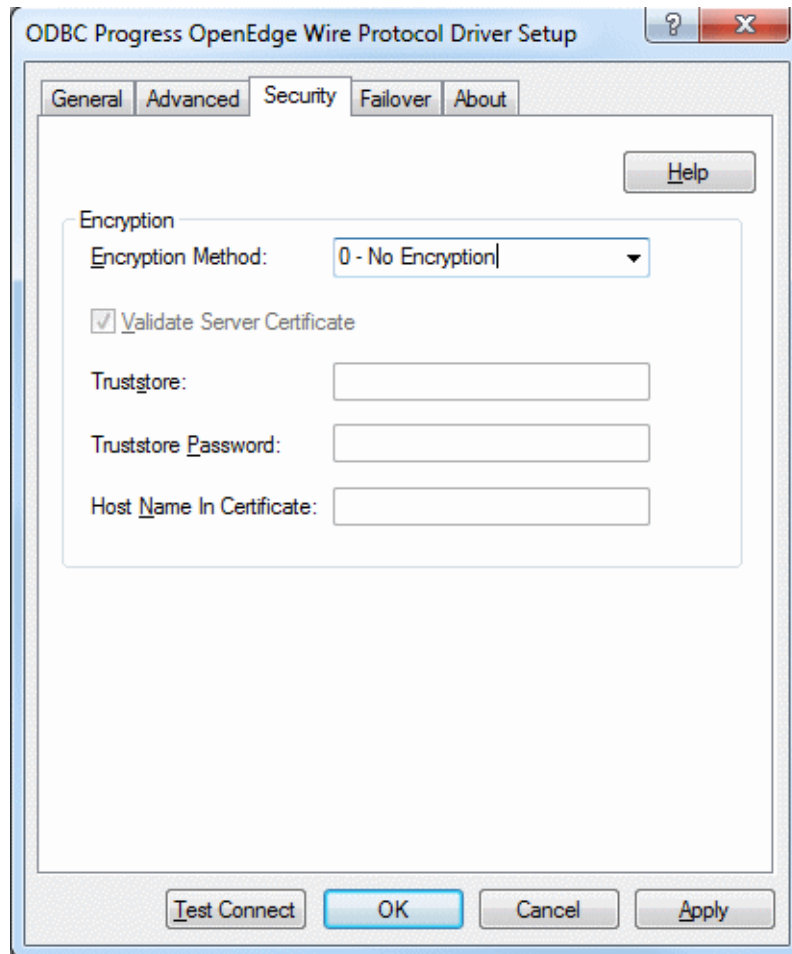
take precedence. However, connection options that are specified on a connection string override any option value specified in the Extended Options string.

**NOTE:** Do not specify the Extended Options configuration option in a connection string, or the driver will return an error. Instead, applications should specify the individual undocumented connection options in the connection string.

**Translate:** Click **Translate** to display the Select Translator dialog box, which lists the translators specified in the ODBC Translators section of the Registry. Progress 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.

- 5 Optionally, click the **Security** tab to specify security data source settings.

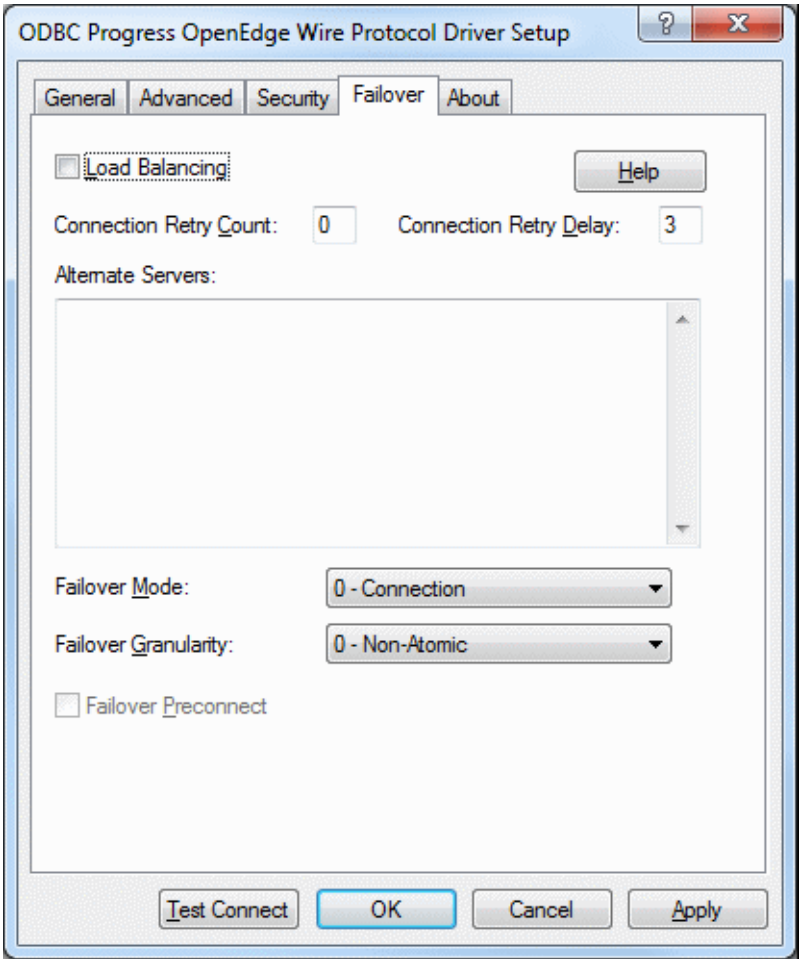


See [Chapter "Using Security" on page 77](#) for a general description of authentication and encryption and their configuration requirements.

On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Security	Default
<a href="#">Encryption Method (see page 363)</a>	0 (No Encryption)
<a href="#">Validate Server Certificate (see page 369)</a>	Enabled
<a href="#">Truststore (see page 368)</a>	None
<a href="#">Truststore Password (see page 369)</a>	None
<a href="#">Host Name In Certificate (see page 365)</a>	None

6 Optionally, click the **Failover** tab to specify failover data source settings.



See ["Using Failover" on page 65](#) for a general description of failover and its related connection options.

On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Failover	Default
<a href="#">Load Balancing (see page 366)</a>	Disabled
<a href="#">Connection Retry Count (see page 360)</a>	0
<a href="#">Connection Retry Delay (see page 361)</a>	3
<a href="#">Alternate Servers (see page 360)</a>	None

Connection Options: Failover	Default
Failover Mode (see page 364)	0 (Connection)
Failover Granularity (see page 363)	0 (Non-Atomic)
Failover Preconnect (see page 364)	Disabled

- 7 At any point during the configuration process, you can click **Test Connect** to attempt to connect to the data source using the connection options specified in the driver Setup dialog box. A logon dialog box appears (see ["Using a Logon Dialog Box" on page 358](#) for details). 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 incorrect environment or connection value, it displays an appropriate error message. Click **OK**.

NOTE: If you are configuring alternate servers for use with the connection failover feature, be aware that the Test Connect button tests only the primary server, not the alternate servers.

- 8 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.

## Using a Connection String

If you want to use a connection string for connecting to a database, or if your application requires it, you must specify either a DSN (data source name), a File DSN, or a DSN-less connection in the string. The difference is whether you use the `DSN=`, `FILEDSN=`, or the `DRIVER=` keyword in the connection string, as described in the ODBC specification. A DSN or FILEDSN connection string tells the driver where to find the default connection information. Optionally, you may specify *attribute=value* pairs in the connection string to override the default values stored in the data source.

The DSN connection string has the form:

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

The FILEDSN connection string has the form:

```
FILEDSN=filename.dsn[;attribute=value[;attribute=value]...]
```

The DSN-less connection string specifies a driver instead of a data source. All connection information must be entered in the connection string because the information is not stored in a data source.

The DSN-less connection string has the form:

```
DRIVER=[{]driver_name[;attribute=value[;attribute=value]...]
```

[Table 10-1 on page 359](#) lists the long and short names for each attribute, as well as the initial default value when the driver is first installed. You can specify either long or short names in the connection string.

An example of a DSN connection string with overriding attribute values for Progress OpenEdge is:

```
DSN=PROGRESS;DB=PAYROLL;UID=JOHN;PWD=XYZZY
```

A FILEDSN connection string is similar except for the initial keyword:

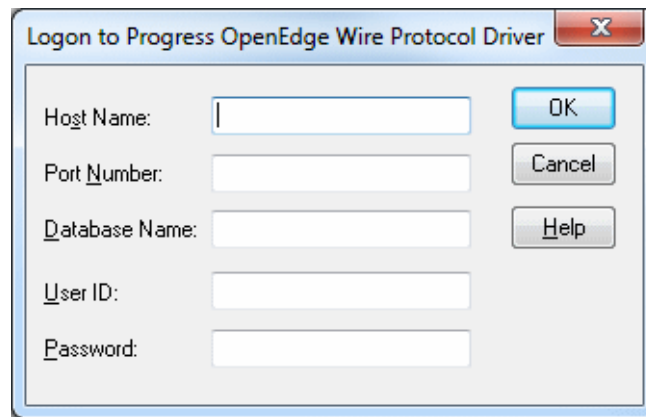
```
FILEDSN=ProgOpen.dsn;DB=PAYROLL;UID=JOHN;PWD=XYZZY
```

A DSN-less connection string must provide all necessary connection information:

```
DRIVER=DataDirect 7.1 Progress OpenEdge Wire Protocol;DB=PAYROLL;UID=JOHN;  
PWD=XYZZY;HOST=LOCALHOST;PORT=2055
```

## 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, provide the following information:

- 1 In the Host Name field, type the name of the system where the database is stored.
- 2 Type the Port Number setup for the database listener process.
- 3 Type the name of the database to which you want to connect.
- 4 Type your user name.
- 5 Type your password.
- 6 Click **OK** to complete the logon.

---

## Connection Option Descriptions

The following connection option descriptions are listed alphabetically by the GUI name that appears on the driver Setup dialog box. The connection string attribute name, along with its short name, is listed immediately underneath the GUI name. For example:

## Application Using Threads

Attribute ApplicationUsingThreads (AUT)

In most cases, the GUI name and the attribute name are the same; however, some exceptions exist. If you need to look up an option by its connection string attribute name, please refer to the alphabetical table of connection string attribute names.

Also, a few connection string attributes, for example, Password, do not have equivalent options that appear on the GUI. They are in the list of descriptions alphabetically by their attribute names.

Table 10-1 lists the connection string attributes supported by the Progress OpenEdge Wire Protocol driver.

**Table 10-1. Progress OpenEdge Wire Protocol Attribute Names**

Attribute (Short Name)	Default
AlternateServers (ASVR)	None
ArraySize (AS)	None
ConnectionRetryCount (CRC)	0
ConnectionRetryDelay (CRD)	3
Database (DB)	None
DataSourceName (DSN)	None
DefaultIsolationLevel (DIL)	1- READ COMMITTED
Description (n/a)	None
EnableTimestampwithTimezone (ETWT)	1 (enabled)
EncryptionMethod (EM)	0 (No Encryption)
FailoverGranularity (FG)	0 (Non-Atomic)
FailoverMode (FM)	0 (Connection)
FailoverPreconnect (FP)	0 (Disabled)
Fetch Array Size	50
HostName (HOST)	None
HostNameInCertificate (HNIC)	None
IANAAppCodePage (IACP) UNIX ONLY	4 (ISO 8559-1 Latin-1)
LoadBalancing (LB)	0 (Disabled)
LoginTimeout (LT)	15
LogonID (UID)	None
Password (PWD)	None
PortNumber (PORT)	None
QueryTimeout (QT)	0 (Disabled)
Truststore (TS)	None
TruststorePassword (TSP)	None
UseWideCharacterTypes (UWCT)	0 (disabled)
ValidateServerCertificate (VSC)	1 (enabled)

### Alternate Servers

Attribute	AlternateServers (ASVR)
Description	<p>A list of alternate database servers to which the driver tries to connect if the primary database server is unavailable. Specifying a value for this option enables connection failover for the driver. The value you specify must be in the form of a string that defines the physical location of each alternate server. All of the other required connection information for each alternate server is the same as what is defined for the primary server connection.</p>
Valid Values	<p>(HostName=hostvalue:PortNumber=portvalue:Database=databasevalue[, . . .])</p> <p>You must specify the host name, port number, and database name of each alternate server.</p> <p>NOTE: An alternate server address in IPv6 format must be enclosed in double quotation marks.</p>
Example	<p>The following Alternate Servers values define two alternate database servers for connection failover:</p> <pre>AlternateServers=(HostName=123.456.78.90:PortNumber=5177:Database=PAYROLL1, HostName=223.456.78.90:PortNumber=5178:Database=PAYROLL2)</pre>
Default	None
GUI Tab	<a href="#">Failover tab</a> on page 356

### Connection Retry Count

Attribute	ConnectionRetryCount (CRC)
Description	<p>The number of times the driver retries connection attempts to the primary database server, and if specified, alternate servers until a successful connection is established.</p> <p>This option and the Connection Retry Delay connection option, which specifies the wait interval between attempts, can be used in conjunction with failover.</p>
Valid Values	<p>0   <i>x</i></p> <p>where <i>x</i> is a positive integer from 1 to 65535.</p> <p>If set to 0, the driver does not try to connect after the initial unsuccessful attempt.</p> <p>If set to <i>x</i>, the driver retries connection attempts the specified number of times. If a connection is not established during the retry attempts, the driver returns an error that is generated by the last server to which it tried to connect.</p>
Default	0
GUI Tab	<a href="#">Failover tab</a> on page 356



## Connection Retry Delay

Attribute	ConnectionRetryDelay (CRD)
Description	<p>The number of seconds the driver waits between connection retry attempts when Connection Retry Count is set to a positive integer.</p> <p>This option and the Connection Retry Count connection option can be used in conjunction with failover.</p>
Valid Values	<p>0   <math>x</math></p> <p>where <math>x</math> is a positive integer from 1 to 65535.</p> <p>If set to 0, there is no delay between retries.</p> <p>If set to <math>x</math>, the driver waits the specified number of seconds between connection retry attempts.</p>
Default	3
GUI Tab	<a href="#">Failover tab</a> on page 356

## Data Source Name

Attribute	DataSourceName (DSN)
Description	The name of a data source in your Windows Registry or odbc.ini file.
Valid Values	<p><i>string</i></p> <p>where <i>string</i> is the name of a data source.</p>
Default	None
GUI Tab	<a href="#">General tab</a> on page 353

## Database Name

Attribute	Database (DB)
Description	The name of the database to which you want to connect.
Valid Values	<p><i>database_name</i></p> <p>where <i>database_name</i> is the name of a valid database.</p>
Default	None
GUI Tab	<a href="#">General tab</a> on page 353

## Default Isolation Level

Attribute	DefaultIsolationLevel (DIL)
Description	The method by which locks on data in the database are acquired and released.

Valid Values	0   1   2   3
	<p>If set to 0 (READ_UNCOMMITTED), other processes can be read from the database. Only modified data is locked and is not released until the transaction ends.</p> <p>If set to 1 (READ_COMMITTED) other processes can change a row that your application has read if the cursor is not on the row you want to change. This level prevents other processes from changing records that your application has changed until your application commits them or ends the transaction.</p> <p>If set to 2 (REPEATABLE_READ), other processes are prevented from accessing data that your application has read or modified. All read or modified data is locked until transaction ends.</p> <p>If set to 3 (SERIALIZABLE), other processes are prevented from changing records that are read or changed by your application (including phantom records) until your program commits them or ends the transaction. This level prevents the application from reading modified records that have not been committed by another process. If your application 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 application.</p> <p>Refer to <a href="#">Chapter 7 “Locking and Isolation Levels”</a> in the <i>DataDirect Connect Series for ODBC Reference</i> for details about ODBC isolation levels.</p>
Default	1 - READ COMMITTED
GUI Tab	<a href="#">Advanced tab</a> on page 354

Description

Attribute	Description (n/a)
Description	An optional long description of a data source. This description is not used as a runtime connection attribute, but does appear in the ODBC.INI section of the Registry and in the odbc.ini file.
Valid Values	<i>string</i>  where <i>string</i> is a description of a data source.
Default	None
GUI Tab	<a href="#">General tab</a> on page 353

Enable Timestamp with Timezone

Attribute	EnableTimestampwithTimezone (ETWT)
Description	Determines whether the driver exposes timestamps with timezones to the application.
Valid Values	0   1
	<p>If set to 1 (Enabled), the driver exposes timestamps with timezones to the application.</p> <p>If set to 0 (Disabled), timestamps with timezones are not exposed to the application.</p>

Default 1 (Enabled)  
 GUI Tab [Advanced tab](#) on page 354

## Encryption Method

Attribute EncryptionMethod (EM)

Description The method the driver uses to encrypt data sent between the driver and the database server. If the specified encryption method is not supported by the database server, the connection fails and the driver returns an error.

This connection option can affect performance. See ["Performance Considerations" on page 370](#) for details.

Valid Values 0 | 1

If set to 0 (No Encryption), data is not encrypted.

If set to 1 (SSL), data is encrypted using SSL. If the server supports protocol negotiation, the driver and server negotiate the use of TLS v1, SSL v3, or SSL v2 in that order.

Default 0 (No Encryption)

GUI Tab [Security tab](#) on page 355

## Failover Granularity

Attribute FailoverGranularity (FG)

Description Determines whether the driver fails the entire failover process or continues with the process if errors occur while trying to reestablish a lost connection.

This option applies only when Failover Mode is set to 1 (Extended Connection) or 2 (Select).

The Alternate Servers option specifies one or multiple alternate servers for failover and is required for all failover methods.

Valid Values 0 | 1 | 2 | 3

If set to 0 (Non-Atomic), the driver continues with the failover process and posts any errors on the statement on which they occur.

If set to 1 (Atomic) the driver fails the entire failover process if an error is generated as the result of anything other than executing and repositioning a Select statement. If an error is generated as a result of repositioning a result set to the last row position, the driver continues with the failover process, but generates a warning that the Select statement must be reissued.

If set to 2 (Atomic Including Repositioning), the driver fails the entire failover process if any error is generated as the result of restoring the state of the connection or the state of work in progress.

If set to 3 (Disable Integrity Check), the driver does not verify that the rows that were restored during the failover process match the original rows. This value applies only when Failover Mode is set to 2 (Select).

Default 0 (Non-Atomic)  
 GUI Tab [Failover tab](#) on page 356

## Failover Mode

Attribute FailoverMode (FM)  
 Description Specifies the type of failover method the driver uses.

The Alternate Servers option specifies one or multiple alternate servers for failover and is required for all failover methods.

This connection option can affect performance. See ["Performance Considerations" on page 370](#) for details.

Valid Values 0 | 1 | 2

If set to 0 (Connection), the driver provides failover protection for new connections only.

If set to 1 (Extended Connection), the driver provides failover protection for new and lost connections, but not any work in progress.

If set to 2 (Select), the driver provides failover protection for new and lost connections. In addition, it preserves the state of work performed by the last Select statement executed.

Default 0 (Connection)  
 GUI Tab [Failover tab](#) on page 356

## Failover Preconnect

Attribute FailoverPreconnect (FP)  
 Description Specifies whether the driver tries to connect to the primary and an alternate server at the same time.

This attribute applies only when Failover Mode is set to 1 (Extended Connection) or 2 (Select) and at least one alternate server is specified.

The Alternate Servers option specifies one or multiple alternate servers for failover and is required for all failover methods.

Valid Values 0 | 1

If set to 0 (Disabled), the driver tries to connect to an alternate server only when failover is caused by an unsuccessful connection attempt or a lost connection. This value provides the best performance, but your application typically experiences a short wait while the failover connection is attempted.

If set to 1 (Enabled), the driver tries to connect to the primary and an alternate server at the same time. This can be useful if your application is time-sensitive and cannot absorb the wait for the failover connection to succeed.

Default 0 (Disabled)  
 GUI Tab [Failover tab](#) on page 356

## Fetch Array Size

Attribute	ArraySize (AS)
Description	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 connection option can affect performance. See <a href="#">"Performance Considerations" on page 370</a> for details.
Valid Values	$x$  where $x$ is a positive integer specifying the number of bytes.
Default	50
GUI Tab	<a href="#">Advanced tab</a> on page 354

## Host Name

Attribute	HostName (HOST)
Description	The name or the IP address of the server to which you want to connect.
Valid Values	<i>server_name</i>   <i>IP_address</i>  where:  <i>server_name</i> is the name of the server to which you want to connect.  <i>IP_address</i> is the IP address of the server to which you want to connect.  The IP address can be specified in either IPv4 or IPv6 format, or a combination of the two. See <a href="#">"Using IP Addresses" on page 54</a> for details about these formats.
Default	None
GUI Tab	<a href="#">General tab</a> on page 353

## Host Name In Certificate

Attribute	HostNameInCertificate (HNIC)
Description	A host name for certificate validation when SSL encryption is enabled (Encryption Method=1) and validation is enabled (Validate Server Certificate=1). This option provides additional security against man-in-the-middle (MITM) attacks by ensuring that the server the driver is connecting to is the server that was requested.
Valid Values	<i>host_name</i>   #SERVERNAME#  where the <i>host_name</i> is the host name specified in the certificate. Consult your SSL administrator for the correct value.  If set to a host name, the driver examines the subjectAltName values included in the certificate. If a dnsName value is present in the subjectAltName values, then the driver compares the value specified for Host Name In Certificate with the dnsName value. The connection succeeds if the values match. The connection fails if the Host Name In Certificate value does not match the dnsName value.

If no subjectAltName values exist or a dnsName value is not in the list of subjectAltName values, then the driver compares the value specified for Host Name In Certificate with the commonName part of the Subject name in the certificate. The commonName typically contains the host name of the machine for which the certificate was created. The connection succeeds if the values match. The connection fails if the Host Name In Certificate value does not match the commonName. If multiple commonName parts exist in the Subject name of the certificate, the connection succeeds if the Host Name In Certificate value matches any of the commonName parts.

If set to #SERVERNAME#, the driver compares the host server name specified as part of a data source or connection string to the dnsName or the commonName value.

Default      None  
GUI Tab     [Security tab](#) on page 355



## IANAAppCodePage

Attribute    IANAAppCodePage (IACP)  
Description    An Internet Assigned Numbers Authority (IANA) value. You must specify a value for this option if your application is not Unicode-enabled or if your database character set is not Unicode. Refer to [Chapter 4 “Internationalization, Localization, and Unicode”](#) in the *DataDirect Connect Series for ODBC Reference* for details.

The driver uses the specified IANA code page to convert "W" (wide) functions to ANSI.

The driver and Driver Manager both check for the value of IANAAppCodePage in the following order:

- In the connection string
- In the Data Source section of the system information file (odbc.ini)
- In the ODBC section of the system information file (odbc.ini)

If the driver does not find an IANAAppCodePage value, the driver uses the default value of 4 (ISO 8859-1 Latin-1).

Valid Values    *IANA\_code\_page*

where *IANA\_code\_page* is one of the valid values listed in [Chapter 1 “Values for the Attribute IANAAppCodePage”](#) in the *DataDirect Connect Series for ODBC Reference*. The value must match the database character encoding and the system locale.

Default      4 (ISO 8559-1 Latin-1)  
GUI Tab     N/A

## Load Balancing

Attribute    LoadBalancing (LB)  
Description    Determines whether the driver uses client load balancing in its attempts to connect to the database servers (primary and alternate). You can specify one or multiple alternate servers by setting the Alternate Servers option.

Valid Values 0 | 1

If set to 1 (Enabled), the driver uses client load balancing and attempts to connect to the database servers (primary and alternate servers) in random order.

If set to 0 (Disabled), the driver does not use client load balancing and connects to each server based on their sequential order (primary server first, then, alternate servers in the order they are specified).

NOTE: This option has no effect unless alternate servers are defined for the Alternate Servers connection option.

Default 0 (Disabled)

GUI Tab [Failover tab](#) on page 357

## Login Timeout

Attribute LoginTimeout (LT)

Description The number of seconds the driver waits for a connection to be established before returning control to the application and generating a timeout error. To override the value that is set by this connection option for an individual connection, set a different value in the SQL\_ATTR\_LOGIN\_TIMEOUT connection attribute using the SQLSetConnectAttr() function.

Valid Values -1 | 0 | *x*

where *x* is a positive integer that specifies a number of seconds.

If set to -1, the connection request does not time out. The driver silently ignores the SQL\_ATTR\_LOGIN\_TIMEOUT attribute.

If set to 0, the connection request does not time out, but the driver responds to the SQL\_ATTR\_LOGIN\_TIMEOUT attribute.

If set to *x*, the connection request times out after the specified number of seconds unless the application overrides this setting with the SQL\_ATTR\_LOGIN\_TIMEOUT attribute.

Default 15

GUI Tab [Advanced tab](#) on page 354

## Password

Attribute Password (PWD)

Description The password that the application uses to connect to your database. The Password option cannot be specified through the driver Setup dialog box and should not be stored in a data source. It is specified through the Logon dialog box or a connection string.

Valid Values *pwd*

where *pwd* is a valid password.

Default None

GUI Tab n/a

Port Number

Attribute	PortNumber (PORT)
Description	The port number of the server listener.
Valid Values	<i>port_name</i>  where the <i>port_name</i> is the port number of the server listener. Check with your database administrator for the correct number.
Default	None
GUI Tab	<a href="#">General tab</a> on page 353

Query Timeout

Attribute	QueryTimeout (QT)
Description	The number of seconds for the default query timeout for all statements that are created by a connection. To override the value set by this connection option for an individual statement, set a different value in the SQL_ATTR_QUERY_TIMEOUT statement attribute on the SQLSetStmtAttr() function.
Valid Values	-1   0   <i>x</i>  where <i>x</i> is a positive integer that specifies a number of seconds.  If set to -1, the query does not time out. The driver silently ignores the SQL_ATTR_QUERY_TIMEOUT attribute.  If set to 0, the query does not time out, but the driver responds to the SQL_ATTR_QUERY_TIMEOUT attribute.  If set to <i>x</i> , all queries time out after the specified number of seconds unless the application overrides this value by setting the SQL_ATTR_QUERY_TIMEOUT attribute.
Default	0
GUI Tab	<a href="#">Advanced tab</a> on page 354

Truststore

Attribute	Truststore (TS)
Description	The directory that contains the truststore file and the truststore file name to be used when SSL is enabled (Encryption Method=1) and server authentication is used. The truststore file contains a list of the valid Certificate Authorities (CAs) that are trusted by the client machine for SSL server authentication. If you do not specify a directory, the current directory is used.  NOTE: The truststore and keystore files may be the same file.
Valid Values	<i>truststore_directory\filename</i>  where <i>truststore_directory</i> is the directory where the truststore file is located and <i>filename</i> is the file name of the truststore file.
Default	None
GUI Tab	<a href="#">Security tab</a> on page 355



## Truststore Password

Attribute	TruststorePassword (TSP)
Description	The password that is used to access the truststore file when SSL is enabled (Encryption Method=1) and server authentication is used. The truststore file contains a list of the Certificate Authorities (CAs) that the client trusts.  NOTE: The truststore and keystore files may be the same file; therefore, they may have the same password.
Valid Values	<i>truststore_password</i>  where <i>truststore_password</i> is a valid password for the truststore file.
Default	None
GUI Tab	<a href="#">Security tab</a> on page 355

## Use Wide Character Types

Attribute	UseWideCharacterTypes (UWCT)
Description	A value that determines whether character data types are described to the application as SQL_CHAR or SQL_WCHAR when connected to a Unicode database.
Valid Values	0   1  If set to 0 (Disabled), character data types are described to the application as SQL_CHAR.  If set to 1 (Enabled), character data types are described to the application as SQL_WCHAR.
Default	0 (Disabled)
GUI Tab	<a href="#">Advanced tab</a> on page 354

## User ID

Attribute	LogonID (UID)
Description	The default user ID that is used to connect to your database. Your ODBC application may override this value or you may override it in the logon dialog box or connection string.
Valid Values	<i>userid</i>  where <i>userid</i> is a valid user ID with permissions to access the database.
Default	None
GUI Tab	<a href="#">General tab</a> on page 353

## Validate Server Certificate

Attribute	ValidateServerCertificate (VSC)
Description	Determines whether the driver validates the certificate that is sent by the database server when SSL encryption is enabled (Encryption Method=1). When using SSL server authentication, any certificate sent by the server must be issued by a trusted Certificate

Authority (CA). Allowing the driver to trust any certificate returned from the server even if the issuer is not a trusted CA is useful in test environments because it eliminates the need to specify truststore information on each client in the test environment.

Truststore information is specified using the Trust Store and Trust Store Password options.

Valid Values 0 | 1

If set to 1 (Enabled), the driver validates the certificate that is sent by the database server. Any certificate from the server must be issued by a trusted CA in the truststore file. If the Host Name In Certificate option is specified, the driver also validates the certificate using a host name. The Host Name In Certificate option provides additional security against man-in-the-middle (MITM) attacks by ensuring that the server the driver is connecting to is the server that was requested.

If set to 0 (Disabled), the driver does not validate the certificate that is sent by the database server. The driver ignores any truststore information specified by the Trust Store and Trust Store Password options.

Default 1 (Enabled)

GUI Tab [Security tab](#) on page 355

---

# Performance Considerations

The following connection options can enhance driver performance. You can also enhance performance through efficient application design. Refer to the *DataDirect Connect Series for ODBC Reference* for details.

**Encryption Method (EncryptionMethod):** Data encryption may adversely affect performance because of the additional overhead (mainly CPU usage) that is required to encrypt and decrypt data.

**Failover Mode (FailoverMode):** Although high availability that replays queries after a failure provides increased levels of protection, it can adversely affect performance because of increased overhead.

**Fetch Array Size (ArraySize):** Reducing the number of round trips on the network to the approximate number of rows being fetched increases performance. For example, if your application normally fetches 200 rows, it is more efficient for the driver to fetch 200 rows at one time over the network than to fetch 50 rows at a time during four round trips over the network.

NOTE: The ideal setting for your application will vary. To calculate the ideal setting for this option, you must know the size in bytes of the rows that you are fetching and the size in bytes of your Network Packet. Then, you must calculate the number of rows that will fit in your Network Packet, leaving space for packet overhead. For example, suppose your Network Packet size is 1024 bytes and the row size is 8 bytes. Dividing 1024 by 8 equals 128; however, the ideal setting for Fetch Array Size is 127, not 128, because the number of rows times the row size must be slightly smaller than the Network Packet size.

## Data Types

[Table 10-2](#) shows how the Progress OpenEdge data types are mapped to the standard ODBC data types.

**Table 10-2. Progress OpenEdge Data Types**

OpenEdge	ODBC
Bigint	SQL_BIGINT
Binary	SQL_BINARY
Bit	SQL_BIT
Blob	SQL_LONGVARBINARY
Char	SQL_CHAR
Clob	SQL_LONGVARCHAR
Date	SQL_TYPE_DATE
Decimal	SQL_DECIMAL
Double precision	SQL_DOUBLE
Float	SQL_FLOAT
Integer	SQL_INTEGER
Lvarbinary	SQL_LONGVARBINARY
Lvarchar	SQL_LONGVARCHAR
Numeric	SQL_NUMERIC
Real	SQL_FLOAT
Smallint	SQL_SMALLINT
Time	SQL_TYPE_TIME
Timestamp	SQL_TYPE_TIMESTAMP
Timestamp with Time Zone	SQL_CHAR
Tinyint	SQL_TINYINT
Varbinary	SQL_VARBINARY
Varchar	SQL_VARCHAR

See ["Retrieving Data Type Information" on page 59](#) for more information about data types.

## Unicode Support

When connected to a Unicode database, the Progress OpenEdge Wire Protocol driver supports the Unicode data types listed in the following table, in addition to standard ODBC data types listed in [Table 10-2 on page 371](#). The Use Wide Character Types connection string option must be enabled.

Progress OpenEdge Data Type	Mapped to. . .
Char	SQL_WCHAR
Varchar	SQL_WVARCHAR

The driver supports the Unicode ODBC W (Wide) function calls, such as SQLConnectW. This allows the Driver Manager to transmit these calls directly to the driver. Otherwise, the Driver Manager would incur the additional overhead of converting the W calls to ANSI function calls, and vice versa.

See ["UTF-16 Applications on UNIX and Linux" on page 125](#) for related details. Also, refer to [Chapter 4 "Internationalization, Localization, and Unicode"](#) in the *DataDirect Connect Series for ODBC Reference* for a more detailed explanation of Unicode.

---

## Advanced Features

The driver supports the following advanced features:

- Failover
- Security

### Failover

The driver supports failover and its related connection options. Failover connection options are located on the [Failover tab](#) of the driver Setup dialog box. See ["Using Failover" on page 65](#) for a general description of failover and its implementation.

### Security

The driver supports data security by providing SSL data encryption. See ["Using Security" on page 77](#) for a general description of encryption and its configuration requirements.

You configure the driver for data security on the [Security tab](#) of the driver Setup dialog box. See the description of the Security tab under ["Configuring and Connecting to Data Sources" on page 351](#) for specific implementations.

---

## Isolation and Lock Levels Supported

Progress OpenEdge supports isolation level 0 (read uncommitted), isolation level 1 (read committed), isolation level 2 (repeatable read), and isolation level 3 (serializable).

See [Chapter 7 "Locking and Isolation Levels"](#) in the *DataDirect Connect Series for ODBC Reference* for details.

---

## SQL Grammar Support

The driver supports the core SQL grammar.

---

## ODBC Conformance Level

The driver is Level 1 compliant, that is, it supports all ODBC Core and Level 1 functions.

The driver also supports the function `SQLSetPos`.

See [Chapter 2 “ODBC API and Scalar Functions”](#) in the *DataDirect Connect Series for ODBC Reference* for a list of the API functions supported by the Progress OpenEdge Wire Protocol driver.

---

## Number of Connections and Statements Supported

The Progress OpenEdge database system supports multiple connections and multiple statements per connection.



# 11 The SQL Server Wire Protocol Driver

The DataDirect Connect *for* ODBC and DataDirect Connect64 *for* ODBC SQL Server Wire Protocol driver (the SQL Server Wire Protocol driver) each support the following database versions:

## Cloud:

- Microsoft Windows Azure SQL Database

Note: For the versions that are supported by the SQL Server driver, refer to the product matrix on the Progress DataDirect Web site:

<http://www.datadirect.com/products/odbc/connectodbcmatrix/index.html>

## On premise:

- Microsoft SQL Server 2012
- Microsoft SQL Server 2008 R1, R2
- Microsoft SQL Server 2005
- Microsoft SQL Server 2000 Desktop Engine (MSDE 2000)
- Microsoft SQL Server 2000 Enterprise Edition (64-bit)

The SQL Server Wire Protocol driver is supported in the Windows, UNIX, and Linux environments. See "[Environment-Specific Information](#)" on [page 45](#) for detailed information about the environments supported by this driver.

Refer to the readme file shipped with your DataDirect product for the file name of the SQL Server Wire Protocol driver.

---

## Driver Requirements

The driver has no client requirements.

The SQL Server Wire Protocol driver connects via TCP/IP. TCP/IP connections must be configured on the Windows server on which the Microsoft SQL Server database resides.

---

## Configuring and Connecting to Data Sources

After you install the driver, you configure data sources to connect to the database. See [Chapter 1 "Quick Start Connect" on page 35](#) for an explanation of different types of data sources. The data source contains connection options that allow you to tune the driver for specific performance. If you want to use a data source but need to change some of its values, you can either modify the data source or override its values at connection time through a connection string.

If you choose to use a connection string, you must use specific connection string attributes. See ["Using a Connection String" on page 388](#) and [Table 11-1 on page 390](#) for an alphabetical list of driver connection string attributes and their initial default values.

## Data Source Configuration in the UNIX/Linux odbc.ini File

On UNIX and Linux, you must set up the proper ODBC environment before configuring data sources. See ["Environment Configuration" on page 37](#) for basic setup information and ["Environment Variables" on page 97](#) for more detail about this procedure.

Data sources for UNIX and Linux are stored in the system information file (by default, odbc.ini). You can configure and modify data sources directly by editing the odbc.ini file and storing default connection values there. See ["Configuration Through the System Information \(odbc.ini\) File" on page 103](#) for detailed information about the specific steps necessary to configure a data source.

[Table 11-1 on page 390](#) lists driver connection string attributes that must be used in the odbc.ini file to set the value of the attributes. Note that only the long name of the attribute can be used in the file. The default listed in the table is the initial default value when the driver is installed.

## Data Source Configuration through a GUI



On Windows, data sources are stored in the Windows Registry. You can configure and modify data sources through the ODBC Administrator using a driver Setup dialog box, as described in this section.



On UNIX and Linux, data sources are stored in the odbc.ini file.

When the driver is first installed, the values of its connection options are set by default. These values appear on the driver Setup dialog box tabs when you create a new data source. You can change these default values by modifying the data source. In the following procedure, the description of each tab is followed by a table that lists the connection options for that tab and their initial default values. This table links you to a complete description of the options and their connection string attribute equivalents. The connection string attributes are used to override the default values of the data source if you want to change these values at connection time.

### To configure a Microsoft SQL Server data source:

- 1 Start the ODBC Administrator by selecting its icon from the DataDirect Connect program group.
- 2 Select a tab:

- **User DSN:** If you are configuring an existing user data source, select the data source name and click **Configure** to display the driver Setup dialog box.

If you are configuring a new user data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.



- **System DSN:** If you are configuring an existing system data source, select the data source name and click **Configure** to display the driver Setup dialog box.



If you are configuring a new system data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.

- **File DSN:** If you are configuring an existing file data source, select the data source file and click **Configure** to display the driver Setup dialog box.

If you are configuring a new file data source, click **Add** to display a list of installed drivers; then, select a driver. Click **Advanced** if you want to specify attributes; otherwise, click **Next** to proceed. Specify a name for the data source and click **Next**. Verify the data source information; then, click **Finish** to display the driver Setup dialog box.

The General tab of the Setup dialog box appears by default.

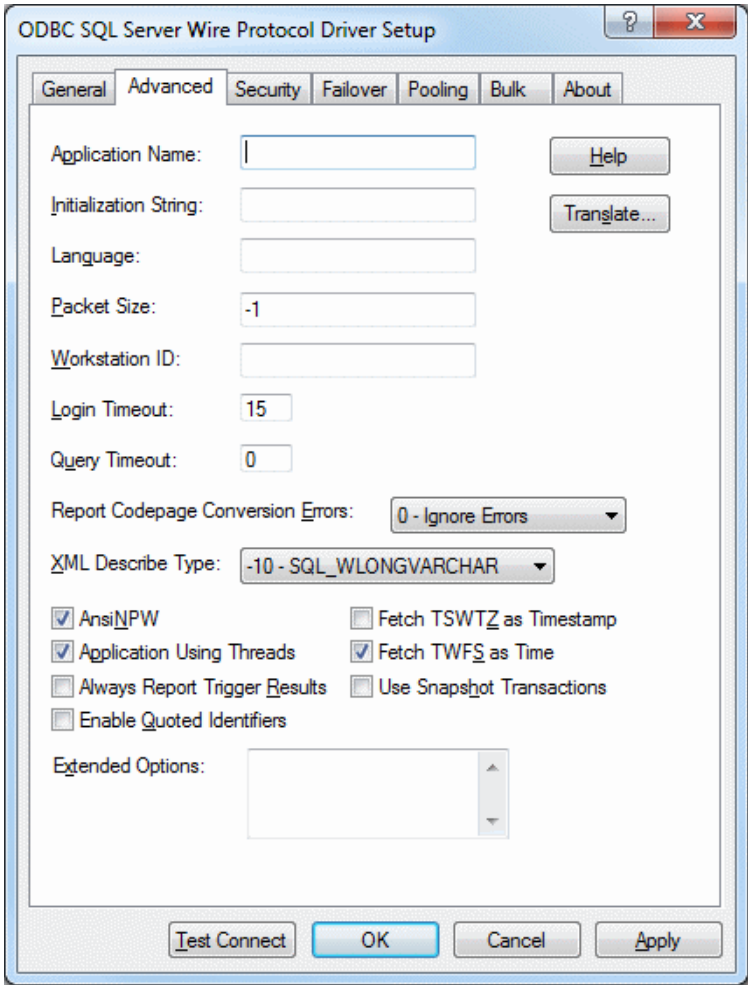
The screenshot shows the 'ODBC SQL Server Wire Protocol Driver Setup' dialog box with the 'General' tab selected. The dialog has a title bar with a question mark and a close button. Below the title bar are tabs for 'General', 'Advanced', 'Security', 'Failover', 'Pooling', 'Bulk', and 'About'. The 'General' tab contains the following fields: 'Data Source Name' (empty), 'Description' (empty), 'Host Name' (empty), 'Port Number' (1433), and 'Database' (empty). A 'Help' button is located to the right of the 'Data Source Name' field. At the bottom of the dialog are four buttons: 'Test Connect', 'OK', 'Cancel', and 'Apply'.

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 this tab, provide values for the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: General	Default
<a href="#">Data Source Name (see page 398)</a>	None
<a href="#">Description (see page 399)</a>	None
<a href="#">Host Name (see page 404)</a>	None
<a href="#">Port Number (see page 409)</a>	1433
<a href="#">Database (see page 398)</a>	None

- 4 Optionally, click the **Advanced** tab to specify additional data source settings.



On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Advanced	Default
<a href="#">Application Name (see page 393)</a>	None
<a href="#">Initialization String (see page 405)</a>	None
<a href="#">Language (see page 406)</a>	None
<a href="#">Packet Size (see page 408)</a>	-1
<a href="#">Workstation ID (see page 413)</a>	None

Connection Options: Advanced	Default
Login Timeout (see page 407)	15
Query Timeout (see page 409)	0
Report Codepage Conversion Errors (see page 410)	0 (Ignore Errors)
XML Describe Type (see page 413)	-10
AnsiNPW (see page 393)	Enabled
Application Using Threads (see page 394)	Enabled
Always Report Trigger Results (see page 392)	0 (Disabled)
Enable Quoted Identifiers (see page 399)	Disabled
Fetch TWFS as Time (see page 402)	Disabled
Fetch TSWTZ as Timestamp (see page 402)	Disabled
Use Snapshot Transactions (see page 411)	Disabled
IANAAppCodePage (IACP) (see page 405)	4 (ISO 8559-1 Latin-1)
UNIX ONLY	



**Extended Options:** Type a semi-colon separated list of connection options and their values. Use this configuration option to set the value of undocumented connection options that are provided by Progress DataDirect customer support. You can include any valid connection option in the Extended Options string, for example:

```
Database=Server1;UndocumentedOption1=value[;UndocumentedOption2=value;]
```

If the Extended Options string contains option values that are also set in the setup dialog or data source, the values of the options specified in the Extended Options string take precedence. However, connection options that are specified on a connection string override any option value specified in the Extended Options string.

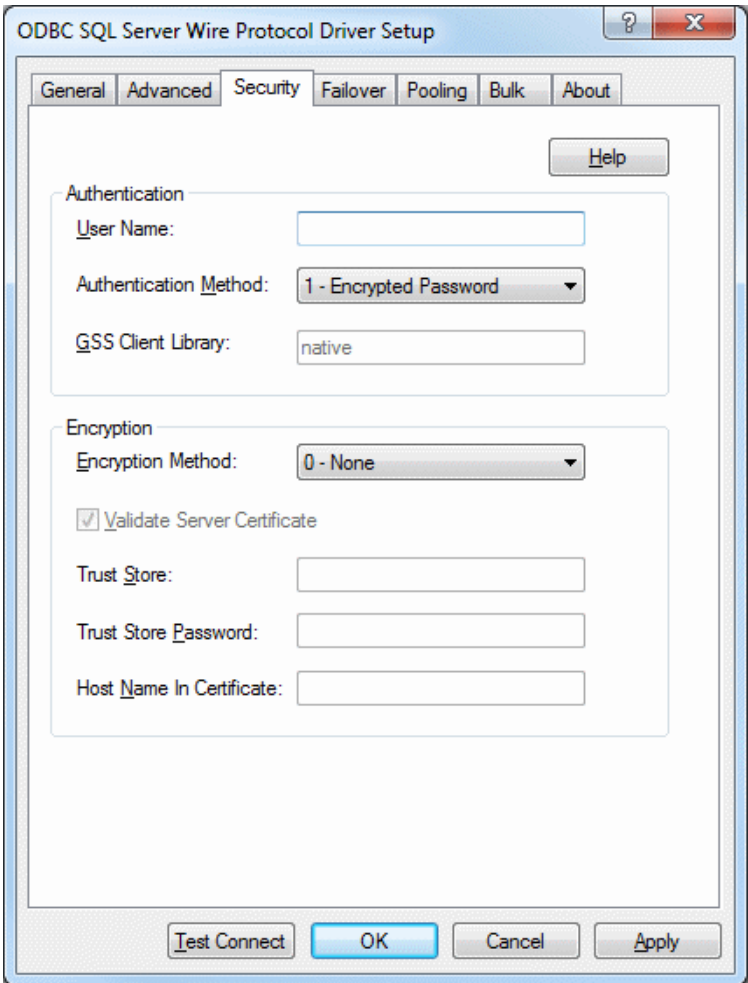
**NOTE:** Do not specify the Extended Options configuration option in a connection string, or the driver will return an error. Instead, applications should specify the individual undocumented connection options in the connection string.



**Translate:** Click **Translate** to display the Select Translator dialog box, which lists the translators specified in the ODBC Translators section of the Registry. Progress 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.

Optionally, click the **Security** tab to specify additional data source settings.

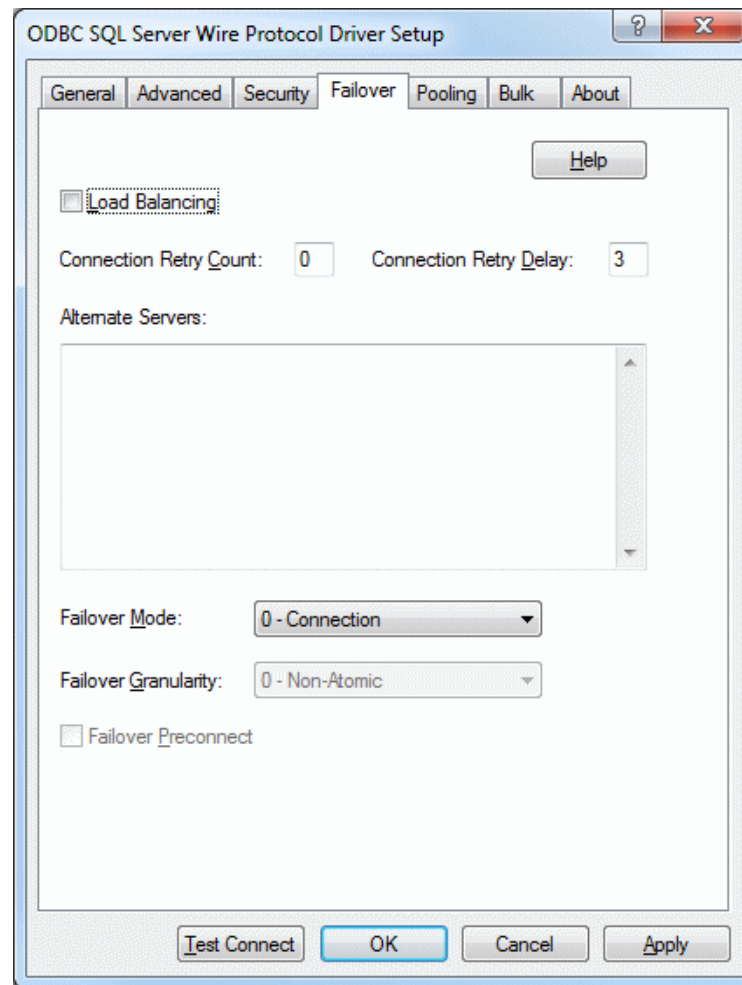


See ["Using Security" on page 77](#) for a general description of authentication and encryption and their configuration requirements.

On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Security	Default
<a href="#">User Name (see page 412)</a>	None
<a href="#">Authentication Method (see page 394)</a>	1 (Encrypt Password)
<a href="#">GSS Client Library (see page 403)</a>	native
<a href="#">Encryption Method (see page 400)</a>	0 (No Encryption)
<a href="#">Validate Server Certificate (see page 412)</a>	Enabled
<a href="#">Truststore (see page 411)</a>	None
<a href="#">Truststore Password (see page 411)</a>	None
<a href="#">Host Name In Certificate (see page 404)</a>	None

- 5 Optionally, click the **Failover** tab to specify additional data source settings.

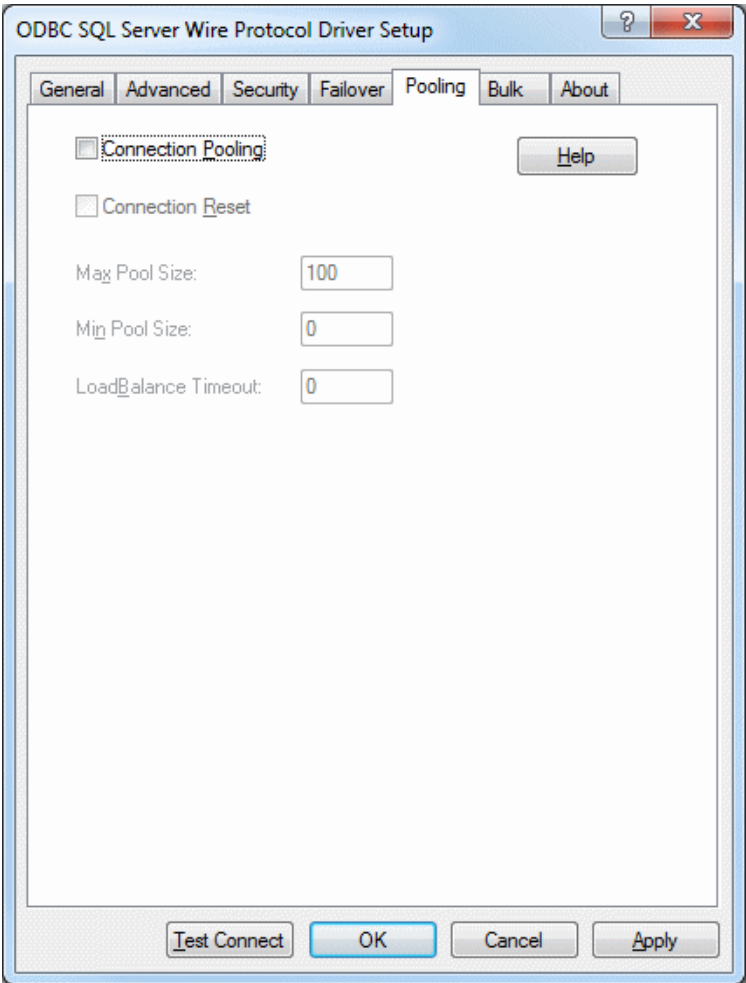


See ["Using Failover" on page 65](#) for a general description of failover and its related connection options.

On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Failover	Default
<a href="#">Load Balancing (see page 406)</a>	Disabled
<a href="#">Connection Retry Count (see page 397)</a>	0
<a href="#">Connection Retry Delay (see page 398)</a>	3
<a href="#">Alternate Servers (see page 392)</a>	None
<a href="#">Failover Mode (see page 401)</a>	0 (Connection)
<a href="#">Failover Granularity (see page 400)</a>	0 (Non-Atomic)
<a href="#">Failover Preconnect (see page 402)</a>	Disabled

6 Optionally, click the **Pooling** tab to specify additional data source settings.

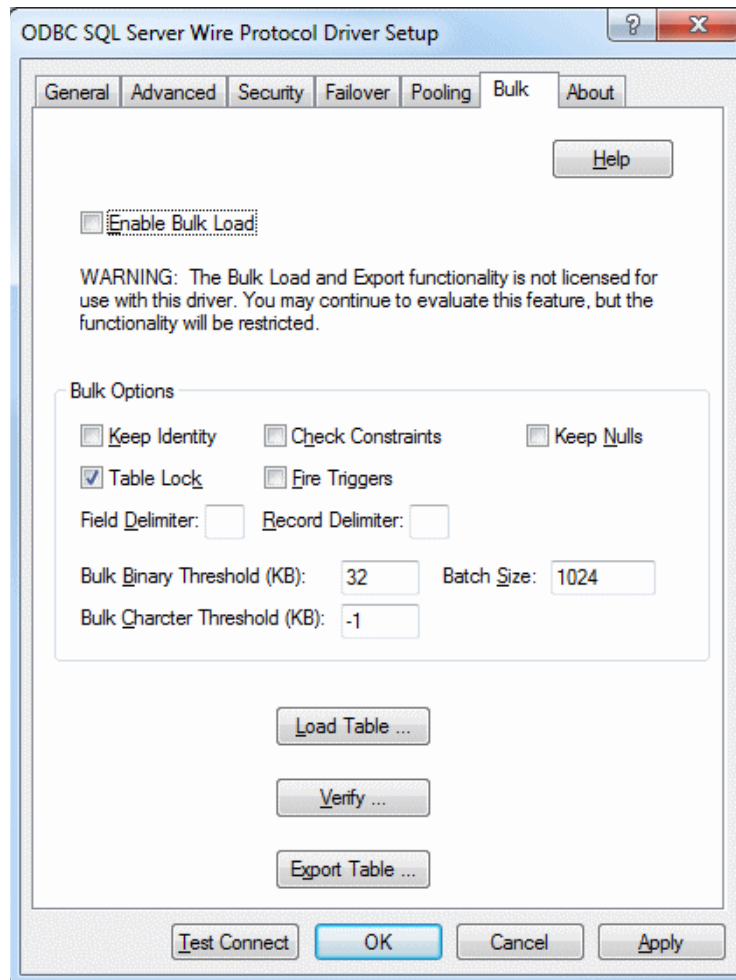


See ["Using DataDirect Connection Pooling" on page 82](#) for a general description of connection pooling.

On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Pooling	Default
<a href="#">Connection Pooling (see page 396)</a>	Disabled
<a href="#">Connection Reset (see page 397)</a>	Disabled
<a href="#">Max Pool Size (see page 407)</a>	100
<a href="#">Min Pool Size (see page 408)</a>	0
<a href="#">Load Balance Timeout (see page 406)</a>	0

- 7 Optionally, click the **Bulk** tab to specify additional data source settings.



See ["Using DataDirect Bulk Load" on page 85](#) for a general description of DataDirect Bulk Load.

On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

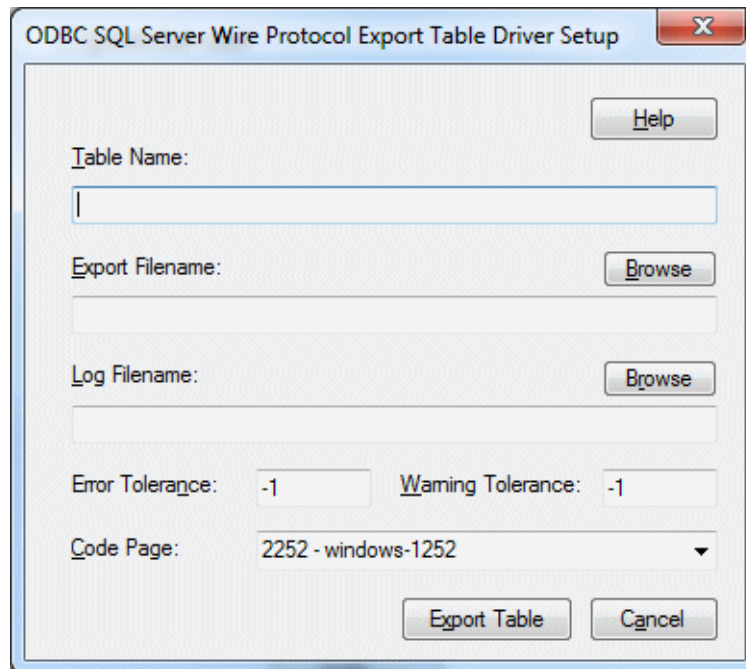
Connection Options: Bulk	Default
<a href="#">Enable Bulk Load (see page 399)</a>	Disabled
<a href="#">Bulk Options (see page 396)</a> Individual item descriptions are grouped under the Bulk Option description.	Only Table Lock enabled
<a href="#">Field Delimiter (see page 403)</a>	None
<a href="#">Record Delimiter (see page 410)</a>	None
<a href="#">Bulk Binary Threshold (see page 395)</a>	32
<a href="#">Bulk Character Threshold (see page 395)</a>	-1
<a href="#">Batch Size (see page 394)</a>	1024

If your application is already coded to use parameter array batch functionality, you can leverage DataDirect Bulk Load features through the Enable Bulk Load connection

option. Enabling this option automatically converts the parameter array batch operation to use the database bulk load protocol.

If you are not using parameter array batch functionality, you can export data to a bulk load data file, verify the metadata of the bulk load configuration file against the structure of the target table, and bulk load data to a table. Use the following steps to accomplish these tasks.

- a To export data from a table to a bulk load data file, click **Export Table** from the Bulk tab. The Export Table dialog box appears.



Both a bulk data file and a bulk configuration file are produced by exporting a table. The configuration file has the same name as the data file, but with an XML extension. See ["Using DataDirect Bulk Load" on page 85](#) for details about these files.

The bulk export operation can create a log file and can also export to external files. See ["External Overflow Files" on page 95](#) for more information. The export operation can be configured such that if any errors or warnings occur:

- The operation always completes.
- The operation always terminates.
- The operation terminates after a certain threshold of warnings or errors is exceeded.

**Table Name:** A string that specifies the name of the source database table containing the data to be exported.

**Export Filename:** A string that specifies the path (relative or absolute) and file of the bulk load data file to which the data is to be exported. It also specifies the file name of the bulk configuration file. These files must not already exist; if one of both of them already exists, an error is returned.



**Log Filename:** A string that specifies the path (relative or absolute) and file name of the bulk log file. The log file is created if it does not exist. Events logged to this file are:

- Total number of rows fetched
- A message for each row that failed to export
- Total number of rows that failed to export
- Total number of rows successfully exported

Information about the load is written to this file, preceded by a header. Information about the next load is appended to the end of the file.

If you do not supply a value for Log Filename, no log file is created.

**Error Tolerance:** A value that specifies the number of errors to tolerate before an operation terminates. A value of 0 indicates that no errors are tolerated; the operation fails when the first error is encountered.

The default of -1 means that an infinite number of errors is tolerated.

**Warning Tolerance:** A value that specifies the number of warnings to tolerate before an operation terminates. A value of 0 indicates that no warnings are tolerated; the operation fails when the first warning is encountered.

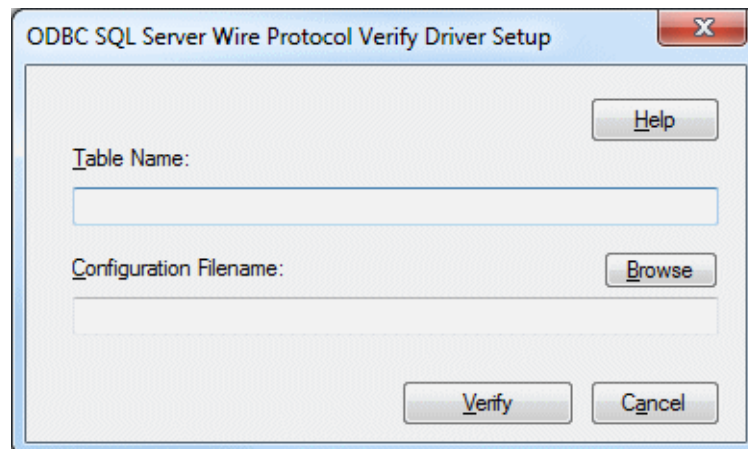
The default of -1 means that an infinite number of warnings is tolerated.

**Code Page:** A value that specifies the code page value to which the driver must convert all data for storage in the bulk data file. See ["Character Set Conversions" on page 94](#) for more information.

The default value on Windows is the current code page of the machine. On UNIX/Linux, the default value is 4 (ISO 8559-1 Latin-1).

Click **Export Table** to connect to the database and export data to the bulk data file or click **Cancel**.

To verify the metadata of the bulk load configuration file against the structure of the target database table, click **Verify** from the Bulk tab. See ["Verification of the Bulk Load Configuration File" on page 92](#) for details. The Verify dialog box appears.



**Table Name:** A string that specifies the name of the target database table into which the data is to be loaded.

**Configuration Filename:** A string that specifies the path (relative or absolute) and file name of the bulk configuration file.

Click **Verify** to verify table structure or click **Cancel**.

- b** To load data from the bulk data file to a database table, click **Load Table** from the Bulk tab. The Load File dialog box appears.

ODBC SQL Server Wire Protocol Load File Driver Setup

Table Name:  [Help](#)

Load Data Filename:  [Browse](#)

Configuration Filename:  [Browse](#)

Log Filename:  [Browse](#)

Discard Filename:  [Browse](#)

Error Tolerance:  Warning Tolerance:

Load Start:  Load Count:

Read Buffer Size (KB):

[Load Table](#) [Cancel](#)

The load operation can create a log file and can also create a discard file that contains rows rejected during the load. The discard file is in the same format as the bulk load data file. After fixing reported issues in the discard file, the bulk load can be reissued using the discard file as the bulk load data file.

The export operation can be configured such that if any errors or warnings occur:

- The operation always completes.
- The operation always terminates.
- The operation terminates after a certain threshold of warnings or errors is exceeded.

If a load fails, the Load Start and Load Count options can be used to control which rows are loaded when a load is restarted after a failure.

**Table Name:** A string that specifies the name of the target database table into which the data is loaded.

**Load Data Filename:** A string that specifies the path (relative or absolute) and file name of the bulk data file from which the data is loaded.

**Configuration Filename:** A string that specifies the path (relative or absolute) and file name of the bulk configuration file.

**Log Filename:** A string that specifies the path (relative or absolute) and file name of the bulk log file. Specifying a value for Log Filename creates the file if it does not already exist. Events logged to this file are:

- Total number of rows read
- Message for each row that failed to load
- Total number of rows that failed to load
- Total number of rows successfully loaded

Information about the load is written to this file, preceded by a header. Information about the next load is appended to the end of the file.

If you do not specify a value for Log Filename, no log file is created.

**Discard Filename:** A string that specifies the path (relative or absolute) and file name of the bulk discard file. Any row that cannot be inserted into database as result of bulk load is added to this file, with the last row rejected added to the end of the file.

Information about the load is written to this file, preceded by a header. Information about the next load is appended to the end of the file.

If you do not specify a value for Discard Filename, a discard file is not created.

**Error Tolerance:** A value that specifies the number of errors to tolerate before an operation terminates. A value of 0 indicates that no errors are tolerated; the operation fails when the first error is encountered.

The default of -1 means that an infinite number of errors is tolerated.

**Load Start:** A value that specifies the first row to be loaded from the data file. Rows are numbered starting with 1. For example, when Load Start is 10, the first 9 rows of the file are skipped and the first row loaded is row 10. This option can be used to restart a load after a failure.

The default value is 1.

**Read Buffer Size (KB):** A value that specifies the size, in KB, of the buffer that is used to read the bulk data file for a bulk load operation.

The default value is 2048.

**Warning Tolerance:** A value that specifies the number of warnings to tolerate before an operation terminates. A value of 0 indicates that no warnings are tolerated; the operation fails when the first warning is encountered.

The default of -1 means that an infinite number of warnings is tolerated.

**Load Count:** A value that specifies the number of rows to be loaded from the data file. The bulk load operation loads rows up to the value of Load Count from the file to the database. It is valid for Load Count to specify more rows than exist in the data file. The bulk load operation completes successfully when either the number of

rows specified by the Load Count value has been loaded or the end of the data file is reached. This option can be used in conjunction with Load Start to restart a load after a failure.

The default value is the maximum value for SQLULEN. If set to 0, no rows are loaded.

Click **Load Table** to connect to the database and load the table or click **Cancel**.

At any point during the configuration process, you can click **Test Connect** to attempt to connect to the data source using the connection options specified in the driver Setup dialog box. A Logon dialog box appears; see ["Using a Connection String" on page 388](#) for details. 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 incorrect environment or connection value, it displays an appropriate error message. Click **OK**.

NOTE: If you are configuring alternate servers for use with the connection failover feature, be aware that the Test Connect button tests only the primary server, not the alternate servers.

- 8 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.

## Using a Connection String

If you want to use a connection string for connecting to a database, or if your application requires it, you must specify either a DSN (data source name), a File DSN, or a DSN-less connection in the string. The difference is whether you use the DSN=, FILEDSN=, or the DRIVER= keyword in the connection string, as described in the ODBC specification. A DSN or FILEDSN connection string tells the driver where to find the default connection information. Optionally, you may specify *attribute=value* pairs in the connection string to override the default values stored in the data source.

The DSN connection string has the form:

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

The FILEDSN connection string has the form:

```
FILEDSN=filename.dsn[;attribute=value[;attribute=value]...]
```

The DSN-less connection string specifies a driver instead of a data source. All connection information must be entered in the connection string because the information is not stored in a data source.

The DSN-less connection string has the form:

```
DRIVER={{driver_name}}[;attribute=value[;attribute=value]...]
```

[Table 11-1 on page 390](#) lists the long and short names for each attribute, as well as the initial default value when the driver is first installed. You can specify either long or short names in the connection string.

An example of a DSN connection string with overriding attribute values for Microsoft SQL Server is:

```
DSN=ACCOUNTING;DATABASE=ACCT
```

A FILEDSN connection string is similar except for the initial keyword:

```
FILEDSN=SQLServer.dsn;DATABASE=ACCT
```

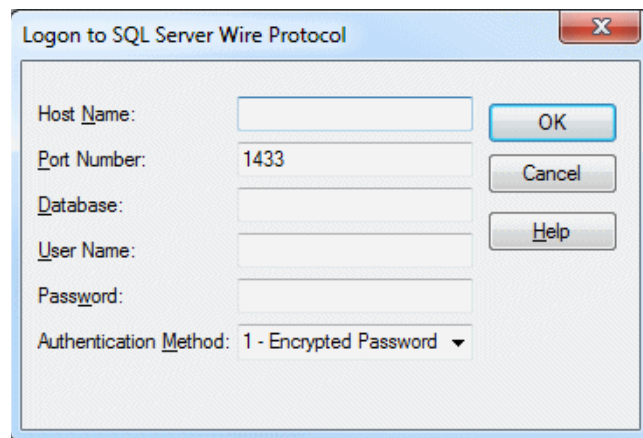
A DSN-less connection string must provide all necessary connection information:

```
DRIVER=DataDirect 7.1 SQL Server Wire Protocol;HOST=SQLServer1;PORT=1433;  
UID=JOHN;PWD=XYZZY;DB=SQLSdb1
```

## 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.

NOTE: The Login Dialog is not displayed if Authentication Mode has previously been set to Kerberos and the Host Name is specified in the data source.



In the Logon dialog box, provide the following information:

- 1 Type an IP address in Host Name in following format: *IP\_address*. For example, you can enter 199.226.224.34.

The IP address can be specified in IPv4 on Windows, and in either IPv4 or IPv6 format, or a combination of the two, on UNIX. See ["Using IP Addresses" on page 54](#) for details about these formats.

If your network supports named servers, you can specify an address as: *server\_name*. For example, you can enter *SSserver*.

To specify a named instance of Microsoft SQL Server, use the format:  
*server\_name\instance\_name*. If only a server name is specified with no instance name, the driver uses the default instance on the server.

- 2 Type the Port Number of the server listener.
- 3 Type the name of the database to which you want to connect. If you do not specify a value, the default database that is defined by Microsoft SQL Server is used.
- 4 Type your Microsoft SQL Server login ID.
- 5 Type your password.
- 6 Select an Authentication Method:  
  
If set to 1 - Encrypt Password, the driver sends the user ID in clear text and an encrypted password to the server for authentication.  
  
If set to 4 - Kerberos, the driver uses Kerberos authentication. This method supports both Windows Active Directory Kerberos and MIT Kerberos environments.
- 7 Click **OK** to complete the logon and to update the values in the Registry.

---

# Connection Option Descriptions

The following connection option descriptions are listed alphabetically by the GUI name that appears on the driver Setup dialog box. The connection string attribute name, along with its short name, is listed immediately underneath the GUI name. For example:

## Application Using Threads

Attribute    ApplicationUsingThreads (AUT)

In most cases, the GUI name and the attribute name are the same; however, some exceptions exist. If you need to look up an option by its connection string attribute name, please refer to the alphabetical table of connection string attribute names.

Also, a few connection string attributes, for example, Password, do not have equivalent options that appear on the GUI. They are in the list of descriptions alphabetically by their attribute names.

[Table 11-1](#) lists the connection string attributes supported by the SQL Server Wire Protocol driver.

---

**Table 11-1. SQL Server Wire Protocol Attribute Names**

---

Attribute (Short Name)	Default
<a href="#">AlternateServers (ASRV)</a>	None
<a href="#">AlwaysReportTriggerResults (ARTR)</a>	0 (Disabled)
<a href="#">AnsiNPW (ANPW)</a>	1 (Enabled)
<a href="#">ApplicationName (APP)</a>	None

**Table 11-1. SQL Server Wire Protocol Attribute Names** (cont.)

Attribute (Short Name)	Default
ApplicationUsingThreads (AUT)	1 (Enabled)
AuthenticationMethod (AM)	1 (Encrypt Password)
BulkBinaryThreshold (BBT)	32
BulkCharacterThreshold (BCT)	-1
BulkLoadBatchSize (BLBS)	1024
BulkLoadOptions (BLO)	2
BulkLoadFieldDelimiter (BLFD)	None
BulkLoadRecordDelimiter (BLRD)	None
ConnectionReset (CR)	0 (Disabled)
ConnectionRetryCount (CRC)	0
ConnectionRetryDelay (CRD)	3
Database (DB)	None
DataSourceName (DSN)	None
Description (n/a)	None
Domain	None
EnableBulkLoad (EBL)	0 (Disabled)
EnableQuotedIdentifiers (EQI)	0 (Disabled)
EncryptionMethod (EM)	0 (No Encryption)
FailoverGranularity (FG)	0 (Non-Atomic)
FailoverMode (FM)	0 (Connection)
FailoverPreconnect (FP)	0 (Disabled)
FetchTSWTZasTimestamp (FTSWTZAT)	0 (Disabled)
FetchTWFSasTime (FTWFSAT)	1 (Enabled)
GSSClient (GSSC)	native
HostName (HOST)	None
HostNameInCertificate (HNIC)	None
IANAAppCodePage (IACP)	4 (ISO 8559-1 Latin-1)
UNIX ONLY	
InitializationString (IS)	None
Language (LANG)	None
LoadBalanceTimeout (LBT)	0
LoadBalancing (LB)	0 (Disabled)
LoginTimeout (LT)	None
LogonID (UID)	None
MaxPoolSize (MXPS)	100
MinPoolSize (MNPS)	0
PacketSize (PS)	-1
Password (PWD)	None

**Table 11-1. SQL Server Wire Protocol Attribute Names** (cont.)

Attribute (Short Name)	Default
Pooling (POOL)	0 (disabled)
PortNumber (PORT)	1433
QueryTimeout (QT)	0
ReportCodepageConversionErrors (RCCE)	0 (Ignore Errors)
SnapshotSerializable (SS)	0 (Disabled)
Truststore (TS)	None
TruststorePassword (TSP)	None
User Name	None
ValidateServerCertificate (VSC)	1 (enabled)
WorkstationID (WSID)	None
XMLDescribeType (XDT)	-10

Alternate Servers

Attribute	AlternateServers (ASRV)
Description	A list of alternate database servers to which the driver tries to connect if the primary database server is unavailable. Specifying a value for this option enables connection failover for the driver. The value you specify must be in the form of a string that defines the physical location of each alternate server. All of the other required connection information for each alternate server is the same as what is defined for the primary server connection.
Valid Values	<p>(HostName=hostvalue:PortNumber=portvalue:Database=databasevalue[, . . .])</p> <p>You must specify the host name, port number, and database name of each alternate server.</p> <p>NOTE: An alternate server address in IPv6 format must be enclosed in double quotation marks.</p>
Example	<p>The following Alternate Servers value defines two alternate database servers for connection failover:</p> <pre>AlternateServers=(HostName=SqlsServer:PortNumber=1433:Database=Sqlsdb1, HostName=255.201.11.24:PortNumber=1434:Database=Sqlsdb2)</pre>
Default	None
GUI Tab	Failover tab on page 381

Always Report Trigger Results

Attribute	AlwaysReportTriggerResults (ARTR)
Description	Determines how the driver reports results that are generated by database triggers (procedures that are stored in the database and executed, or fired, when a table is modified). For Microsoft SQL Server 2005 and higher and Windows Azure SQL Database, this includes triggers that are fired by Data Definition Language (DDL) events.



Valid Values 0 | 1

If set to 1 (Enabled), the driver returns all results, including results that are generated by triggers. Multiple trigger results are returned one at a time. You can use the `SQLMoreResults` function to return individual trigger results. Warnings and errors are reported in the results as they are encountered.

If set to 0 (Disabled):

- For Microsoft SQL Server 2005 and higher and Windows Azure SQL Database, the driver does not report trigger results if the statement is a single INSERT, UPDATE, DELETE, CREATE, ALTER, DROP, GRANT, REVOKE, or DENY statement.
- For other Microsoft SQL Server databases, the driver does not report trigger results if the statement is a single INSERT, UPDATE, or DELETE statement.

When set to 0, the only result that is returned is the update count that is generated by the statement that was executed (if no errors occurred). Although trigger results are ignored, any errors and warnings that are generated by the trigger are reported. If errors are reported, the update count is not reported.

Default 0 (Disabled)

GUI Tab [Advanced tab](#) on page 378

## AnsiNPW

Attribute AnsiNPW (ANPW)

Description Determines whether ANSI-defined behaviors are exposed. Setting this option has no effect on NULL concatenation for Windows Azure SQL Database or SQL Server versions higher than SQL Server 2012.

Valid Values 0 | 1

When set to 1 (Enabled), the driver sets four ANSI-defined behaviors for handling NULL comparisons: NULLS, character data padding, warnings, and NULL concatenation.

When set to 0 (Disabled), ANSI-defined behaviors are not exposed. If the driver appears to be truncating trailing blank spaces, set this attribute to 0 (Disabled).

Default 1 (Enabled)

GUI Tab [Advanced tab](#) on page 378

## Application Name

Attribute ApplicationName (APP)

Description The name the database uses to identify your application.

Valid Values *string*

where *string* is your application name.


Default None

GUI Tab [Advanced tab](#) on page 378

## Application Using Threads

Attribute	ApplicationUsingThreads (AUT)
Description	Determines whether the driver works with applications using multiple ODBC threads.  This connection option can affect performance. See <a href="#">"Performance Considerations" on page 413</a> for details.
Valid Values	0   1  If set to 1 (Enabled), the driver works with single-threaded and multi-threaded applications.  If set to 0 (Disabled), the driver does not work with multi-threaded applications. If using the driver with single-threaded applications, this value avoids additional processing required for ODBC thread-safety standards.
Default	1 (Enabled)
GUI Tab	<a href="#">Advanced tab</a> on page 378

## Authentication Method

Attribute	AuthenticationMethod (AM)
Description	Specifies the method the driver uses to authenticate the user to the server when a connection is established. If the specified authentication method is not supported by the database server, the connection fails and the driver generates an error.
Valid Values	1   4   9  If set to 1 (Encrypt Password), the driver sends the user ID in clear text and an encrypted password to the server for authentication.  If set to 4 (Kerberos Authentication), the driver uses Kerberos authentication. This method supports both Windows Active Directory Kerberos and MIT Kerberos environments.  NOTE: Microsoft Windows Azure SQL Database supports only SQL Server authentication. You must provide credentials every time when you connect to SQL Database.  Setting this value to 4 also enables NTLM authentication on Windows platforms.   If set to 9 on Linux and UNIX platforms, the driver uses NTLM authentication. To connect to the database, users must supply the Windows User Id, Password, and, in some cases, Domain to the driver. Note that NTLM single sign on is supported only on Windows.
Default	1 (Encrypt Password)
GUI Tab	<a href="#">Security tab</a> on page 379

## Batch Size

Attribute	BulkLoadBatchSize (BLBS)
Description	The number of rows that the driver sends to the database at a time during bulk operations. This value applies to all methods of bulk loading.
Valid Values	x

where  $x$  is a positive integer that specifies the number of rows to be sent.

Default 1024

GUI Tab [Bulk tab](#) on page 383

## Bulk Binary Threshold

Attribute BulkBinaryThreshold (BBT)

Description The maximum size, in KB, of binary data that is exported to the bulk data file.

Valid Values -1 | 0 |  $x$

where  $x$  is an integer that specifies the number of KB.

If set to -1, all binary data, regardless of size, is written to the bulk data file, not to an external file.

If set to 0, all binary data, regardless of size, is written to an external file, not the bulk data file. A reference to the external file is written to the bulk data file.

If set to  $x$ , any binary data exceeding this specified number of KB is written to an external file, not the bulk data file. A reference to the external file is written to the bulk data file.

Default 32

GUI Tab [Bulk tab](#) on page 383

## Bulk Character Threshold

Attribute BulkCharacterThreshold (BCT)

Description The maximum size, in KB, of character data that is exported to the bulk data file.

Valid Values -1 | 0 |  $x$

where  $x$  is an integer that specifies the number of KB.

If set to -1, all character data, regardless of size, is written to the bulk data file, not to an external file.

If set to 0, all character data regardless of size, is written to an external file, not the bulk data file. A reference to the external file is written to the bulk data file.

If set to  $x$ , any character data exceeding this specified number of KB is written to an external file, not the bulk data file. A reference to the external file is written to the bulk data file.

Default -1

GUI Tab [Bulk tab](#) on page 383

## Bulk Options

Attribute	BulkLoadOptions (BLO)
Description	Toggles options for the bulk load process.
Valid Values	0   x

where x is a positive integer representing the cumulative total of the Bulk Options values.

If set to 0, none of the options for bulk load are enabled.

If set to x, the values represented by x are enabled.

NOTE: The cumulative value of the options is only used in a connection string with the connection string attribute, BulkLoadOptions. On the Bulk tab of the driver Setup dialog, the individual options are enabled by selecting the appropriate check box.

The following bulk load options are available:

- Check Constraints - Checks constraints while data is being inserted. Value=16.
- Fire Triggers - Causes the server to fire the insert triggers for rows being inserted into the database. Value=32.
- Keep Identity - Preserves source identity values. When not enabled, identity values are assigned by the destination. Value=1.
- Keep Nulls - Preserves null values in the destination table regardless of the settings for default values. When not enabled, null values are replaced by column default values, where applicable. Value=64.
- Table Lock - Assigns a table lock for the duration of the bulk copy operation. Other applications are not permitted to update the table during the copy operation. When not enabled, the default bulk locking mechanism (row or table) specified by the table lock on bulk load server option is used. Value=2.

**Example** If you wanted to enable Check Constraints (16), Fire Triggers (32), and Keep Identity (1) in a connection string, you would add the values together:

```
BulkLoadOptions=49
```

To enable these options on the Bulk tab of the driver Setup dialog, you would simply select the check box for each one.

Default	2 (Table Lock enabled)
GUI Tab	<a href="#">Bulk tab</a> on page 383

## Connection Pooling

Attribute	Pooling (POOL)
Description	Specifies whether to use the driver's connection pooling.

NOTE: The application must be thread-enabled to use connection pooling.

This connection option can affect performance. See ["Performance Considerations" on page 413](#) for details.

Valid Values	0   1
	If set to 1 (Enabled), the driver uses connection pooling.
	If set to 0 (Disabled), the driver does not use connection pooling.
Default	0 (Disabled)
GUI Tab	<a href="#">Pooling tab</a> on page 382

## Connection Reset

Attribute	ConnectionReset (CR)
Description	<p>Determines whether the state of connections that are removed from the connection pool for reuse by the application is reset to the initial configuration of the connection.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 413</a> for details.</p>
Valid Values	0   1
	<p>If set to 1 (Enabled), the state of connections removed from the connection pool for reuse by an application is reset to the initial configuration of the connection. Resetting the state can negatively impact performance because additional commands must be sent over the network to the server to reset the state of the connection.</p> <p>If set to 0 (Disabled), the state of connections is not reset.</p>
Default	0 (Disabled)
GUI Tab	<a href="#">Pooling tab</a> on page 382

## Connection Retry Count

Attribute	ConnectionRetryCount (CRC)
Description	<p>The number of times the driver retries connection attempts to the primary database server, and if specified, alternate servers until a successful connection is established.</p> <p>This option and the Connection Retry Delay connection option, which specifies the wait interval between attempts, can be used in conjunction with failover.</p>
Valid Values	0   $x$
	<p>where <math>x</math> is a positive integer from 1 to 65535.</p> <p>If set to 0, the driver does not try to connect after the initial unsuccessful attempt.</p> <p>If set to <math>x</math>, the driver retries connection attempts the specified number of times. If a connection is not established during the retry attempts, the driver returns an error that is generated by the last server to which it tried to connect.</p>
Default	0
GUI Tab	<a href="#">Failover tab</a> on page 381

## Connection Retry Delay

Attribute	ConnectionRetryDelay (CRD)
Description	<p>The number of seconds the driver waits between connection retry attempts when Connection Retry Count is set to a positive integer.</p> <p>This option and the Connection Retry Count connection option can be used in conjunction with failover.</p>
Valid Values	<p>0   <math>x</math></p> <p>where <math>x</math> is a positive integer from 1 to 65535.</p> <p>If set to 0, there is no delay between retries.</p> <p>If set to <math>x</math>, the driver waits the specified number of seconds between connection retry attempts.</p>
Default	3
GUI Tab	<a href="#">Failover tab</a> on page 381

## Data Source Name

Attribute	DataSourceName (DSN)
Description	The name of a data source in your Windows Registry or odbc.ini file.
Valid Values	<p><i>string</i></p> <p>where <i>string</i> is the name of a data source.</p>
Default	None
GUI Tab	<a href="#">General tab</a> on page 377

## Database

Attribute	Database (DB)
Description	The name of the database to which you want to connect.
Valid Values	<p><i>database_name</i></p> <p>where <i>database_name</i> is the name of a valid database.</p>
Default	None
GUI Tab	<a href="#">General tab</a> on page 377

## Description

Attribute	Description (n/a)
Description	An optional long description of a data source. This description is not used as a runtime connection attribute, but does appear in the ODBC.INI section of the Registry and in the odbc.ini file.
Valid Values	<i>string</i>  where <i>string</i> is a description of a data source.
Default	None
GUI Tab	<a href="#">General tab</a> on page 377



## Domain

Attribute	Domain (DOM)
Description	Specifies the Windows domain that the driver uses when connecting to a SQL Server Instance.  To connect to the database, users must supply the Windows User Id, Password, and, in some cases, domain to the driver. NTLM single sign on is not supported.
Valid Values	<i>string</i>  where <i>string</i> is a valid Windows domain for the user specified by LoginId. This attribute applies only when Authentication Mode is set to 9.
Default	None
GUI Tab	n/a

## Enable Bulk Load

Attribute	EnableBulkLoad (EBL)
Description	Specifies the bulk load method.
Valid Values	0   1  If set to 1 (Enabled), the driver uses the database bulk load protocol when an application executes an INSERT with multiple rows of parameter data. If the protocol cannot be used, the driver returns a warning.  If set to 0 (Disabled), the driver uses standard parameter arrays.
Default	0 (Disabled)
GUI Tab	<a href="#">Bulk tab</a> on page 383

## Enable Quoted Identifiers

Attribute	EnableQuotedIdentifiers (EQI)
Description	Determines whether the driver allows the use of quoted identifiers.

Valid Values 0 | 1

If set to 1 (Enabled), the database enforces ANSI rules regarding quotation marks. Double quotation marks can only be used for identifiers, such as column and table names. Character strings must be enclosed in single quotation marks, for example:

```
SELECT "au_id"
FROM "authors"
WHERE "au_lname" = 'O'Brien'
```

If set to no (Disabled), applications that use quoted identifiers encounter errors when they generate SQL statements with quoted identifiers.

Default 0 (Disabled)

GUI Tab [Advanced tab](#) on page 378

## Encryption Method

Attribute EncryptionMethod (EM)

Description The method the driver uses to encrypt data sent between the driver and the database server. If the specified encryption method is not supported by the database server, the connection fails and the driver returns an error.

This connection option can affect performance. See ["Performance Considerations" on page 413](#) for details.

Valid Values 0 | 1 | 6 | 7

If set to 0 (No Encryption), data is not encrypted.

If set to 1 (SSL), data is encrypted using SSL. If the server supports protocol negotiation, the driver and server negotiate the use of TLS v1, SSL v3, or SSL v2 in that order.

If set to 6 (RequestSSL) - The login request and data are encrypted using SSL if the server is configured for SSL. If the server is not configured for SSL, an unencrypted connection is established.

If set to 7 (LoginSSL) - The login request is encrypted using SSL regardless of whether the server is configured for SSL. The data is encrypted using SSL if the server is configured for SSL, and the data is unencrypted if the server is not configured for SSL.

NOTE: The driver must use the server-specified packet size when using SSL encryption. If SSL is used, any value set for the Packet Size connection option is ignored.

This option can only be set to 1 when Authentication Method is set to 1.

Default 0 (No Encryption)

GUI Tab [Security tab](#) on page 379

## Failover Granularity

Attribute FailoverGranularity (FG)

Description Determines whether the driver fails the entire failover process or continues with the process if errors occur while trying to reestablish a lost connection.



This option applies only when Failover Mode is set to 1 (Extended Connection) or 2 (Select).

The Alternate Servers option specifies one or multiple alternate servers for failover and is required for all failover methods.

Valid Values 0 | 1 | 2 | 3

If set to 0 (Non-Atomic), the driver continues with the failover process and posts any errors on the statement on which they occur.

If set to 1 (Atomic) the driver fails the entire failover process if an error is generated as the result of anything other than executing and repositioning a Select statement. If an error is generated as a result of repositioning a result set to the last row position, the driver continues with the failover process, but generates a warning that the Select statement must be reissued.

If set to 2 (Atomic Including Repositioning), the driver fails the entire failover process if any error is generated as the result of restoring the state of the connection or the state of work in progress.

If set to 3 (Disable Integrity Check), the driver does not verify that the rows that were restored during the failover process match the original rows. This value applies only when Failover Mode is set to 2 (Select).

Default 0 (Non-Atomic)

GUI Tab [Failover tab](#) on page 381

## Failover Mode

Attribute FailoverMode (FM)

Description Specifies the type of failover method the driver uses.

The Alternate Servers option specifies one or multiple alternate servers for failover and is required for all failover methods.

This connection option can affect performance. See ["Performance Considerations" on page 413](#) for details.

Valid Values 0 | 1 | 2

If set to 0 (Connection), the driver provides failover protection for new connections only.

If set to 1 (Extended Connection), the driver provides failover protection for new and lost connections, but not any work in progress.

If set to 2 (Select), the driver provides failover protection for new and lost connections. In addition, it preserves the state of work performed by the last Select statement executed.

Default 0 (Connection)

GUI Tab [Failover tab](#) on page 381

## Failover Preconnect

Attribute	FailoverPreconnect (FP)
Description	<p>Specifies whether the driver tries to connect to the primary and an alternate server at the same time.</p> <p>This attribute applies only when Failover Mode is set to 1 (Extended Connection) or 2 (Select) and at least one alternate server is specified.</p> <p>The Alternate Servers option specifies one or multiple alternate servers for failover and is required for all failover methods.</p>
Valid Values	<p>0   1</p> <p>If set to 0 (Disabled), the driver tries to connect to an alternate server only when failover is caused by an unsuccessful connection attempt or a lost connection. This value provides the best performance, but your application typically experiences a short wait while the failover connection is attempted.</p> <p>If set to 1 (Enabled), the driver tries to connect to the primary and an alternate server at the same time. This can be useful if your application is time-sensitive and cannot absorb the wait for the failover connection to succeed.</p>
Default	0 (Disabled)
GUI Tab	<a href="#">Failover tab</a> on page 381

## Fetch TSWTZ as Timestamp

Attribute	FetchTSWTZasTimestamp (FTSWTZAT)
Description	Determines whether the driver returns column values with the timestamp with time zone data type as the ODBC data type SQL_TYPE_TIMESTAMP or SQL_VARCHAR.
Valid Values	<p>0   1</p> <p>If set to 1 (Enabled), the driver returns column values with the timestamp with time zone data type as the ODBC type SQL_TYPE_TIMESTAMP. The time zone information in the fetched value is truncated. Use this value if your application needs to process values the same way as TIMESTAMP columns.</p> <p>If set to 0 (Disabled), the driver returns column values with the timestamp with time zone data type as the ODBC data type SQL_VARCHAR. Use this value if your application requires the time zone information in the fetched value.</p>
Default	0 (Disabled)
GUI Tab	<a href="#">Advanced tab</a> on page 378

## Fetch TWFS as Time

Attribute	FetchTWFSasTime (FTWFSAT)
Description	<p>Determines whether the driver returns column values with the time data type as the ODBC data type SQL_TYPE_TIME or SQL_TYPE_TIMESTAMP.</p> <p>Supported only for Microsoft SQL Server 2008.</p>

Valid Values 0 | 1

If set to 1 (Enabled), the driver returns column values with the time data type as the ODBC data type SQL\_TYPE\_TIME. The fractional seconds portion of the value is truncated.

If set to 0 (Disabled), the driver returns column values with the time data type as the ODBC data type SQL\_TYPE\_TIMESTAMP. The fractional seconds portion of the value is preserved. Time columns are not searchable when they are described and fetched as timestamp

NOTE: When returning time with fractional seconds data as SQL\_TYPE\_TIMESTAMP, the Year, Month and Day parts of the timestamp must be set to zero.

Default 1 (Enabled)

GUI Tab [Advanced tab](#) on page 378

## Field Delimiter

Attribute BulkLoadFieldDelimiter (BLFD)

Description Specifies the character that the driver will use to delimit the field entries in a bulk load data file.

Valid Values *x*

where *x* is any printable character.

For simplicity, avoid using a value that can be in the data, including all alphanumeric characters, the dash(-), the colon(:), the period (.), the forward slash (/), the space character, the single quote (') and the double quote ("). You can use some of these characters as delimiters if all of the data in the file is contained within double quotes.

NOTE: The Bulk Load Field Delimiter character must be different from the Bulk Load Record Delimiter.

Default None

GUI Tab [Bulk tab](#) on page 383

## GSS Client Library

Attribute GSSClient (GSSC)

Description The name of the GSS client library that the driver uses to communicate with the Key Distribution Center (KDC).

The driver uses the path defined by the PATH environment variable for loading the specified client library.

Valid Values native | *client\_library*

where *client\_library* is a GSS client library installed on the client.

If set to *client\_library*, the driver uses the specified GSS client library.

If set to native, the driver uses the GSS client shipped with the operating system.

Default	native
GUI Tab	<a href="#">Security tab</a> on page 379

Host Name

Attribute	HostName (HOST)
Description	The name or the IP address of the server to which you want to connect.
Valid Values	<i>IP_address</i>   <i>named_server</i>   <i>named_instance</i>

where:

*IP\_address* is the IP address of the server to which you want to connect. Specify this address as: *IP\_address*. For example, you can enter 199.226.224.34.

The IP address can be specified in either IPv4 or IPv6 format, or a combination of the two. See ["Using IP Addresses" on page 54](#) for details about these formats.

*named\_server* is the named server address of the server to which you want to connect. Specify this address as: *named\_server*. For example, you can enter SSserver.

*named\_instance* is a named instance of Microsoft SQL Server or Windows Azure SQL Database. Specify this address as: *server\_name\instance\_name*.

- If only a server name is specified with no instance name, the driver uses the default instance on the server.
- If only a server name is specified with a backward slash \ or \\* at the end with no instance name, the driver uses the first instance on the server with a TCP port.

Default	None
GUI Tab	<a href="#">General tab</a> on page 377

Host Name In Certificate

Attribute	HostNameInCertificate (HNIC)
Description	A host name for certificate validation when SSL encryption is enabled (Encryption Method=1) and validation is enabled (Validate Server Certificate=1). This option provides additional security against man-in-the-middle (MITM) attacks by ensuring that the server the driver is connecting to is the server that was requested.
Valid Values	<i>host_name</i>   #SERVERNAME#

where the *host\_name* is the host name specified in the certificate. Consult your SSL administrator for the correct value.

If set to a host name, the driver examines the subjectAltName values included in the certificate. If a dnsName value is present in the subjectAltName values, then the driver compares the value specified for Host Name In Certificate with the dnsName value. The connection succeeds if the values match. The connection fails if the Host Name In Certificate value does not match the dnsName value.

If no subjectAltName values exist or a dnsName value is not in the list of subjectAltName values, then the driver compares the value specified for Host Name In Certificate with the

commonName part of the Subject name in the certificate. The commonName typically contains the host name of the machine for which the certificate was created. The connection succeeds if the values match. The connection fails if the Host Name In Certificate value does not match the commonName. If multiple commonName parts exist in the Subject name of the certificate, the connection succeeds if the Host Name In Certificate value matches any of the commonName parts.

If set to `#SERVERNAME#`, the driver compares the host server name specified as part of a data source or connection string to the `dnsName` or the `commonName` value.

Default None

GUI Tab [Security tab](#) on page 379



## IANAAppCodePage

IANAAppCodePage (IACP)

**Description** An Internet Assigned Numbers Authority (IANA) value. You must specify a value for this option if your application is not Unicode-enabled or if your database character set is not Unicode. Refer to [Chapter 4 “Internationalization, Localization, and Unicode”](#) in the *DataDirect Connect Series for ODBC Reference* for details.

The driver uses the specified IANA code page to convert "W" (wide) functions to ANSI.

The driver and Driver Manager both check for the value of IANAAppCodePage in the following order:

- In the connection string
- In the Data Source section of the system information file (odbc.ini)
- In the ODBC section of the system information file (odbc.ini)

If the driver does not find an IANAAppCodePage value, the driver uses the default value of 4 (ISO 8859-1 Latin-1).

**Valid Values** *IANA\_code\_page*

where *IANA\_code\_page* is one of the valid values listed in [Chapter 1 “Values for the Attribute IANAAppCodePage”](#) in the *DataDirect Connect Series for ODBC Reference*. The value must match the database character encoding and the system locale.

Default 4 (ISO 8559-1 Latin-1)

GUI Tab [Advanced tab](#) on page 378

## Initialization String

**Attribute** InitializationString (IS)

**Description** A SQL command that is issued immediately after connecting to the database to manage session settings.

**NOTE:** If the statement fails to execute, the connection fails and the driver reports the error returned from the server.

**Valid Values** *SQL\_command*

where *SQL\_command* is a valid SQL command that is supported by the database.

Example To set the date format on every connection, specify:

Set DateStyle='ISO, MDY'

Default None

GUI Tab [Advanced tab](#) on page 378

## Language

Attribute Language (LANG)

Description The national language to use for Microsoft SQL Server system messages.

Valid Values *lang*

where *lang* is the language to use for Microsoft SQL Server system messages. This overrides the default language specified for the login on the server. If no language is specified, the connection uses the default language specified for the login on the server.

Default None

GUI Tab [Advanced tab](#) on page 378

## Load Balance Timeout

Attribute LoadBalanceTimeout (LBT)

Description The number of seconds to keep inactive connections open in a connection pool. An inactive connection is a database session that is not associated with an ODBC connection handle, that is, a connection in the pool that is not in use by an application.

NOTE: The Min Pool Size option may cause some connections to ignore this value.

This connection option can affect performance. See ["Performance Considerations" on page 413](#) for details.

Valid Values 0 | *x*

where *x* is a positive integer that specifies a number of seconds.

If set to 0, inactive connections are kept open.

If set to *x*, inactive connections are closed after the specified number of seconds passes.

Default 0

GUI Tab [Pooling tab](#) on page 382

## Load Balancing

Attribute LoadBalancing (LB)

Description Determines whether the driver uses client load balancing in its attempts to connect to the database servers (primary and alternate). You can specify one or multiple alternate servers by setting the Alternate Servers option.

Valid Values 0 | 1

If set to 1 (Enabled), the driver uses client load balancing and attempts to connect to the database servers (primary and alternate servers) in random order.

If set to 0 (Disabled), the driver does not use client load balancing and connects to each server based on their sequential order (primary server first, then, alternate servers in the order they are specified).

NOTE: This option has no effect unless alternate servers are defined for the Alternate Servers connection option.

Default 0 (Disabled)

GUI Tab [Failover tab](#) on page 381

## Login Timeout

Attribute LoginTimeout (LT)

Description The number of seconds the driver waits for a connection to be established before returning control to the application and generating a timeout error. To override the value that is set by this connection option for an individual connection, set a different value in the SQL\_ATTR\_LOGIN\_TIMEOUT connection attribute using the SQLSetConnectAttr() function.

Valid Values -1 | 0 |  $x$

where  $x$  is a positive integer that specifies a number of seconds.

If set to -1, the connection request does not time out. The driver silently ignores the SQL\_ATTR\_LOGIN\_TIMEOUT attribute.

If set to 0, the connection request does not time out, but the driver responds to the SQL\_ATTR\_LOGIN\_TIMEOUT attribute.

If set to  $x$ , the connection request times out after the specified number of seconds unless the application overrides this setting with the SQL\_ATTR\_LOGIN\_TIMEOUT attribute.

Default 15

GUI Tab [Advanced tab](#) on page 378

## Max Pool Size

Attribute MaxPoolSize (MXPS)

Description The maximum number of connections allowed within a single connection pool. When the maximum number of connections is reached, no additional connections can be created in the connection pool.

This connection option can affect performance. See ["Performance Considerations" on page 413](#) for details.

Valid Values An integer from 1 to 65535

For example, if set to 20, the maximum number of connections allowed in the pool is 20.

Default 100  
GUI Tab [Pooling tab](#) on page 382

Min Pool Size

Attribute MinPoolSize (MNPS)  
Description The minimum number of connections that are opened and placed in a connection pool, in addition to the active connection, when the pool is created. The connection pool retains this number of connections, even when some connections exceed their Load Balance Timeout value.  
  
This connection option can affect performance. See ["Performance Considerations" on page 413](#) for details.  
Valid Values 0 | x  
  
where x is an integer from 1 to 65535.  
  
For example, if set to 5, the start-up number of connections in the pool is 5 in addition to the current existing connection.  
  
If set to 0, no connections are opened in addition to the current existing connection.  
Default 0  
GUI Tab [Pooling tab](#) on page 382

Packet Size

Attribute PacketSize (PS)  
Description Determines the number of bytes for each database protocol packet that is transferred from the database server to the client machine. Adjusting the packet size can improve performance. The optimal value depends on the typical size of data that is inserted, updated, or returned by the application and the environment in which it is running. Typically, larger packet sizes work better for large amounts of data. For example, if an application regularly returns character values that are 10,000 characters in length, using a value of 32 (16 KB) typically results in improved performance.  
  
NOTE: The ODBC connection option SQL\_PACKET\_SIZE provides the same functionality as the Packet Size option; however SQL\_PACKET\_SIZE and the Packet Size option are mutually exclusive. If Packet Size is specified, the driver returns the message `Driver Not Capable` if an application attempts to call SQL\_PACKET\_SIZE. If you do not set the Packet Size option, application calls to SQL\_PACKET\_SIZE are accepted by the driver.  
  
This connection option can affect performance. See ["Performance Considerations" on page 413](#) for details.  
Valid Values -1 | 0 | x  
  
If set to -1, the driver uses the maximum packet size that is set by the database server.  
  
If set to 0, the driver uses the default packet size that is used by the database server.



If set to  $x$ , an integer from 1 to 127, the driver uses a packet size that is a multiple of 512 bytes. For example, PacketSize=8 means to set the packet size to  $8 * 512$  bytes (4096 bytes).

NOTE: IF SSL encryption is used, the driver must use the packet size that is specified by the server. Any value set for this option or the SQL\_PACKET\_SIZE connect option is ignored if SSL encryption is used.

Default -1

GUI Tab [Advanced tab](#) on page 378

## Password

Attribute Password (PWD)

Description The password that the application uses to connect to your database. The Password option cannot be specified through the driver Setup dialog box and should not be stored in a data source. It is specified through the Logon dialog box or a connection string.

Valid Values *pwd*

where *pwd* is a valid password.

Default None

GUI Tab n/a

## Port Number

Attribute PortNumber (PORT)

Description The port number of the server listener.

Valid Values *port\_name*

where the *port\_name* is the port number of the server listener. Check with your database administrator for the correct number.

Default 1433

GUI Tab [General tab](#) on page 377

## Query Timeout

Attribute QueryTimeout (QT)

Description The number of seconds for the default query timeout for all statements that are created by a connection. To override the value set by this connection option for an individual statement, set a different value in the SQL\_ATTR\_QUERY\_TIMEOUT statement attribute on the SQLSetStmtAttr() function.

Valid Values -1 | 0 |  $x$

where  $x$  is a positive integer that specifies a number of seconds.

If set to -1, the query does not time out. The driver silently ignores the SQL\_ATTR\_QUERY\_TIMEOUT attribute.

If set to 0, the query does not time out, but the driver responds to the SQL\_ATTR\_QUERY\_TIMEOUT attribute.

If set to *x*, all queries time out after the specified number of seconds unless the application overrides this value by setting the SQL\_ATTR\_QUERY\_TIMEOUT attribute.

Default 0

GUI Tab [Advanced tab](#) on page 378

**Record Delimiter**

Attribute BulkLoadRecordDelimiter (BLRD)

Description Specifies the character that the driver will use to delimit the record entries in a bulk load data file.

Valid Values *x*

where *x* is any printable character.

For simplicity, avoid using a value that can be in the data, including all alphanumeric characters, the dash(-), the colon(:), the period (.), the forward slash (/), the space character, the single quote (') and the double quote ("). You can use some of these characters as delimiters if all of the data in the file is contained within double quotes.

NOTE: The Bulk Load Record Delimiter character must be different from the Bulk Load Field Delimiter.

Default None

GUI Tab [Bulk tab](#) on page 383

**Report Codepage Conversion Errors**

Attribute ReportCodepageConversionErrors (RCCE)

Description Specifies how the driver handles code page conversion errors that occur when a character cannot be converted from one character set to another.

An error message or warning can occur if an ODBC call causes a conversion error, or if an error occurs during code page conversions to and from the database or to and from the application. The error or warning generated is `Code page conversion error encountered`. In the case of parameter data conversion errors, the driver adds the following sentence:  
`Error in parameter x, where x is the parameter number.` The standard rules for returning specific row and column errors for bulk operations apply.

Valid Values 0 | 1 | 2

If set to 0 (Ignore Errors), the driver substitutes 0x1A for each character that cannot be converted and does not return a warning or error.

If set to 1 (Return Error), the driver returns an error instead of substituting 0x1A for unconverted characters.

If set to 2 (Return Warning), the driver substitutes 0x1A for each character that cannot be converted and returns a warning.

Default 0 (Ignore Errors)  
 GUI Tab [Advanced tab](#) on page 378

## Truststore

Attribute Truststore (TS)  
 Description The directory that contains the truststore file and the truststore file name to be used when SSL is enabled (Encryption Method=1) and server authentication is used. The truststore file contains a list of the valid Certificate Authorities (CAs) that are trusted by the client machine for SSL server authentication. If you do not specify a directory, the current directory is used.

NOTE: The truststore and keystore files may be the same file.

Valid Values *truststore\_directory\filename*

where *truststore\_directory* is the directory where the truststore file is located and *filename* is the file name of the truststore file.

Default None  
 GUI Tab [Security tab](#) on page 379

## Truststore Password

Attribute TruststorePassword (TSP)  
 Description The password that is used to access the truststore file when SSL is enabled (Encryption Method=1) and server authentication is used. The truststore file contains a list of the Certificate Authorities (CAs) that the client trusts.

NOTE: The truststore and keystore files may be the same file; therefore, they may have the same password.

Valid Values *truststore\_password*

where *truststore\_password* is a valid password for the truststore file.

Default None  
 GUI Tab [Security tab](#) on page 379

## Use Snapshot Transactions

Attribute SnapshotSerializable (SS)  
 Description Allows your application to use the snapshot isolation level if your Microsoft SQL Server database is configured for Snapshot isolation. Supported only for Microsoft SQL Server 2005 and higher.

See ["Using The Snapshot Isolation Level" on page 419](#) for details about using the snapshot isolation level.

This connection option can affect performance. See ["Performance Considerations" on page 413](#) for details.

Valid Values 0 | 1

When set to 1 (Enabled) and your application has the transaction isolation level set to serializable, the application uses the snapshot isolation level.

When set to 0 (Disabled) and your application has the transaction isolation level set to serializable, the application uses the serializable isolation level.

This option is useful for existing applications that set the isolation level to serializable. Using Snapshot Transactions in this case allows you to change to the snapshot isolation level with no or minimum code changes. If developing a new application, you can code it to set the connection attribute SQL\_COPT\_SS\_TXN\_ISOLATION to the value SQL\_TXN\_SS\_SNAPSHOT.

Default 0 (Disabled)

GUI Tab [Advanced tab](#) on page 378

### User Name

Attribute LogonID (UID)

Description The default user ID that is used to connect to your database. Your ODBC application may override this value or you may override it in the logon dialog box or connection string.

Valid Values *userid*

where *userid* is a valid user ID with permissions to access the database.

Default None

GUI Tab [Security tab](#) on page 379

### Validate Server Certificate

Attribute ValidateServerCertificate (VSC)

Description Determines whether the driver validates the certificate that is sent by the database server when SSL encryption is enabled (Encryption Method=1). When using SSL server authentication, any certificate sent by the server must be issued by a trusted Certificate Authority (CA). Allowing the driver to trust any certificate returned from the server even if the issuer is not a trusted CA is useful in test environments because it eliminates the need to specify truststore information on each client in the test environment.

Truststore information is specified using the Trust Store and Trust Store Password options.

Valid Values 0 | 1

If set to 1 (Enabled), the driver validates the certificate that is sent by the database server. Any certificate from the server must be issued by a trusted CA in the truststore file. If the Host Name In Certificate option is specified, the driver also validates the certificate using a host name. The Host Name In Certificate option provides additional security against man-in-the-middle (MITM) attacks by ensuring that the server the driver is connecting to is the server that was requested.

If set to 0 (Disabled), the driver does not validate the certificate that is sent by the database server. The driver ignores any truststore information specified by the Trust Store and Trust Store Password options.

Default 1 (Enabled)  
 GUI Tab [Security tab](#) on page 379

## Workstation ID

Attribute WorkstationID (WSID)  
 Description The workstation ID that is used by the client.  
 Valid Values *string*  
 where *string* is the workstation ID.  
 Default None  
 GUI Tab [Advanced tab](#) on page 378

## XML Describe Type

Attribute XMLDescribeType (XDT)  
 Description The SQL data type that is returned by SQLGetTypeInfo for the XML data type.  
 See ["Using the XML Data Type" on page 417](#) for further information about the XML data type.  
 Valid Values -4 | -10  
 If set to -4 (SQL\_LONGVARBINARY), the driver uses the description SQL\_LONGVARBINARY for columns that are defined as the XML data type.  
 If set to -10 (SQL\_WLONGVARCHAR), the driver uses the description SQL\_WLONGVARCHAR for columns that are defined as the XML data type.  
 Default -10  
 GUI Tab [Advanced tab](#) on page 378

---

# Performance Considerations

The following connection options can enhance driver performance. You can also enhance performance through efficient application design. Refer to [Chapter 5 "Designing ODBC Applications for Performance Optimization"](#) in the *DataDirect Connect Series for ODBC Reference* for details.

**Connection Pooling (ConnectionPooling):** If you enable the driver to use connection pooling, you can set additional options that affect performance:

- **Load Balance Timeout:** You can define how long to keep connections in the pool. The time that a connection was last used is compared to the current time and, if the

timespan exceeds the value of the Load Balance Timeout option, the connection is destroyed. The Min Pool Size option can cause some connections to ignore this value.

- **Connection Reset:** Resetting a re-used connection to the initial configuration settings impacts performance negatively because the connection must issue additional commands to the server.
- **Max Pool Size:** Setting the maximum number of connections that the pool can contain too low might cause delays while waiting for a connection to become available. Setting the number too high wastes resources.
- **Min Pool Size:** A connection pool is created when the first connection with a unique connection string connects to the database. The pool is populated with connections up to the minimum pool size, if one has been specified. The connection pool retains this number of connections, even when some connections exceed their Load Balance Timeout value.

**Enable Bulk Load (EnableBulkLoad):** If your application performs bulk loading of data, you can improve performance by configuring the driver to use the database system's bulk load functionality instead of database array binding. The trade-off to consider for improved performance is that using the bulk load functionality can bypass data integrity constraints.

**Encryption Method (EncryptionMethod):** Data encryption may adversely affect performance because of the additional overhead (mainly CPU usage) that is required to encrypt and decrypt data.

**Failover Mode (FailoverMode):** Although high availability that replays queries after a failure provides increased levels of protection, it can adversely affect performance because of increased overhead.

**Packet Size (PacketSize):** Typically, it is optimal for the client to use the maximum packet size that the database server allows. This reduces the total number of round trips required to return data to the client, thus improving performance. Therefore, performance can be improved if the PacketSize attribute is set to the maximum packet size of the server.

**Use Snapshot Transactions (SnapshotSerializable):** You must have your Microsoft SQL Server 2005 and higher database configured for snapshot isolation for this connection option to work. Snapshot Isolation provides transaction-level read consistency and an optimistic approach to data modifications by not acquiring locks on data until data is to be modified. This Microsoft SQL Server 2005 and higher feature can be useful if you want to consistently return the same result set even if another transaction has changed the data and 1) your application executes many read operations or 2) your application has long running transactions that could potentially block users from reading data. This feature has the potential to eliminate data contention between read operations and update operations. When this connection option is enabled, performance is improved due to increased concurrency.

See ["Using The Snapshot Isolation Level" on page 419](#) for details.

# Data Types

Table 11-2 shows how the Microsoft SQL Server and Windows Azure SQL Database data types are mapped to the standard ODBC data types. ["Unicode Support" on page 416](#) lists Microsoft SQL Server to Unicode data type mappings.

**Table 11-2. Microsoft SQL Server Data Types**

SQL Server	ODBC
binary	SQL_BINARY
bigint	SQL_BIGINT
bigint identity	SQL_BIGINT
bit	SQL_BIT
char	SQL_CHAR
date <sup>1</sup>	SQL_TYPE_DATE
datetime	SQL_TYPE_TIMESTAMP
datetime2 <sup>1</sup>	SQL_TYPE_TIMESTAMP
decimal	SQL_DECIMAL
decimal() identity	SQL_DECIMAL
float	SQL_FLOAT
image	SQL_LONGVARBINARY
int	SQL_INTEGER
int identity	SQL_INTEGER
money	SQL_DECIMAL
numeric	SQL_NUMERIC
numeric() identity	SQL_NUMERIC
real	SQL_REAL
smalldatetime	SQL_TYPE_TIMESTAMP
smallint	SQL_SMALLINT
smallint identity	SQL_SMALLINT
smallmoney	SQL_DECIMAL
text	SQL_LONGVARCHAR
time <sup>1 2</sup>	SQL_TYPE_TIMESTAMP
timestamp	SQL_BINARY
tinyint	SQL_TINYINT
tinyint identity	SQL_TINYINT
uniqueidentifier	SQL_GUID
varbinary	SQL_VARBINARY
varbinary(max) <sup>3</sup>	SQL_LONGVARBINARY

**Table 11-2. Microsoft SQL Server Data Types** (cont.)

SQL Server	ODBC
varchar	SQL_VARCHAR
varchar(max) <sup>3</sup>	SQL_LONGVARCHAR

1. Supported only on Microsoft SQL Server 2008 and higher.  
2. Time mapping changes based on the setting of the Fetch TWFS as Time option.  
3. Supported only on Microsoft SQL Server 2005 and higher.

See ["Retrieving Data Type Information" on page 59](#) for information about retrieving data types.

# Unicode Support

The SQL Server Wire Protocol driver maps the Microsoft SQL Server and Windows Azure SQL Database data types to Unicode data types as shown in the following table:

**Table 11-3. Mapping Microsoft SQL Server and Windows Azure SQL Database Data Types to Unicode Data Types**

SQL Server Data Type	Mapped to. . .
datetimeoffset <sup>1, 2</sup>	SQL_WVARCHAR
nchar	SQL_WCHAR
ntext	SQL_WLONGVARCHAR
nvarchar	SQL_WVARCHAR
nvarchar(max) <sup>3</sup>	SQL_WLONGVARCHAR
sysname	SQL_WVARCHAR
xml <sup>3</sup>	SQL_WLONGVARCHAR

1. Supported only for Microsoft SQL Server 2008 and higher, and Windows Azure SQL Database.  
2. Datetimeoffset mapping changes based on the setting of the Fetch TSWTZ as Timestamp option.  
3. nvarchar(max) and xml are supported for Microsoft SQL Server 2005 and higher.

The driver supports the Unicode ODBC W (Wide) function calls, such as SQLConnectW. This allows the Driver Manager to transmit these calls directly to the driver. Otherwise, the Driver Manager would incur the additional overhead of converting the W calls to ANSI function calls, and vice versa.

See ["UTF-16 Applications on UNIX and Linux" on page 125](#) for related details. Also, refer to [Chapter 4 "Internationalization, Localization, and Unicode"](#) in the *DataDirect Connect Series for ODBC Reference* for a more detailed explanation of Unicode.



## Using the XML Data Type

By default, Microsoft SQL Server returns XML data to the driver encoded as UTF-8. To avoid data loss, an application must bind XML data as SQL\_C\_WCHAR. The driver then returns the data as either UTF-8 or UTF-16, depending on platform and application settings. If the application binds XML data as SQL\_C\_CHAR, the driver converts it to the client character encoding, possibly causing data loss or corruption. To prevent any conversion of XML data, the application must set the attribute [XML Describe Type](#) to SQL\_LONGVARIABLE (-10) and bind the data as SQL\_C\_BINARY.

---

## Advanced Features

The driver supports the following advanced features:

- Failover
- Security
- Connection Pooling
- DataDirect Bulk Load

### Failover

The driver supports failover and its related connection options. Failover connection options are located on the [Failover tab](#) of the driver Setup dialog box. See ["Using Failover" on page 65](#) for a general description of failover and its implementation.

### Security

The driver supports authentication and encryption. Security connection options are located on the [Security tab](#) of the driver Setup dialog box. See ["Using Security" on page 77](#) for a general description of security and its implementation.

### Authentication

If you are using Kerberos, verify that your environment meets the requirements listed in [Table 11-4](#) before you configure the driver for Kerberos authentication.<sup>1</sup>

---

1. Not supported for Microsoft Windows Azure for SQL Database. You must provide credentials every time when you connect to SQL Database.

**Table 11-4. Kerberos Authentication Requirements for the SQL Server Wire Protocol Driver**

Component	Requirements
Microsoft SQL Server database server	<p>The database server must be administered by the same domain controller that administers the client and must be running one of the following databases:</p> <ul style="list-style-type: none"> <li>■ Microsoft SQL Server 2012</li> <li>■ Microsoft SQL Server 2008</li> <li>■ Microsoft SQL Server 2005</li> <li>■ Microsoft SQL Server 2000</li> <li>■ Microsoft SQL Server 2000 Enterprise Edition (64-bit) Service Pack 2 or higher</li> </ul>
Kerberos server	<p>The Kerberos server is the machine where the user IDs for authentication are administered. The Kerberos server is also the location of the Kerberos KDC.</p> <p>Network authentication must be provided by Windows Active Directory on one of the following operating systems:</p> <ul style="list-style-type: none"> <li>■ Windows Server 2003</li> <li>■ Windows 2000 Server Service Pack 3 or higher</li> </ul>
Client	The client must be administered by the same domain controller that administers the database server.

## Connection Pooling

The driver supports connection pooling and its related connection options. Connection pooling connection options are located on the [Pooling tab](#) of the driver Setup dialog box. See ["Using DataDirect Connection Pooling" on page 82](#) for a general description of connection pooling and its implementation.

## DataDirect Bulk Load

The driver supports DataDirect bulk load and its related connection options. Bulk load connection options are located on the [Bulk tab](#) of the driver Setup dialog box. See ["Using DataDirect Bulk Load" on page 85](#) for a general description of DataDirect bulk load and its implementation.

For optimal performance, you must enable minimal logging and Table Locking. Please refer to the following Web site for more information on minimal logging:

<http://msdn.microsoft.com/en-us/library/ms190422.aspx>

Table Locking, one of the Bulk Options, is enabled by default. This prevents other transactions from accessing the table during bulk load. See ["Bulk Options" on page 396](#) for details about this option.

---

## Persisting a Result Set as an XML Data File

The driver allows you to persist a result as an XML data file with embedded schema. See ["Persisting a Result Set as an XML Data File" on page 60](#) for details about implementation.

---

## Isolation and Lock Levels Supported

Microsoft SQL Server supports isolation levels 0 (Read Uncommitted), 1 (Read Committed), 2 (Repeatable Read), and 3 (Serializable). Microsoft SQL Server supports row-level and table-level locking. Refer to [Chapter 7 "Locking and Isolation Levels"](#) in the *DataDirect Connect Series for ODBC Reference* for details.

Microsoft SQL Server 2005 and higher supports the following additional isolation levels:

- Snapshot
- Read Committed with Snapshots
- Read Committed with Locks (equivalent to Read Committed in previous Microsoft SQL Server versions)

### Using The Snapshot Isolation Level

The Snapshot isolation level is available only with Microsoft SQL Server 2005 and higher. Setting the SnapshotSerializable connection string attribute changes the behavior of the Serializable isolation level to use the Snapshot Isolation level. This allows an application to use the Snapshot Isolation level with minimal or no code changes.

If you are writing a new application, you may want to code it to set the connection attribute SQL\_COPT\_SS\_TXN\_ISOLATION to the value SQL\_TXN\_SS\_SNAPSHOT. The application then uses the snapshot isolation level without requiring the Use Snapshot Transactions connection option.

See ["Use Snapshot Transactions" on page 411](#) for additional information.

---

## SQL Support

The driver supports the core SQL grammar.

---

## ODBC Conformance Level

The driver supports ODBC conformance level 1.

In addition, the following functions are supported:

- SQLForeignKeys
- SQLTablePrivileges
- SQLDescribeParam
- SQLColumnPrivileges

Refer to [Chapter 2 “ODBC API and Scalar Functions”](#) in the *DataDirect Connect Series for ODBC Reference* for a list of the API functions supported by the SQL Server Wire Protocol driver.

---

## Number of Connections and Statements Supported

The SQL Server Wire Protocol driver supports multiple connections and multiple statements per connection.

---

## Using Arrays of Parameters

Microsoft SQL Server databases natively support parameter arrays, and the SQL Server Wire Protocol driver, in turn, supports them. When designing an application for performance, using native parameter arrays for bulk inserts or updates, for example, can improve performance. Refer to [Chapter 5 “Designing ODBC Applications for Performance Optimization”](#) in the *DataDirect Connect Series for ODBC Reference* for more information about using arrays of parameters to improve performance.

# 12 The Sybase Wire Protocol Driver

The DataDirect Connect *for* ODBC and DataDirect Connect64 *for* ODBC Sybase Wire Protocol driver (the Sybase Wire Protocol driver) each support the following Sybase database servers:

- Sybase Adaptive Server 15, 15.5, and 15.7
- Sybase Adaptive Server 12.0, 12.5, and 12.5.x
- Sybase Adaptive Server 11.9

The Sybase Wire Protocol driver is supported in the Windows, UNIX, and Linux environments. See ["Environment-Specific Information" on page 45](#) for detailed information about the environments supported by this driver.

Refer to the readme file shipped with your DataDirect product for the file name of the Sybase Wire Protocol driver.

---

## Driver Requirements

The driver has no client requirements.

---

## Configuring and Connecting to Data Sources

After you install the driver, you configure data sources to connect to the database. See [Chapter 1 "Quick Start Connect" on page 35](#) for an explanation of different types of data sources. The data source contains connection options that allow you to tune the driver for specific performance. If you want to use a data source but need to change some of its values, you can either modify the data source or override its values at connection time through a connection string.

If you choose to use a connection string, you must use specific connection string attributes. See ["Using a Connection String" on page 436](#) and [Table 12-1 on page 438](#) for an alphabetical list of driver connection string attributes and their initial default values.

### Data Source Configuration in the UNIX/Linux odbc.ini File

On UNIX and Linux, you must set up the proper ODBC environment before configuring data sources. See ["Environment Configuration" on page 37](#) for basic setup information and ["Environment Variables" on page 97](#) for more detail about this procedure.

Data sources for UNIX and Linux are stored in the system information file (by default, odbc.ini). If you have a Motif GUI environment on Linux, you can configure and modify data sources through the DataDirect ODBC Data Source Administrator for Linux (the Linux

ODBC Administrator) using a driver Setup dialog box. (See ["Configuration Through the Administrator" on page 100](#) for a detailed explanation of the Administrator.)

If you do not have a GUI environment, you can configure and modify data sources directly by editing the `odbc.ini` file and storing default connection values there. See ["Configuration Through the System Information \(odbc.ini\) File" on page 103](#) for detailed information about the specific steps necessary to configure a data source.

[Table 12-1 on page 438](#) lists driver connection string attributes that must be used in the `odbc.ini` file to set the value of the attributes. Note that only the long name of the attribute can be used in the file. The default listed in the table is the initial default value when the driver is installed.

## Data Source Configuration through a GUI

On Windows, data sources are stored in the Windows Registry. You can configure and modify data sources through the ODBC Administrator using a driver Setup dialog box, as described in this section.

On UNIX and Linux, data sources are stored in the `odbc.ini` file. On Linux, you can configure and modify data sources through the Linux ODBC Administrator using a driver Setup dialog box, as described in this section.

**NOTE:** This book shows dialog box images that are specific to Windows. If you are using the drivers in the Linux environment, the dialog box that you see may differ slightly from the Windows version. Windows-only and UNIX-only connection options are specifically noted by icons in the Setup dialog box descriptions.

When the driver is first installed, the values of its connection options are set by default. These values appear on the driver Setup dialog box tabs when you create a new data source. You can change these default values by modifying the data source. In the following procedure, the description of each tab is followed by a table that lists the connection options for that tab and their initial default values. This table links you to a complete description of the options and their connection string attribute equivalents. The connection string attributes are used to override the default values of the data source if you want to change these values at connection time.

### To configure a Sybase data source:

- 1 Start the ODBC Administrator:
  - On Windows, start the ODBC Administrator by selecting its icon from the DataDirect Connect program group.
  - On Linux, change to the `install_dir/tools` directory and, at a command prompt, enter:
 

```
odbcadmin
```

 where `install_dir` is the path to the product installation directory.
- 2 Select a tab:
  - **User DSN:** If you are configuring an existing user data source, select the data source name and click **Configure** to display the driver Setup dialog box.

If you are configuring a new user data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.

- **System DSN:** If you are configuring an existing system data source, select the data source name and click **Configure** to display the driver Setup dialog box.

If you are configuring a new system data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.

- **File DSN:** If you are configuring an existing file data source, select the data source file and click **Configure** to display the driver Setup dialog box.

If you are configuring a new file data source, click **Add** to display a list of installed drivers; then, select a driver. Click **Advanced** if you want to specify attributes; otherwise, click **Next** to proceed. Specify a name for the data source and click **Next**. Verify the data source information; then, click **Finish** to display the driver Setup dialog box.

The General tab of the Setup dialog box appears by default.

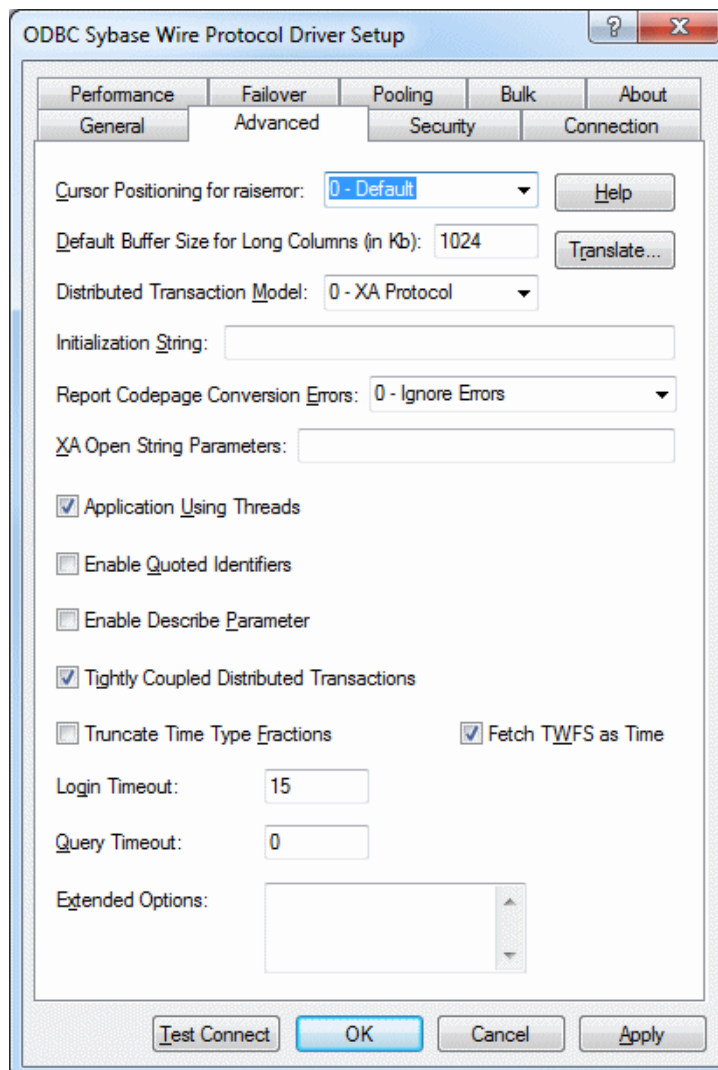
The screenshot shows the 'ODBC Sybase Wire Protocol Driver Setup' dialog box with the 'General' tab selected. The dialog has a title bar with a question mark and a close button. Below the title bar are five tabs: 'Performance', 'Failover', 'Pooling', 'Bulk', and 'About'. The 'General' tab is active, showing a 'Data Source Name' field with a 'Help' button, a 'Description' field, a 'Network Address' field, and a 'Database Name' field. Below these is a section titled 'Use Interfaces File for Connection Information (Optional)' containing 'Interfaces File' and 'Server Name' fields. At the bottom are 'Test Connect', 'OK', 'Cancel', and 'Apply' buttons.

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

- 3 On this tab, provide values for the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: General	Default
<a href="#">Data Source Name</a> (see page 446)	None
<a href="#">Description</a> (see page 446)	None
<a href="#">Network Address</a> (see page 457)	None
<a href="#">Database Name</a> (see page 446)	None
<a href="#">Interfaces File</a> (see page 455)	None
<a href="#">Server Name</a> (see page 461)	None

- 4 Optionally, click the **Advanced** tab to specify additional data source settings.



On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.



Connection Options: Advanced	Default
<a href="#">Cursor Positioning for Raiserror (see page 445)</a>	0 - Default
<a href="#">Default Buffer Size for Long/LOB Columns (in Kb) (see page 447)</a>	1024
<a href="#">Distributed Transaction Model (see page 447)</a>	0 - XA Protocol
<a href="#">Initialization String (see page 454)</a>	None
<a href="#">Report Codepage Conversion Errors (see page 460)</a>	0 - Ignore Errors
<a href="#">XA Open String Parameters (see page 464)</a>	None
<a href="#">Application Using Threads (see page 441)</a>	Enabled
<a href="#">Enable Quoted Identifiers (see page 448)</a>	Disabled
<a href="#">Enable Describe Parameter (see page 448)</a>	Disabled
<a href="#">Tightly Coupled Distributed Transactions (see page 462)</a>	Enabled
<a href="#">Truncate Time Type Fractions (see page 462)</a>	Disabled
<a href="#">Fetch TWFS as Time (see page 451)</a>	Enabled
<a href="#">Login Timeout (see page 456)</a>	15
<a href="#">Query Timeout (see page 459)</a>	0
<a href="#">IANAAppCodePage (see page 454)</a>	4 (ISO 8559-1 Latin-1)
UNIX ONLY	

**Extended Options:** Type a semi-colon separated list of connection options and their values. Use this configuration option to set the value of undocumented connection options that are provided by Progress DataDirect customer support. You can include any valid connection option in the Extended Options string, for example:

```
Database=Server1;UndocumentedOption1=value[;UndocumentedOption2=value;]
```

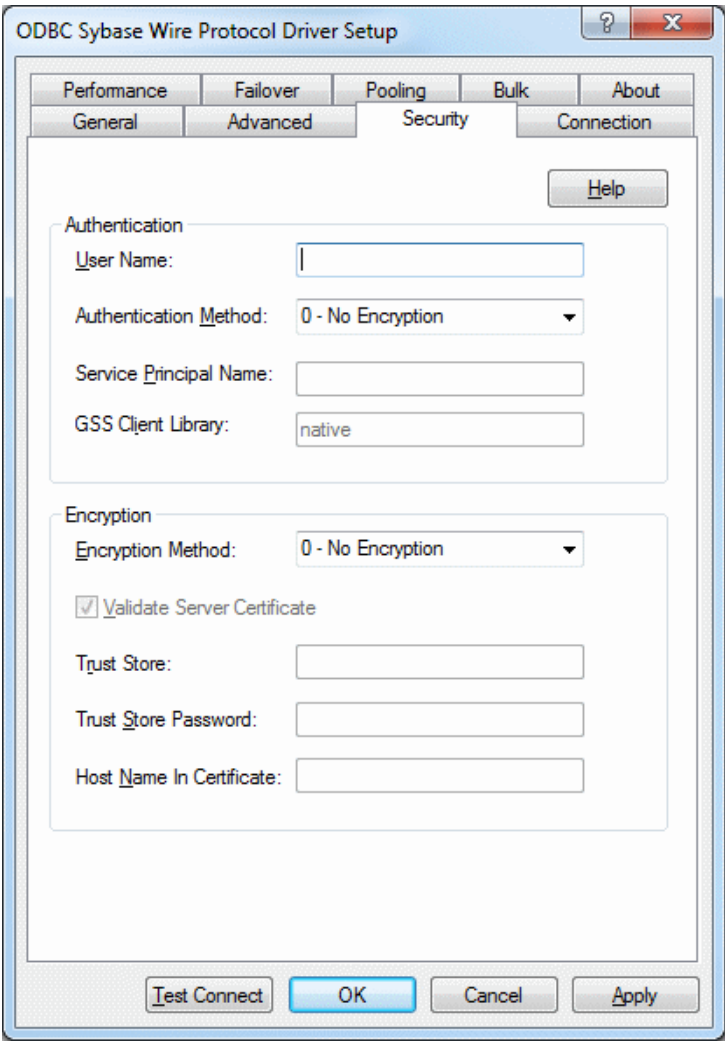
If the Extended Options string contains option values that are also set in the setup dialog or data source, the values of the options specified in the Extended Options string take precedence. However, connection options that are specified on a connection string override any option value specified in the Extended Options string.

NOTE: Do not specify the Extended Options configuration option in a connection string, or the driver will return an error. Instead, applications should specify the individual undocumented connection options in the connection string.

**Translate:** Click **Translate** to display the Select Translator dialog box, which lists the translators specified in the ODBC Translators section of the Registry. Progress 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.

5 Optionally, click the **Security** tab to specify security data source settings.

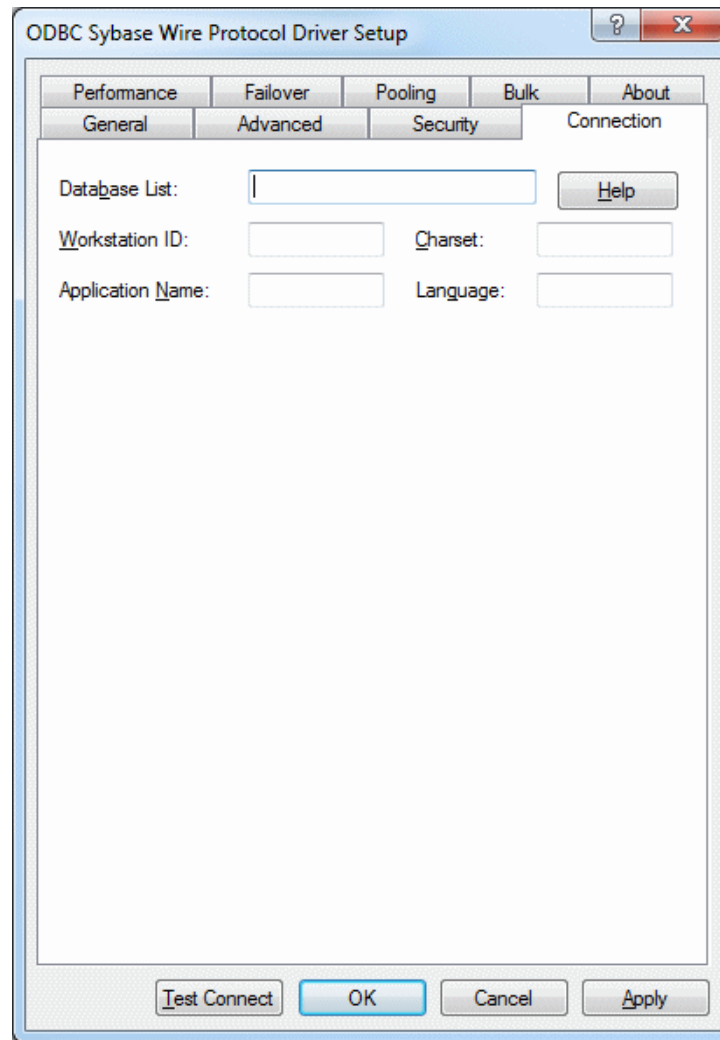


See ["Using Security" on page 77](#) for a general description of authentication and encryption and their configuration requirements.

On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Security	Default
<a href="#">User Name (see page 463)</a>	None
<a href="#">Authentication Method (see page 441)</a>	0 - No Encryption
<a href="#">Service Principal Name (see page 461)</a>	None
<a href="#">GSS Client Library (see page 452)</a>	native
<a href="#">Encryption Method (see page 448)</a>	0 - No Encryption
<a href="#">Validate Server Certificate (see page 463)</a>	Enabled
<a href="#">Truststore (see page 462)</a>	None
<a href="#">Truststore Password (see page 463)</a>	None
<a href="#">Host Name In Certificate (see page 453)</a>	None

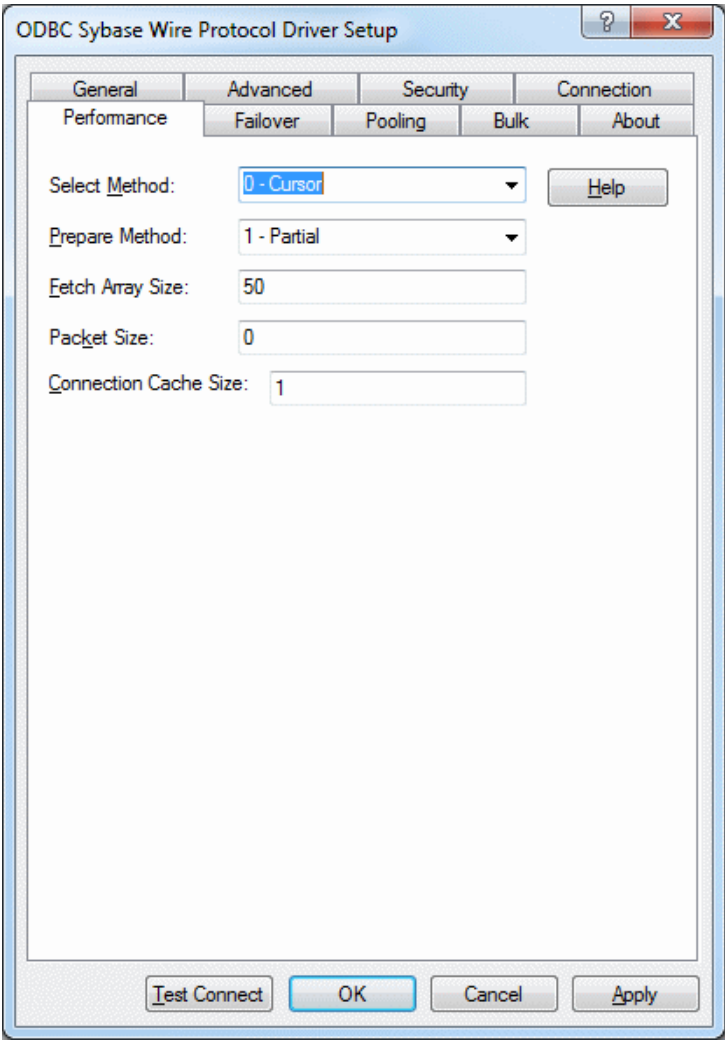
- 6 Optionally, click the **Connection** tab to specify data source settings.



On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Connection	Default
<a href="#">Database List (see page 446)</a>	None
<a href="#">Workstation ID (see page 464)</a>	None
<a href="#">Charset (see page 443)</a>	None
<a href="#">Application Name (see page 440)</a>	None
<a href="#">Language (see page 455)</a>	None

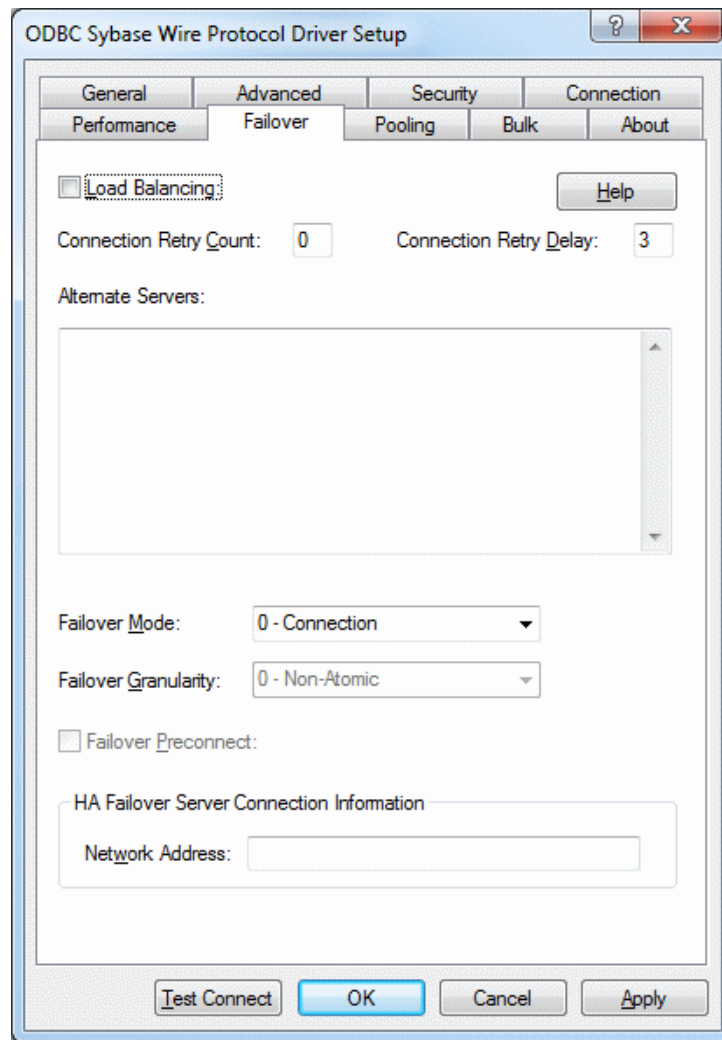
7 Optionally, click the **Performance** tab to specify performance data source settings.



On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Performance	Default
<a href="#">Select Method (see page 461)</a>	0 - Cursor
<a href="#">Prepare Method (see page 459)</a>	1 - Partial
<a href="#">Fetch Array Size (see page 451)</a>	50
<a href="#">Packet Size (see page 458)</a>	0
<a href="#">Connection Cache Size (see page 443)</a>	1

- 8 Optionally, click the **Failover** tab to specify failover data source settings.



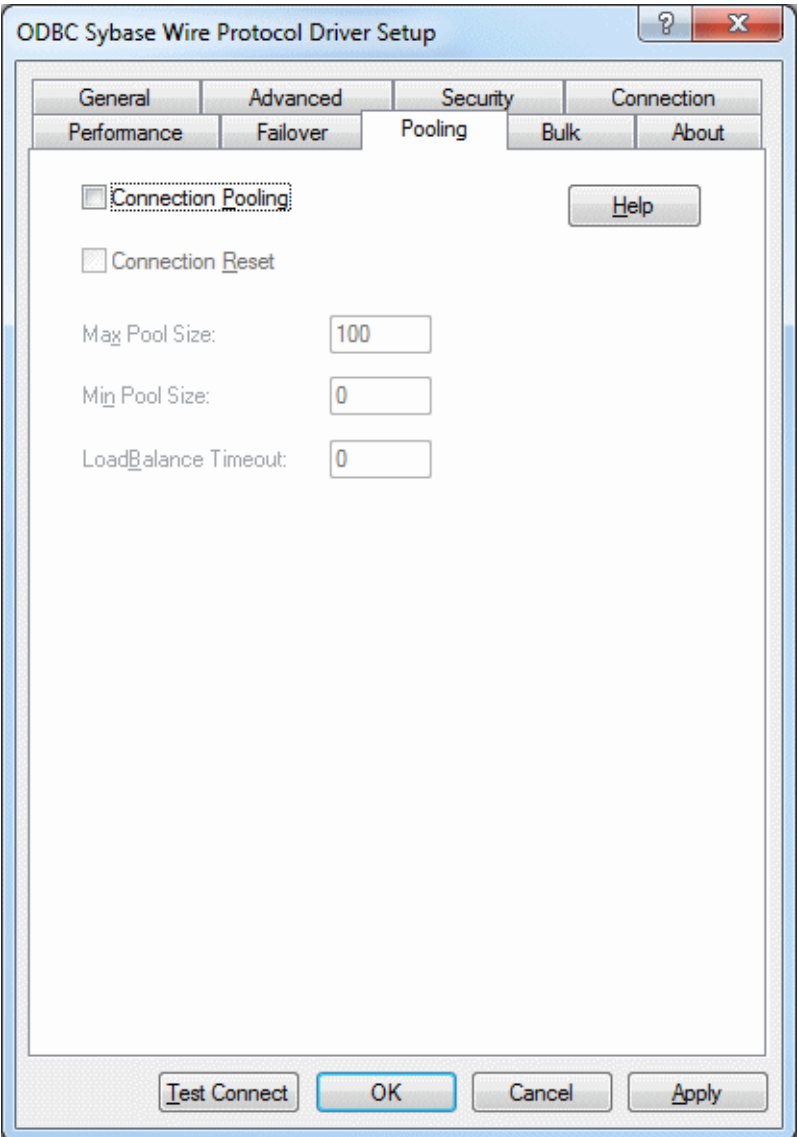
See ["Using Failover" on page 65](#) for a general description of failover and its related connection options.

On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

#### Connection Options: Failover

	Default
<a href="#">Load Balancing (see page 456)</a>	Disabled
<a href="#">Connection Retry Count (see page 444)</a>	0
<a href="#">Connection Retry Delay (see page 445)</a>	3
<a href="#">Alternate Servers (see page 440)</a>	None
<a href="#">Failover Mode (see page 450)</a>	0 - Connection
<a href="#">Failover Granularity (see page 449)</a>	0 - Non-Atomic
<a href="#">Failover Preconnect (see page 450)</a>	Disabled
<a href="#">HA Failover Server Connection Information/Network Address (see page 452)</a>	None

- 9 Optionally, click the **Pooling** tab to specify connection pooling data source settings.

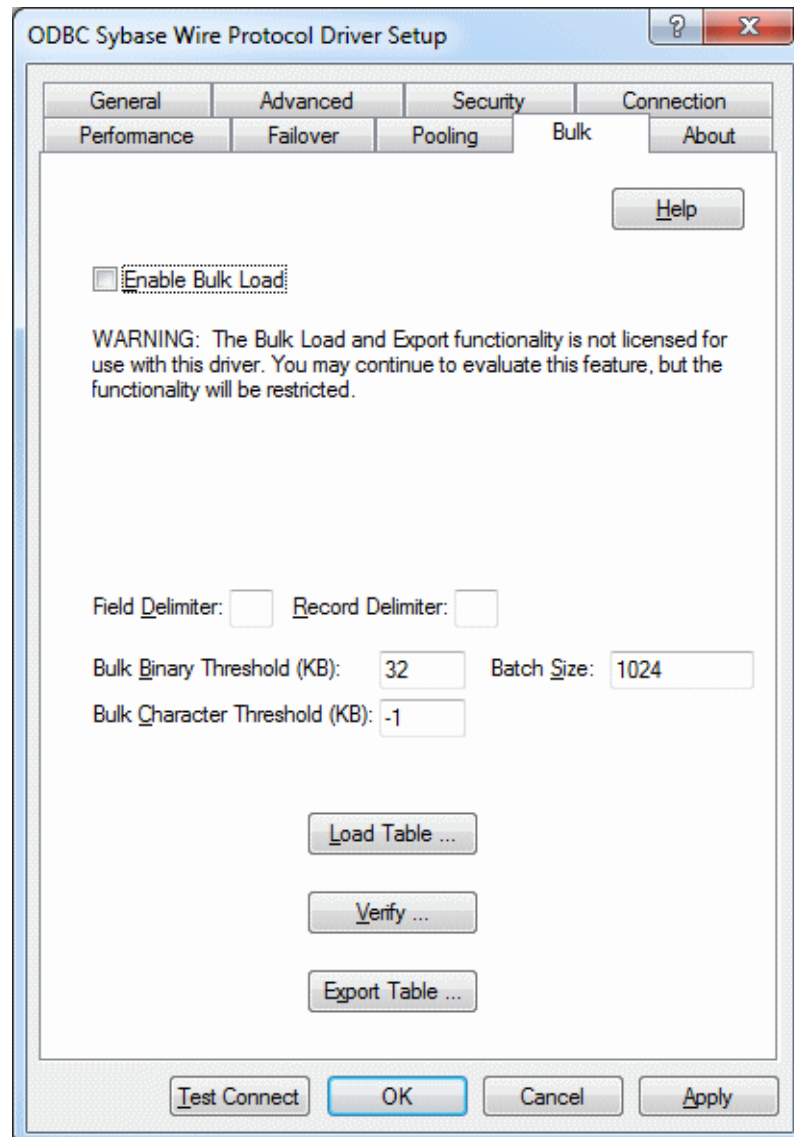


See ["Using DataDirect Connection Pooling" on page 82](#) for a general description of connection pooling.

On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Pooling	Default
<a href="#">Connection Pooling (see page 444)</a>	Disabled
<a href="#">Connection Reset (see page 444)</a>	Disabled
<a href="#">Max Pool Size (see page 457)</a>	100
<a href="#">Min Pool Size (see page 457)</a>	0
<a href="#">Load Balance Timeout (see page 455)</a>	0

- 10 Optionally, click the **Bulk** tab to specify DataDirect Bulk Load data source settings.



See ["Using DataDirect Bulk Load" on page 85](#) for a general description of DataDirect Bulk Load.

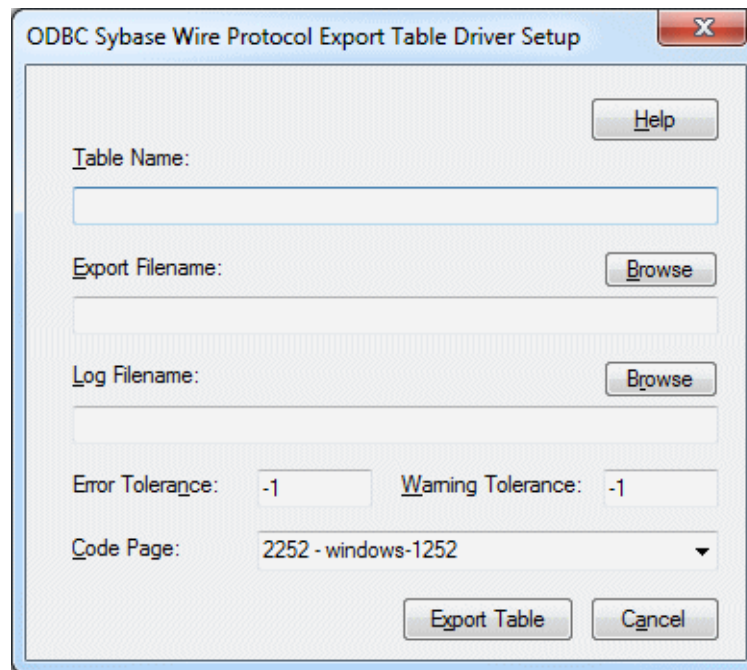
On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Bulk	Default
<a href="#">Enable Bulk Load (see page 447)</a>	Disabled
<a href="#">Field Delimiter (see page 451)</a>	None
<a href="#">Record Delimiter (see page 460)</a>	None
<a href="#">Bulk Binary Threshold (see page 442)</a>	32
<a href="#">Bulk Character Threshold (see page 442)</a>	-1
<a href="#">Batch Size (see page 442)</a>	1024

If your application is already coded to use parameter array batch functionality, you can leverage DataDirect Bulk Load features through the Enable Bulk Load connection option. Enabling this option automatically converts the parameter array batch operation to use the database bulk load protocol.

If you are not using parameter array batch functionality, you can export data to a bulk load data file, verify the metadata of the bulk load configuration file against the structure of the target table, and bulk load data to a table. Use the following steps to accomplish these tasks.

- a To export data from a table to a bulk load data file, click **Export Table** from the Bulk tab. The Export Table dialog box appears.



**Table Name:** A string that specifies the name of the source database table containing the data to be exported.

**Export Filename:** A string that specifies the path (relative or absolute) and file of the bulk load data file to which the data is to be exported. It also specifies the file name of the bulk configuration file. These files must not already exist; if one of both of them already exists, an error is returned.

**Log Filename:** A string that specifies the path (relative or absolute) and file name of the bulk log file. The log file is created if it does not exist. Events logged to this file are:

- Total number of rows fetched
- A message for each row that failed to export
- Total number of rows that failed to export
- Total number of rows successfully exported

Information about the load is written to this file, preceded by a header. Information about the next load is appended to the end of the file.

If you do not supply a value for Log Filename, no log file is created.



**Error Tolerance:** A value that specifies the number of errors to tolerate before an operation terminates. A value of 0 indicates that no errors are tolerated; the operation fails when the first error is encountered.

The default of -1 means that an infinite number of errors is tolerated.

**Warning Tolerance:** A value that specifies the number of warnings to tolerate before an operation terminates. A value of 0 indicates that no warnings are tolerated; the operation fails when the first warning is encountered.

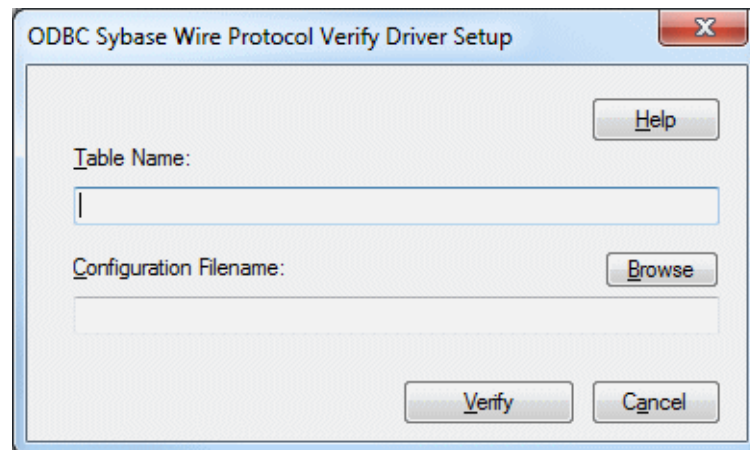
The default of -1 means that an infinite number of warnings is tolerated.

**Code Page:** A value that specifies the code page value to which the driver must convert all data for storage in the bulk data file. See ["Character Set Conversions" on page 94](#) for more information.

The default value on Windows is the current code page of the machine. On UNIX/Linux, the default value is 4 (ISO 8559-1 Latin-1).

Click **Export Table** to connect to the database and export data to the bulk data file or click **Cancel**.

- b To verify the metadata of the bulk load configuration file against the structure of the target database table, click **Verify** from the Bulk tab. See ["Verification of the Bulk Load Configuration File" on page 92](#) for details. The Verify dialog box appears.



**Table Name:** A string that specifies the name of the target database table into which the data is to be loaded.

**Configuration Filename:** A string that specifies the path (relative or absolute) and file name of the bulk configuration file.

Click **Verify** to verify table structure or click **Cancel**.

- c To bulk load data from the bulk data file to a database table, click **Load Table** from the Bulk tab. The Load File dialog box appears.

**Table Name:** A string that specifies the name of the target database table into which the data is loaded.

**Load Data Filename:** A string that specifies the path (relative or absolute) and file name of the bulk data file from which the data is loaded.

**Configuration Filename:** A string that specifies the path (relative or absolute) and file name of the bulk configuration file.

**Log Filename:** A string that specifies the path (relative or absolute) and file name of the bulk log file. Specifying a value for Log Filename creates the file if it does not already exist. Events logged to this file are:

- Total number of rows read
- Message for each row that failed to load
- Total number of rows that failed to load
- Total number of rows successfully loaded

Information about the load is written to this file, preceded by a header. Information about the next load is appended to the end of the file.

If you do not specify a value for Log Filename, no log file is created.

**Discard Filename:** A string that specifies the path (relative or absolute) and file name of the bulk discard file. Any row that cannot be inserted into database as result of bulk load is added to this file, with the last row rejected added to the end of the file.

Information about the load is written to this file, preceded by a header. Information about the next load is appended to the end of the file.

If you do not specify a value for Discard Filename, a discard file is not created.

**Error Tolerance:** A value that specifies the number of errors to tolerate before an operation terminates. A value of 0 indicates that no errors are tolerated; the operation fails when the first error is encountered.

The default of -1 means that an infinite number of errors is tolerated.

**Load Start:** A value that specifies the first row to be loaded from the data file. Rows are numbered starting with 1. For example, when Load Start is 10, the first 9 rows of the file are skipped and the first row loaded is row 10. This option can be used to restart a load after a failure.

The default value is 1.

**Read Buffer Size (KB):** A value that specifies the size, in KB, of the buffer that is used to read the bulk data file for a bulk load operation.

The default value is 2048.

**Warning Tolerance:** A value that specifies the number of warnings to tolerate before an operation terminates. A value of 0 indicates that no warnings are tolerated; the operation fails when the first warning is encountered.

The default of -1 means that an infinite number of warnings is tolerated.

**Load Count:** A value that specifies the number of rows to be loaded from the data file. The bulk load operation loads rows up to the value of Load Count from the file to the database. It is valid for Load Count to specify more rows than exist in the data file. The bulk load operation completes successfully when either the number of rows specified by the Load Count value has been loaded or the end of the data file is reached. This option can be used in conjunction with Load Start to restart a load after a failure.

The default value is the maximum value for SQLULEN. If set to 0, no rows are loaded.

Click **Load Table** to connect to the database and load the table or click **Cancel**.

- 11 At any point during the configuration process, you can click **Test Connect** to attempt to connect to the data source using the connection options specified in the driver Setup dialog box. A logon dialog box appears (see ["Using a Logon Dialog Box" on page 437](#) for details). 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 incorrect environment or connection value, it displays an appropriate error message. Click **OK**.

NOTE: If you are configuring alternate servers for use with the connection failover feature, be aware that the Test Connect button tests only the primary server, not the alternate servers.

- 12** 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.

## Using a Connection String

If you want to use a connection string for connecting to a database, or if your application requires it, you must specify either a DSN (data source name), a File DSN, or a DSN-less connection in the string. The difference is whether you use the DSN=, FILEDSN=, or the DRIVER= keyword in the connection string, as described in the ODBC specification. A DSN or FILEDSN connection string tells the driver where to find the default connection information. Optionally, you may specify *attribute=value* pairs in the connection string to override the default values stored in the data source.

The DSN connection string has the form:

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

The FILEDSN connection string has the form:

```
FILEDSN=filename.dsn[;attribute=value[;attribute=value]...]
```

The DSN-less connection string specifies a driver instead of a data source. All connection information must be entered in the connection string because the information is not stored in a data source.

The DSN-less connection string has the form:

```
DRIVER=[{ }driver_name[ } ][;attribute=value[;attribute=value]...]
```

[Table 12-1](#) lists the long and short names for each attribute, as well as the initial default value when the driver is first installed. You can specify either long or short names in the connection string.

An example of a DSN connection string with overriding attribute values for Sybase is:

```
DSN=SYB TABLES;DB=PAYROLL;UID=JOHN;PWD=XYZZY
```

A FILEDSN connection string is similar except for the initial keyword:

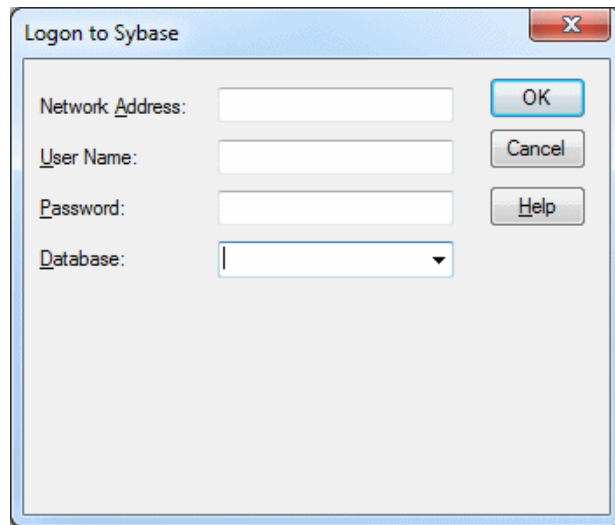
```
FILEDSN=SYB.dsn;DB=PAYROLL;UID=JOHN;PWD=XYZZY
```

A DSN-less connection string must provide all necessary connection information:

```
DRIVER=DataDirect 7.1 Sybase Wire Protocol;NA=123.456.78.90, 5000;DB=SYBACCT;  
UID=JOHN;PWD=XYZZY
```

## 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 the Logon dialog box, provide the following information:

- 1 In the Network Address field, specify an IP address for the Sybase server as follows: *IP address,port\_number*. For example, you might enter *199.226.224.34,5000*. If your network supports named servers, you can specify an address as: *servername,port\_number*. For example, you might enter *Sybaseserver,5000*.

The IP address can be specified in either IPv4 or IPv6 format, or a combination of the two. See ["Using IP Addresses" on page 54](#) for details concerning these formats.

- 2 If required, type your case-sensitive login ID.
- 3 If required, type your case-sensitive password for the system.
- 4 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 that you specified on the Connection tab of the ODBC Sybase Wire Protocol driver Setup dialog box.

NOTE: If you are connecting through the **Test Connect** button of the Setup dialog box, only the default database specified on the General tab of the Setup dialog box is available in the Database drop-down list. The database names specified on the Connection tab are not available.

- 5 Click **OK** to complete the logon and to update the values in the Registry.

# Connection Option Descriptions

The following connection option descriptions are listed alphabetically by the GUI name that appears on the driver Setup dialog box. The connection string attribute name, along with its short name, is listed immediately underneath the GUI name. For example:

## Application Using Threads

Attribute    ApplicationUsingThreads (AUT)

In most cases, the GUI name and the attribute name are the same; however, some exceptions exist. If you need to look up an option by its connection string attribute name, please refer to the alphabetical table of connection string attribute names.

Also, a few connection string attributes, for example, Password, do not have equivalent options that appear on the GUI. They are in the list of descriptions alphabetically by their attribute names.

Table 12-1 lists the connection string attributes supported by the Sybase Wire Protocol driver.

Table 12-1. Sybase Wire Protocol Attribute Names	
Attribute (Short Name)	Default
AlternateServers (ASRV)	None
ApplicationName (APP)	None
ApplicationUsingThreads (AUT)	1 (Enabled)
ArraySize (AS)	50
AuthenticationMethod (AM)	0 (No Encryption)
BulkBinaryThreshold (BBT)	32
BulkCharacterThreshold (BCT)	-1
BulkLoadBatchSize (BLBS)	1024
BulkLoadFieldDelimiter (BLFD)	None
BulkLoadRecordDelimiter (BLRD)	None
Charset (CS)	None
ConnectionReset (CR)	0 (Disabled)
ConnectionRetryCount (CRC)	0
ConnectionRetryDelay (CRD)	3
CursorCacheSize (CCS)	1
Database (DB)	None
Database List	None
DataSourceName (DSN)	None
DefaultLongDataBuffLen (DLDBL)	1024
Description (n/a)	None
DistributedTransactionModel (DTM)	0 (XA Protocol)

**Table 12-1. Sybase Wire Protocol Attribute Names** (cont.)

Attribute (Short Name)	Default
EnableBulkLoad (EBL)	0 (Disabled)
EnableDescribeParam (EDP)	0 (Disabled)
EnableQuotedIdentifiers (EQI)	0 (Disabled)
EncryptionMethod (EM)	0 (No Encryption)
FailoverGranularity (FG)	0 (Non-Atomic)
FailoverMode (FM)	0 (Connection)
FailoverNetworkAddress (FNA)	None
FailoverPreconnect (FP)	0 (Disabled)
FetchTWFSasTime (FTWFSAT)	1
GSSClient (GSSC)	native
HostNameInCertificate (HNIC)	None
IANAAppCodePage (IACP)	4 (ISO 8559-1 Latin-1)
UNIX ONLY	
InitializationString (IS)	None
InterfacesFile (IF)	None
InterfacesFileServerName (IFSN)	None
Language (LANG)	None
LoadBalanceTimeout (LBT)	0
LoadBalancing (LB)	0 (Disabled)
LoginTimeout (LT)	15
LogonID (UID)	None
MaxPoolSize (MXPS)	100
MinPoolSize (MNPS)	0
NetworkAddress (NA)	None
OptimizePrepare (OP)	1 (Partial)
PacketSize (PS)	0
Password (PWD)	None
Pooling (POOL)	0 (Disabled)
QueryTimeout (QT)	0
RaiseErrorPositionBehavior (REPB)	0 (Default)
ReportCodepageConversionErrors (RCCE)	0 (Ignore Errors)
SelectMethod (SM)	0 (Cursor)
ServicePrincipalName (SPN)	None
TightlyCoupledDistributedTransactions (TCDT)	1 (Enabled)
TruncateTimeTypeFractions (TTTF)	0 (Disabled)
Truststore (TS)	None
TruststorePassword (TSP)	Password
ValidateServerCertificate (VSC)	1 (Enabled)

Table 12-1. Sybase Wire Protocol Attribute Names (cont.)

Attribute (Short Name)	Default
WorkstationID (WKID)	None
XAOpenStringParameters (XAOSP)	None

Alternate Servers

Attribute	AlternateServers (ASRV)
Description	<p>A list of alternate database servers to which the driver tries to connect if the primary database server is unavailable. Specifying a value for this option enables connection failover for the driver. The value you specify must be in the form of a string that defines the physical location of each alternate server. All of the other required connection information for each alternate server is the same as what is defined for the primary server connection.</p>
Valid Values	<p><code>({NetworkAddress=addressvalue   InterfacesFileServerName=sectionvalue}[ , ...])</code></p> <p>NetworkAddress and InterfacesFileServerName can be used in the same string.</p> <p>NOTE: An alternate server address in IPv6 format must be enclosed in double quotation marks.</p> <p>You must specify the network address of each alternate database server or the section in the Interfaces file that contains the network connection information for the Sybase database server you want to access (InterfacesFileServerName).</p> <p>NOTE: The Alternate Servers option and the HA Failover Server Connection Information option are mutually exclusive.</p>
Example	<p>The following example Alternate Servers values define three alternate database servers for connection failover:</p> <pre>(InterfacesFileServerName=Accounting, NetworkAddress="255.125.1.11, 4200", NetworkAddress="SybaseASE2, 4200")</pre> <p>In this example, the network address of the last two alternates contain commas. In this case, enclose the network address with double quotation marks as shown.</p>
Default	None
GUI Tab	Failover tab on page 429

Application Name

Attribute	ApplicationName (APP)
Description	The name used by Sybase to identify your application.
Valid Values	<p><i>string</i></p> <p>where <i>string</i> is a valid application name.</p>



Default	None
GUI Tab	<a href="#">Connection tab</a> on page 427

## Application Using Threads

Attribute	ApplicationUsingThreads (AUT)
Description	<p>Determines whether the driver works with applications using multiple ODBC threads.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 465</a> for details.</p>
Valid Values	<p>0   1</p> <p>If set to 1 (Enabled), the driver works with single-threaded and multi-threaded applications.</p> <p>If set to 0 (Disabled), the driver does not work with multi-threaded applications. If using the driver with single-threaded applications, this value avoids additional processing required for ODBC thread-safety standards.</p>
Default	1 (Enabled)
GUI Tab	<a href="#">Advanced tab</a> on page 424

## Authentication Method

Attribute	AuthenticationMethod (AM)
Description	<p>Specifies the method the driver uses to authenticate the user to the server when a connection is established. If the specified authentication method is not supported by the database server, the connection fails and the driver generates an error.</p>
Valid Values	<p>0   1   4</p> <p>If set to 0 (No Encryption), the driver sends the user ID and password in clear text to the server for authentication.</p> <p>If set to 1 (Encrypt Password), the driver sends the user ID in clear text and an encrypted password to the server for authentication.</p> <p>If set to 4 (Kerberos Authentication), the driver uses Kerberos authentication. This method supports both Windows Active Directory Kerberos and MIT Kerberos environments.</p> <p>If the specified authentication method is not supported by the database server, the connection fails and the driver generates an error.</p>
Default	0 (No Encryption)
GUI Tab	<a href="#">Security tab</a> on page 426

**Batch Size**

Attribute	BulkLoadBatchSize (BLBS)
Description	<p>The number of rows that the driver sends to the database at a time during bulk operations. This value applies to all methods of bulk loading.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 465</a> for details.</p>
Valid Values	<p>0   <i>x</i></p> <p>where <i>x</i> is the number of rows to send during a bulk operation.</p>
Default	1024
GUI Tab	<a href="#">Bulk tab</a> on page 431

**Bulk Binary Threshold**

Attribute	BulkBinaryThreshold (BBT)
Description	<p>The maximum size, in KB, of binary data that is exported to the bulk data file.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 465</a> for details.</p>
Valid Values	<p>-1   0   <i>x</i></p> <p>where <i>x</i> is an integer that specifies the number of KB.</p> <p>If set to -1, all binary data, regardless of size, is written to the bulk data file, not to an external file.</p> <p>If set to 0, all binary data, regardless of size, is written to an external file, not the bulk data file. A reference to the external file is written to the bulk data file.</p> <p>If set to <i>x</i>, any binary data exceeding this specified number of KB is written to an external file, not the bulk data file. A reference to the external file is written to the bulk data file.</p>
Default	32
GUI Tab	<a href="#">Bulk tab</a> on page 431

**Bulk Character Threshold**

Attribute	BulkCharacterThreshold (BCT)
Description	The maximum size, in KB, of character data that is exported to the bulk data file.
Valid Values	<p>-1   0   <i>x</i></p> <p>where <i>x</i> is an integer that specifies the number of KB.</p>

If set to -1, all character data, regardless of size, is written to the bulk data file, not to an external file.

If set to 0, all character data regardless of size, is written to an external file, not the bulk data file. A reference to the external file is written to the bulk data file.

If set to *x*, any character data exceeding this specified number of KB is written to an external file, not the bulk data file. A reference to the external file is written to the bulk data file.

Default -1

GUI Tab [Bulk tab](#) on page 431

## Charset

Attribute Charset (CS)

Description The name of a character set installed on the Sybase server to be used by the driver.

This option is not a substitute for the IANAAppCodePage option. See [IANAAppCodePage](#) for details.

Valid Values *charset*

where *charset* is the name of a character set installed on the Sybase server.

If unspecified, the character set setting on the Sybase server is used.

For the driver to return Unicode SQL types for connections to Sybase 12.5 and higher, use a value of UTF-8. Refer to the Sybase server documentation for a list of valid character sets.

Example If your client needs to receive data in iso-8859-1 from a non-Unicode Sybase server, you would specify a value of iso\_1.

Default None

GUI Tab [Connection tab](#) on page 427

## Connection Cache Size

Attribute CursorCacheSize (CCS)

Description The number of connections that the connection cache can hold.

Valid Values *x*

where *x* is a positive integer representing the number of connections that the connection cache can hold.

To enable the connection cache, you must set the Select Method option to 1 (enabled). Increasing the connection cache may increase performance of some applications but requires additional database resources.

Default 1

GUI Tab [Performance tab](#) on page 430

## Connection Pooling

Attribute	Pooling (POOL)
Description	Specifies whether to use the driver's connection pooling.  NOTE: The application must be thread-enabled to use connection pooling.  Determines whether the driver works with applications using multiple ODBC threads.  This connection option can affect performance. See <a href="#">"Performance Considerations" on page 465</a> for details.
Valid Values	0   1  If set to 1 (Enabled), the driver uses connection pooling.  If set to 0 (Disabled), the driver does not use connection pooling.
Default	0 (Disabled)
GUI Tab	<a href="#">Pooling tab</a> on page 430

## Connection Reset

Attribute	ConnectionReset (CR)
Description	Determines whether the state of connections that are removed from the connection pool for reuse by the application is reset to the initial configuration of the connection.  This connection option can affect performance. See <a href="#">"Performance Considerations" on page 465</a> for details.
Valid Values	0   1  If set to 1 (Enabled), the state of connections removed from the connection pool for reuse by an application is reset to the initial configuration of the connection. Resetting the state can negatively impact performance because additional commands must be sent over the network to the server to reset the state of the connection.  If set to 0 (Disabled), the state of connections is not reset.
Default	0 (Disabled)
GUI Tab	<a href="#">Pooling tab</a> on page 430

## Connection Retry Count

Attribute	ConnectionRetryCount (CRC)
Description	The number of times the driver retries connection attempts to the primary database server, and if specified, alternate servers until a successful connection is established.  This option and the Connection Retry Delay connection option, which specifies the wait interval between attempts, can be used in conjunction with failover.
Valid Values	0   $x$  where $x$ is a positive integer from 1 to 65535.

If set to 0, the driver does not try to connect after the initial unsuccessful attempt.

If set to  $x$ , the driver retries connection attempts the specified number of times. If a connection is not established during the retry attempts, the driver returns an error that is generated by the last server to which it tried to connect.

Default 0

GUI Tab [Failover tab](#) on page 429

## Connection Retry Delay

Attribute ConnectionRetryDelay (CRD)

Description The number of seconds the driver waits between connection retry attempts when Connection Retry Count is set to a positive integer.

This option and the Connection Retry Count connection option can be used in conjunction with failover.

Valid Values 0 |  $x$

where  $x$  is a positive integer from 1 to 65535.

If set to 0, there is no delay between retries.

If set to  $x$ , the driver waits the specified number of seconds between connection retry attempts.

Default 3

GUI Tab [Failover tab](#) on page 429

## Cursor Positioning for Raiserror

Attribute RaiseErrorPositionBehavior (REPB)

Description Determines whether the driver returns raiserrors when the next statement is executed or handles them separately.

Valid Values 0 | 1

If set to 0 (Default), raiserrors are handled separately from surrounding statements. The error is returned when a raiserror is processed (for example, resulting from SQLExecute, SQLExecDirect, or SQLMoreResults). The result set is empty.

If set to 1 (Microsoft compatible), raiserrors are returned when the next statement is processed, and the cursor is positioned on the first row of the subsequent result set. This could result in multiple raiserrors being returned on a single execute.

Default 0 (Default)

GUI Tab [Advanced tab](#) on page 424

## Data Source Name

Attribute	DataSourceName (DSN)
Description	The name of a data source in your Windows Registry or odbc.ini file.
Valid Values	<i>string</i>  where <i>string</i> is the name of a data source.
Default	None
GUI Tab	<a href="#">General tab</a> on page 424

## Database List

Attribute	n/a
Description	A list of database names that will appear in the drop-down list of the logon dialog box (see <a href="#">"Using a Logon Dialog Box" on page 437</a> for a description).
Valid Values	<i>database_list</i>  where <i>database_list</i> is a comma-separated list of database names that will appear in the drop-down list of the logon dialog box.
Default	None
GUI Tab	<a href="#">Connection tab</a> on page 427

## Database Name

Attribute	Database (DB)
Description	The name of the database to which you want to connect.
Valid Values	<i>database_name</i>  where <i>database_name</i> is the name of a valid database.
Default	None
GUI Tab	<a href="#">General tab</a> on page 424

## Description

Attribute	Description (n/a)
Description	An optional long description of a data source. This description is not used as a runtime connection attribute, but does appear in the ODBC.INI section of the Registry and in the odbc.ini file.
Valid Values	<i>string</i>  where <i>string</i> is a description of a data source.
Default	None
GUI Tab	<a href="#">General tab</a> on page 424

## Default Buffer Size for Long/LOB Columns (in Kb)

Attribute	DefaultLongDataBuffLen (DLDBL)
Description	<p>The maximum length of data (in KB) the driver can fetch from long columns in a single round trip and the maximum length of data that the driver can send using the SQL_DATA_AT_EXEC parameter.</p> <p>This option also applies to binding long parameters in chunks. The driver truncates any data passed in a Long/LOB SQL_DATA_AT_EXEC parameter to the size specified.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 465</a> for details.</p>
Valid Values	<p>An integer in multiples of 1024</p> <p>The value must be in multiples of 1024 (for example, 1024, 2048). You need to increase the default value if the total size of any Long data exceeds 1 MB. This value is multiplied by 1024 to determine the total maximum length of fetched data. For example, if you enter a value of 2048, the maximum length of data would be 1024 x 2048, or 2097152 (2 MB).</p>
Default	1024
GUI Tab	<a href="#">Advanced tab</a> on page 424

## Distributed Transaction Model

Attribute	DistributedTransactionModel (DTM)
Description	The model to use for distributed transaction support. The driver supports two different models: XA Protocol and Native OLE.
Valid Values	<p>0   1</p> <p>Specify the appropriate distributed transaction protocol, either 0 (XA Protocol) or 1 (Native OLE)</p>
Default	0 (XA Protocol)
GUI Tab	<a href="#">Advanced tab</a> on page 424

## Enable Bulk Load

Attribute	EnableBulkLoad (EBL)
Description	<p>Specifies the bulk load method.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 465</a> for details.</p>
Valid Values	<p>0   1</p> <p>If set to 1 (Enabled), the driver uses the database bulk load protocol when an application executes an INSERT with multiple rows of parameter data. If the protocol cannot be used, the driver returns a warning.</p> <p>If set to 0 (Disabled), the driver uses standard parameter arrays.</p>

Default 0 (Disabled)  
 GUI Tab [Bulk tab](#) on page 431

## Enable Describe Parameter

Attribute EnableDescribeParam (EDP)  
 Description Determines whether the driver supports the SQLDescribeParam function, which allows an application to describe parameters in SQL statements and in stored procedure calls.  
 Valid Values 0 | 1

If set to 1 (Enabled), the driver supports SQLDescribeParam. The Prepare Method option must be set to 0 or 1, and the SQL statement must not include long parameters. If using Microsoft Remote Data Objects (RDO) to access data, you must use this value.

If set to 0 (Disabled), the driver does not support SQLDescribeParam.

Default 0 (Disabled)  
 GUI Tab [Advanced tab](#) on page 424

## Enable Quoted Identifiers

Attribute EnableQuotedIdentifiers (EQI)  
 Description Determines whether the driver supports the use of quoted identifiers.  
 Valid Values 0 | 1

If set to 1 (Enabled), the driver supports the use of quoted identifiers. Double quotation marks (") must be used to enclose identifiers, such as column and table names. Character strings must be enclosed in single quotation marks, for example:

```
SELECT "au_id"
FROM "authors"
WHERE "au_lname" = 'O'Brien'
```

If set to 0 (Disabled), the driver does not support the use of quoted identifiers and generates an error when quoted identifiers are encountered.

Default 0 (Disabled)  
 GUI Tab [Advanced tab](#) on page 424

## Encryption Method

Attribute EncryptionMethod (EM)  
 Description The method the driver uses to encrypt data sent between the driver and the database server. If the specified encryption method is not supported by the database server, the connection fails and the driver returns an error.

This connection option can affect performance. See ["Performance Considerations" on page 465](#) for details.



Valid Values 0 | 1

If set to 0 (No Encryption), data is not encrypted.

If set to 1 (SSL), data is encrypted using SSL. If the server supports protocol negotiation, the driver and server negotiate the use of TLS v1, SSL v3, or SSL v2 in that order.

This option can only be set to 1 when Authentication Method is set to 0 or 1.

Default 0 (No Encryption)

GUI Tab [Security tab](#) on page 426

## Failover Granularity

Attribute FailoverGranularity (FG)

Description Determines whether the driver fails the entire failover process or continues with the process if errors occur while trying to reestablish a lost connection.

This option applies only when Failover Mode is set to 1 (Extended Connection) or 2 (Select).

The Alternate Servers option specifies one or multiple alternate servers for failover and is required for all failover methods.

Valid Values 0 | 1 | 2 | 3

If set to 0 (Non-Atomic), the driver continues with the failover process and posts any errors on the statement on which they occur.

If set to 1 (Atomic) the driver fails the entire failover process if an error is generated as the result of anything other than executing and repositioning a Select statement. If an error is generated as a result of repositioning a result set to the last row position, the driver continues with the failover process, but generates a warning that the Select statement must be reissued.

If set to 2 (Atomic Including Repositioning), the driver fails the entire failover process if any error is generated as the result of restoring the state of the connection or the state of work in progress.

If set to 3 (Disable Integrity Check), the driver does not verify that the rows that were restored during the failover process match the original rows. This value applies only when Failover Mode is set to 2 (Select).

Default 0 (Non-Atomic)

GUI Tab [Failover tab](#) on page 429

## Failover Mode

Attribute	FailoverMode (FM)
Description	<p>Specifies the type of failover method the driver uses.</p> <p>The Alternate Servers option specifies one or multiple alternate servers for failover and is required for all failover methods.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 465</a> for details.</p>
Valid Values	<p>0   1   2</p> <p>If set to 0 (Connection), the driver provides failover protection for new connections only.</p> <p>If set to 1 (Extended Connection), the driver provides failover protection for new and lost connections, but not any work in progress.</p> <p>If set to 2 (Select), the driver provides failover protection for new and lost connections. In addition, it preserves the state of work performed by the last Select statement executed.</p>
Default	0 (Connection)
GUI Tab	<a href="#">Failover tab</a> on page 429

## Failover Preconnect

Attribute	FailoverPreconnect (FP)
Description	<p>Specifies whether the driver tries to connect to the primary and an alternate server at the same time.</p> <p>This attribute applies only when Failover Mode is set to 1 (Extended Connection) or 2 (Select) and at least one alternate server is specified.</p> <p>The Alternate Servers option specifies one or multiple alternate servers for failover and is required for all failover methods.</p>
Valid Values	<p>0   1</p> <p>If set to 0 (Disabled), the driver tries to connect to an alternate server only when failover is caused by an unsuccessful connection attempt or a lost connection. This value provides the best performance, but your application typically experiences a short wait while the failover connection is attempted.</p> <p>If set to 1 (Enabled), the driver tries to connect to the primary and an alternate server at the same time. This can be useful if your application is time-sensitive and cannot absorb the wait for the failover connection to succeed.</p>
Default	0 (Disabled)
GUI Tab	<a href="#">Failover tab</a> on page 429

## Fetch Array Size

Attribute	ArraySize (AS)
Description	<p>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 connection option can affect performance.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 465</a> for details.</p>
Valid Values	<p><math>x</math></p> <p>where <math>x</math> is a positive integer specifying the number of rows.</p>
Default	50
GUI Tab	<a href="#">Performance tab</a> on page 428

## Fetch TWFS as Time

Attribute	FetchTWFSasTime (FTWFSAT)
Description	<p>Determines which ODBC data type the driver uses to return column values with the BIGTIME data type.</p> <p>NOTE: The BIGTIME data type is supported in Sybase 15.5 and higher.</p>
Valid Values	<p>0   1</p> <p>If set to 1, the driver returns column values for the BIGTIME data type as the ODBC data type SQL_TYPE_TIME. The fractional seconds portion of the value is truncated.</p> <p>If set to 0, the driver returns column values for the BIGTIME data type as the ODBC data type SQL_TYPE_TIMESTAMP. When a timestamp is returned for BIGTIME, the Year, Month and Day parts of the timestamp must be set to zero.</p>
Default	1
GUI Tab	Advanced

## Field Delimiter

Attribute	BulkLoadFieldDelimiter (BLFD)
Description	<p>Specifies the character that the driver will use to delimit the field entries in a bulk load data file.</p>
Valid Values	<p><math>x</math></p> <p>where <math>x</math> is any printable character.</p> <p>For simplicity, avoid using a value that can be in the data, including all alphanumeric characters, the dash(-), the colon(:), the period (.), the forward slash (/), the space character, the single quote (') and the double quote ("). You can use some of these characters as delimiters if all of the data in the file is contained within double quotes.</p> <p>NOTE: The Bulk Load Field Delimiter character must be different from the Bulk Load Record Delimiter.</p>

Default     None  
GUI Tab    [Bulk tab](#) on page 431

**GSS Client Library**

Attribute    GSSClient (GSSC)  
Description   The name of the GSS client library that the driver uses to communicate with the Key Distribution Center (KDC).  
  
              The driver uses the path defined by the PATH environment variable for loading the specified client library.  
Valid Values   native | *client\_library*  
  
              where *client\_library* is a GSS client library installed on the client.  
  
              If set to *client\_library*, the driver uses the specified GSS client library.  
  
              If set to native, the driver uses the GSS client shipped with the operating system.  
Default     native  
GUI Tab    [Security tab](#) on page 426

**HA Failover Server Connection Information/Network Address**

Attribute    FailoverNetworkAddress (FNA)  
Description   The network 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 specified HA Failover server. This option is valid only for Sybase 12 and higher servers that have the High Availability Failover feature enabled.  
Valid Values   *IP\_address, port\_number* | *pipe\_address, port\_number* | *server\_name, port\_number*  
  
              where  
  
              *IP\_ address* is the IP address that uniquely identifies the HA Failover server.  
  
              *port\_number* is the port number assigned to the listener process on the HA Failover server.  
  
              *server\_name* is a name that uniquely identifies the HA Failover server. You can use this format if your environment supports named servers.  
  
              *pipe\_address* is the pipe address of the HA Failover server. This format is required if using NamedPipes as the network protocol.  
  
NOTE: The HA Failover Server Connection Information option and the Alternate Servers option are mutually exclusive.

**Example** 199.226.224.34, 5000

or

\\machine1\sybase\pipe\query, 5000

or

Sybaseserver, 5000

**Default** None

**GUI Tab** [Failover tab](#) on page 429

## Host Name In Certificate

**Attribute** HostNameInCertificate (HNIC)

**Description** A host name for certificate validation when SSL encryption is enabled (Encryption Method=1) and validation is enabled (Validate Server Certificate=1). This option provides additional security against man-in-the-middle (MITM) attacks by ensuring that the server the driver is connecting to is the server that was requested.

**Valid Values** *host\_name* | #SERVERNAME#

where the *host\_name* is the host name specified in the certificate. Consult your SSL administrator for the correct value.

If set to a host name, the driver examines the subjectAltName values included in the certificate. If a dnsName value is present in the subjectAltName values, then the driver compares the value specified for Host Name In Certificate with the dnsName value. The connection succeeds if the values match. The connection fails if the Host Name In Certificate value does not match the dnsName value.

If no subjectAltName values exist or a dnsName value is not in the list of subjectAltName values, then the driver compares the value specified for Host Name In Certificate with the commonName part of the Subject name in the certificate. The commonName typically contains the host name of the machine for which the certificate was created. The connection succeeds if the values match. The connection fails if the Host Name In Certificate value does not match the commonName. If multiple commonName parts exist in the Subject name of the certificate, the connection succeeds if the Host Name In Certificate value matches any of the commonName parts.

If set to #SERVERNAME#, the driver compares the host server name specified as part of a data source or connection string to the dnsName or the commonName value.

**Default** None

**GUI Tab** [Security tab](#) on page 426

## IANAAppCodePage

Attribute	IANAAppCodePage (IACP)
Description	<p>An Internet Assigned Numbers Authority (IANA) value. You must specify a value for this option if your application is not Unicode-enabled or if your database character set is not Unicode. Refer to <a href="#">Chapter 4 “Internationalization, Localization, and Unicode”</a> in the <i>DataDirect Connect Series for ODBC Reference</i> for details.</p> <p>The driver uses the specified IANA code page to convert "W" (wide) functions to ANSI.</p> <p>The driver and Driver Manager both check for the value of IANAAppCodePage in the following order:</p> <ul style="list-style-type: none"> <li>■ In the connection string</li> <li>■ In the Data Source section of the system information file (odbc.ini)</li> <li>■ In the ODBC section of the system information file (odbc.ini)</li> </ul> <p>If the driver does not find an IANAAppCodePage value, the driver uses the default value of 4 (ISO 8859-1 Latin-1).</p>
Valid Values	<p><i>IANA_code_page</i></p> <p>where <i>IANA_code_page</i> is one of the valid values listed in <a href="#">Chapter 1 “Values for the Attribute IANAAppCodePage”</a> in the <i>DataDirect Connect Series for ODBC Reference</i>. The value must match the database character encoding and the system locale.</p>
Default	4 (ISO 8559-1 Latin-1)
GUI Tab	<a href="#">Advanced tab</a> on page 424

## Initialization String

Attribute	InitializationString (IS)
Description	<p>A SQL command that is issued immediately after connecting to the database to manage session settings.</p> <p>NOTE: If the statement fails to execute, the connection fails and the driver reports the error returned from the server.</p>
Valid Values	<p><i>SQL_command</i></p> <p>where <i>SQL_command</i> is a valid SQL command that is supported by the database.</p>
Example	<p>To allow delimited identifiers, specify:</p> <pre>Initialization String=set QUOTED_IDENTIFIER on</pre>
Default	None
GUI Tab	<a href="#">Advanced tab</a> on page 424

## Interfaces File

Attribute	InterfacesFile (IF)
Description	The directory to the Interfaces file.  NOTE: This option and the Network Address option are mutually exclusive.
Valid Values	<i>file_dir</i>  where <i>file_dir</i> is the directory to the Interfaces file.  If unspecified and a value is specified for the Server Name option, the driver looks for the path name of the Interfaces file in the Registry under HKEY_LOCAL_MACHINE\SOFTWARE\DataDirect\InterfacesFile. If this Registry value is empty, the driver will try to open the SQL.INI file found in the same directory where the driver is located and use it as the Interfaces file.
Default	None
GUI Tab	<a href="#">General tab</a> on page 424

## Language

Attribute	Language (LANG)
Description	The national character set installed on the Sybase server.
Valid Values	<i>charset</i>  where <i>charset</i> is the national character set installed on the Sybase server.
Default	None (English)
GUI Tab	<a href="#">Connection tab</a> on page 427

## Load Balance Timeout

Attribute	LoadBalanceTimeout (LBT)
Description	The number of seconds to keep inactive connections open in a connection pool. An inactive connection is a database session that is not associated with an ODBC connection handle, that is, a connection in the pool that is not in use by an application.  NOTE: The Min Pool Size option may cause some connections to ignore this value.  Determines whether the driver works with applications using multiple ODBC threads.  This connection option can affect performance. See <a href="#">"Performance Considerations" on page 465</a> for details.
Valid Values	0   <i>x</i>  where <i>x</i> is a positive integer that specifies a number of seconds.  If set to 0, inactive connections are kept open.  If set to <i>x</i> , inactive connections are closed after the specified number of seconds passes.

Default 0 (Disabled)  
 GUI Tab [Pooling tab](#) on page 430

## Load Balancing

Attribute LoadBalancing (LB)  
 Description Determines whether the driver uses client load balancing in its attempts to connect to the database servers (primary and alternate). You can specify one or multiple alternate servers by setting the Alternate Servers option.  
 Valid Values 0 | 1

If set to 1 (Enabled), the driver uses client load balancing and attempts to connect to the database servers (primary and alternate servers) in random order.

If set to 0 (Disabled), the driver does not use client load balancing and connects to each server based on their sequential order (primary server first, then, alternate servers in the order they are specified).

NOTE: This option has no effect unless alternate servers are defined for the Alternate Servers connection option.

Default 0 (Disabled)  
 GUI Tab [Failover tab](#) on page 429

## Login Timeout

Attribute LoginTimeout (LT)  
 Description The number of seconds the driver waits for a connection to be established before returning control to the application and generating a timeout error. To override the value that is set by this connection option for an individual connection, set a different value in the SQL\_ATTR\_LOGIN\_TIMEOUT connection attribute using the SQLSetConnectAttr() function.  
 Valid Values -1 | 0 |  $x$

where  $x$  is a positive integer that specifies a number of seconds.

If set to -1, the connection request does not time out. The driver silently ignores the SQL\_ATTR\_LOGIN\_TIMEOUT attribute.

If set to 0, the connection request does not time out, but the driver responds to the SQL\_ATTR\_LOGIN\_TIMEOUT attribute.

If set to  $x$ , the connection request times out after the specified number of seconds unless the application overrides this setting with the SQL\_ATTR\_LOGIN\_TIMEOUT attribute.

Default 15  
 GUI Tab [Advanced tab](#) on page 424



## Max Pool Size

Attribute	MaxPoolSize (MXPS)
Description	<p>The maximum number of connections allowed within a single connection pool. When the maximum number of connections is reached, no additional connections can be created in the connection pool.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 465</a> for details.</p>
Valid Values	<p>An integer from 1 to 65535</p> <p>For example, if set to 20, the maximum number of connections allowed in the pool is 20.</p>
Default	100
GUI Tab	<a href="#">Pooling tab</a> on page 430

## Min Pool Size

Attribute	MinPoolSize (MNPS)
Description	<p>The minimum number of connections that are opened and placed in a connection pool, in addition to the active connection, when the pool is created. The connection pool retains this number of connections, even when some connections exceed their Load Balance Timeout value.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 465</a> for details.</p>
Valid Values	<p>0   <i>x</i></p> <p>where <i>x</i> is an integer from 1 to 65535.</p> <p>For example, if set to 5, the start-up number of connections in the pool is 5 in addition to the current existing connection.</p> <p>If set to 0, no connections are opened in addition to the current existing connection.</p>
Default	0
GUI Tab	<a href="#">Pooling tab</a> on page 430

## Network Address

Attribute	NetworkAddress (NA)
Description	<p>A unique identifier assigned to the Sybase server machine.</p> <p>NOTE: This option is mutually exclusive with the Interfaces File and the Server Name option.</p>
Valid Values	<p><i>server_name</i>   <i>IP_address</i></p> <p>where</p> <p><i>server_name</i> is the Sybase server name specified as: <i>named_server</i>, <i>port_number</i>. For example, you can enter <i>SSserver</i>, 5000.</p>

*IP\_address* is the Sybase server address specified as: *IP\_address, port\_number*. For example, you can enter 199.226.224.34, 5000. The IP address can be specified in either IPv4 or IPv6 format, or a combination of the two. See ["Using IP Addresses" on page 54](#) for details about these formats.

- Default None
- GUI Tab [General tab](#) on page 424

Packet Size

- Attribute PacketSize (PS)
- Description Determines the number of bytes for each database protocol packet that is transferred from the database server to the client machine. Adjusting the packet size can improve performance. The optimal value depends on the typical size of data that is inserted, updated, or returned by the application and the environment in which it is running. Typically, larger packet sizes work better for large amounts of data. For example, if an application regularly returns character values that are 10,000 characters in length, using a value of 32 (16 KB) typically results in improved performance.

NOTE: The ODBC connection option SQL\_PACKET\_SIZE provides the same functionality as the Packet Size option; however SQL\_PACKET\_SIZE and the Packet Size option are mutually exclusive. If Packet Size is specified, the driver returns the message *Driver Not Capable* if an application attempts to call SQL\_PACKET\_SIZE. If you do not set the Packet Size option, application calls to SQL\_PACKET\_SIZE are accepted by the driver.

This connection option can affect performance. See ["Performance Considerations" on page 465](#) for details.

- Valid Values -1 | 0 | *x*
  - If set to -1, the driver uses the maximum packet size that is set by the database server.
  - If set to 0, the driver uses the default packet size that is used by the database server.
  - If set to *x*, an integer from 1 to 127, the driver uses a packet size that is a multiple of 512 bytes. For example, PacketSize=8 means to set the packet size to 8 \* 512 bytes (4096 bytes).
- NOTE: IF SSL encryption is used, the driver must use the packet size that is specified by the server. Any value set for this option or the SQL\_PACKET\_SIZE connect option is ignored if SSL encryption is used.

- Default 0
- GUI Tab [Performance tab](#) on page 428

Password

- Attribute Password (PWD)
- Description The password that the application uses to connect to your database. The Password option cannot be specified through the driver Setup dialog box and should not be stored in a data source. It is specified through the Logon dialog box or a connection string.

Valid Values	<i>pwd</i> where <i>pwd</i> is a valid password.
Default	None
GUI Tab	n/a

## Prepare Method

Attribute	OptimizePrepare (OP)
Description	Determines whether stored procedures are created on the server for calls to SQLPrepare.  This connection option can affect performance. See <a href="#">"Performance Considerations" on page 465</a> for details.
Valid Values	0   1   2   3  If set to 0 - None, stored procedures are created for every call to SQLPrepare. This setting can result in decreased performance when processing statements that do not contain parameters.  If set to 1 - Partial, the driver creates stored procedures only if the statement contains parameters. Otherwise, the statement is cached and run directly at the time of SQLExecute.  If set to 2 - Full, 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.  If set to 3 - Full at Prepare, stored procedures are never created. This is identical to value 2 except that any syntax or similar errors are returned at the time of SQLPrepare instead of SQLExecute. Use this setting only if you must have syntax errors reported at the time of SQLPrepare.
Default	1 (Partial)
GUI Tab	<a href="#">Performance tab</a> on page 428

## Query Timeout

Attribute	QueryTimeout (QT)
Description	The number of seconds for the default query timeout for all statements that are created by a connection. To override the value set by this connection option for an individual statement, set a different value in the SQL_ATTR_QUERY_TIMEOUT statement attribute on the SQLSetStmtAttr() function.
Valid Values	-1   0   <i>x</i>  where <i>x</i> is a positive integer that specifies a number of seconds.  If set to -1, the query does not time out. The driver silently ignores the SQL_ATTR_QUERY_TIMEOUT attribute.  If set to 0, the query does not time out, but the driver responds to the SQL_ATTR_QUERY_TIMEOUT attribute.

If set to *x*, all queries time out after the specified number of seconds unless the application overrides this value by setting the SQL\_ATTR\_QUERY\_TIMEOUT attribute.

Default 0

GUI Tab [Advanced tab](#) on page 424

## Record Delimiter

Attribute BulkLoadRecordDelimiter (BLRD)

Description Specifies the character that the driver will use to delimit the record entries in a bulk load data file.

Valid Values *x*

where *x* is any printable character.

For simplicity, avoid using a value that can be in the data, including all alphanumeric characters, the dash(-), the colon(:), the period (.), the forward slash (/), the space character, the single quote (') and the double quote ("). You can use some of these characters as delimiters if all of the data in the file is contained within double quotes.

NOTE: The Bulk Load Record Delimiter character must be different from the Bulk Load Field Delimiter.

Default None

GUI Tab [Bulk tab](#) on page 431

## Report Codepage Conversion Errors

Attribute ReportCodepageConversionErrors (RCCE)

Description Specifies how the driver handles code page conversion errors that occur when a character cannot be converted from one character set to another.

An error message or warning can occur if an ODBC call causes a conversion error, or if an error occurs during code page conversions to and from the database or to and from the application. The error or warning generated is `Code page conversion error encountered`. In the case of parameter data conversion errors, the driver adds the following sentence:

Error in parameter *x*, where *x* is the parameter number. The standard rules for returning specific row and column errors for bulk operations apply.

Valid Values 0 | 1 | 2

If set to 0 (Ignore Errors), the driver substitutes 0x1A for each character that cannot be converted and does not return a warning or error.

If set to 1 (Return Error), the driver returns an error instead of substituting 0x1A for unconverted characters.

If set to 2 (Return Warning), the driver substitutes 0x1A for each character that cannot be converted and returns a warning.

Default 0 (Ignore Errors)

GUI Tab [Advanced tab](#) on page 424

## Select Method

Attribute	SelectMethod (SM)
Description	<p>Determines whether database cursors are used for Select statements.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 465</a> for details.</p>
Valid Values	<p>0   1</p> <p>If set to 0 (Cursor), database cursors are used. In some cases performance degradation can occur when performing large numbers of sequential Select statements because of the amount of overhead associated with creating database cursors.</p> <p>If set to 1 (Direct), Select statements are run directly without using database cursors, and the data source is limited to one active statement.</p>
Default	0 (Cursor)
GUI Tab	<a href="#">Performance tab</a> on page 428

## Server Name

Attribute	InterfacesFileName (IFSN)
Description	<p>The name of the section in the Interfaces file containing the network connection information for the Sybase server. Typically, the section name is the host name of the Sybase server.</p> <p>NOTE: The Network Address option and the Server Name option are mutually exclusive.</p>
Valid Values	<p><i>section_name</i></p> <p>where <i>section_name</i> is a section in the Interfaces file containing the network connection information for the Sybase server.</p>
Default	None
GUI Tab	<a href="#">General tab</a> on page 424

## Service Principal Name

Attribute	ServicePrincipalName (SPN)
Description	The service principal name to be used by driver for Kerberos authentication.
Valid Values	<p><i>servicePrincipalName</i></p> <p>where <i>servicePrincipalName</i> is a valid service principal name.</p> <p>If unspecified, the value of the Network Address option is used as the service principal name. If Authentication Method is set to 0 or 1, the value of the Service Principal Name option is ignored.</p>
Default	None
GUI Tab	<a href="#">Security tab</a> on page 426

## Tightly Coupled Distributed Transactions

Attribute	TightlyCoupledDistributedTransactions (TCDT)
Description	Sybase 12 or higher server only. Determines whether the driver ensures that multiple connections within the same distributed transaction obey other's locks.
Valid Values	0   1  If set to 1 (Enabled), the driver uses tightly coupled distributed transactions. Multiple connections within the same distributed transaction obey other's locks.  If set to 0 (Disabled), the driver does not use tightly coupled distributed transactions. Multiple connections within the same distributed transaction may hang each other because the connections do not obey other's locks. This value can provide better performance if concurrency of data is not needed.
Default	1 (Enabled)
GUI Tab	<a href="#">Advanced tab</a> on page 424

## Truncate Time Type Fractions

Attribute	TruncateTimeTypeFractions (TTTF)
Description	Sybase 12.5.1 and higher only. Determines whether the driver sets fractional seconds to zero (0) when converting data from the TIME data type to TIMESTAMP, CHAR, or WCHAR data types.
Valid Values	0   1  If set to 1 (Enabled), the driver converts fractional seconds to zero when converting the TIME data type.  If set to 0 (Disabled), the driver does not set fractional seconds to zero when converting the TIME data type.
Default	0 (Disabled)
GUI Tab	<a href="#">Advanced tab</a> on page 424

## Truststore

Attribute	Truststore (TS)
Description	The directory that contains the truststore file and the truststore file name to be used when SSL is enabled (Encryption Method=1) and server authentication is used. The truststore file contains a list of the valid Certificate Authorities (CAs) that are trusted by the client machine for SSL server authentication. If you do not specify a directory, the current directory is used.  NOTE: The truststore and keystore files may be the same file.
Valid Values	<i>truststore_directory\filename</i>  where <i>truststore_directory</i> is the directory where the truststore file is located and <i>filename</i> is the file name of the truststore file.

Default None  
 GUI Tab [Security tab](#) on page 426

## Truststore Password

Attribute TruststorePassword (TSP)

Description The password that is used to access the truststore file when SSL is enabled (Encryption Method=1) and server authentication is used. The truststore file contains a list of the Certificate Authorities (CAs) that the client trusts.

NOTE: The truststore and keystore files may be the same file; therefore, they may have the same password.

Valid Values *truststore\_password*

where *truststore\_password* is a valid password for the truststore file.

Default None

GUI Tab [Security tab](#) on page 426

## User Name

Attribute LogonID (UID)

Description The default user ID that is used to connect to your database. Your ODBC application may override this value or you may override it in the logon dialog box or connection string.

Valid Values *userid*

where *userid* is a valid user ID with permissions to access the database.

Default None

GUI Tab [Security tab](#) on page 426

## Validate Server Certificate

Attribute ValidateServerCertificate (VSC)

Description Determines whether the driver validates the certificate that is sent by the database server when SSL encryption is enabled (Encryption Method=1). When using SSL server authentication, any certificate sent by the server must be issued by a trusted Certificate Authority (CA). Allowing the driver to trust any certificate returned from the server even if the issuer is not a trusted CA is useful in test environments because it eliminates the need to specify truststore information on each client in the test environment.

Truststore information is specified using the Trust Store and Trust Store Password options.

Valid Values 0 | 1

If set to 1 (Enabled), the driver validates the certificate that is sent by the database server. Any certificate from the server must be issued by a trusted CA in the truststore file. If the Host Name In Certificate option is specified, the driver also validates the certificate using a host name. The Host Name In Certificate option provides additional security against

man-in-the-middle (MITM) attacks by ensuring that the server the driver is connecting to is the server that was requested.

If set to 0 (Disabled), the driver does not validate the certificate that is sent by the database server. The driver ignores any truststore information specified by the Trust Store and Trust Store Password options.

Default 1 (Enabled)

GUI Tab [Security tab](#) on page 426

**Workstation ID**

Attribute WorkstationID (WKID)

Description An identifier for the client machine.

Valid Values *ID*

where *ID* is workstation ID use by the client machine.

Default None

GUI Tab [Connection tab](#) on page 427

**XA Open String Parameters**

Attribute XAOpenStringParameters (XAOSP)

Description Determines the name of trace files generated for XA open string parameters.

Valid Values *-Ltrace\_filename*

where *trace\_filename* is a string that identifies trace files generated for XA open string parameters. If specified, two trace files are created. The first trace file traces all XA call activities and is named exactly as specified. The second trace file traces any enlistment and unenlistment procedures and is named as specified with a "driver" extension.

Example If you specify *-LXAtrace*, the driver creates two trace files: XAtrace and XAtrace.driver.

Default None

GUI Tab [Advanced tab](#) on page 424



## Performance Considerations

The following connection options can enhance driver performance. You can also enhance performance through efficient application design. Refer to [Chapter 5 “Designing ODBC Applications for Performance Optimization”](#) in the *DataDirect Connect Series for ODBC Reference* for details.

**Application Using Threads (ApplicationUsingThreads):** The driver coordinates concurrent database operations (operations from different threads) by acquiring locks. Although locking prevents errors in the driver, it also decreases performance. If your application does not make ODBC calls from different threads, the driver has no reason to coordinate operations. In this case, the ApplicationUsingThreads attribute should be disabled (set to 0).

NOTE: If you are using a multi-threaded application, you must enable the Application Using Threads option.

**Connection Pooling (ConnectionPooling):** If you enable the driver to use connection pooling, you can set additional options that affect performance:

- **Load Balance Timeout:** You can define how long to keep connections in the pool. The time that a connection was last used is compared to the current time and, if the timespan exceeds the value of the Load Balance Timeout option, the connection is destroyed. The Min Pool Size option can cause some connections to ignore this value.
- **Connection Reset:** Resetting a re-used connection to the initial configuration settings impacts performance negatively because the connection must issue additional commands to the server.
- **Max Pool Size:** Setting the maximum number of connections that the pool can contain too low might cause delays while waiting for a connection to become available. Setting the number too high wastes resources.
- **Min Pool Size:** A connection pool is created when the first connection with a unique connection string connects to the database. The pool is populated with connections up to the minimum pool size, if one has been specified. The connection pool retains this number of connections, even when some connections exceed their Load Balance Timeout value.

**Default Buffer Size for Long/LOB Columns (DefaultLongDataBuffLen):** To improve performance when your application fetches images, pictures, or long text or binary data, a buffer size can be set to accommodate the maximum size of the data. The buffer size should only be large enough to accommodate the maximum amount of data retrieved; otherwise, performance is reduced by transferring large amounts of data into an oversized buffer. If your application retrieves more than 1 MB of data, the buffer size should be increased accordingly.

**Enable Bulk Load (EnableBulkLoad):** If your application performs bulk loading of data, you can improve performance by configuring the driver to use the database system's bulk load functionality instead of database array binding. The trade-off to consider for improved performance is that using the bulk load functionality can bypass data integrity constraints.

**Encryption Method (EncryptionMethod):** Data encryption may adversely affect performance because of the additional overhead (mainly CPU usage) that is required to encrypt and decrypt data.

**Failover Mode (FailoverMode):** Although high availability that replays queries after a failure provides increased levels of protection, it can adversely affect performance because of increased overhead.

**Fetch Array Size (ArraySize):** If the Select Method connection option is set to 0 and your application fetches more than 50 rows at a time, you should set Fetch Array Size to the approximate number of rows being fetched. This reduces the number of round trips on the network, thereby increasing performance. For example, if your application normally fetches 200 rows, it is more efficient for the driver to fetch 200 rows at one time over the network than to fetch 50 rows at a time during four round trips over the network. You should use Fetch Array Size in conjunction with Select Method.

NOTE: The ideal setting for your application will vary. To calculate the ideal setting for this option, you must know the size in bytes of the rows that you are fetching and the size in bytes of your Network Packet. Then, you must calculate the number of rows that will fit in your Network Packet, leaving space for packet overhead. For example, suppose your Network Packet size is 1024 bytes and the row size is 8 bytes. Dividing 1024 by 8 equals 128; however, the ideal setting for Fetch Array Size is 127, not 128, because the number of rows times the row size must be slightly smaller than the Network Packet size.

**Packet Size (PacketSize):** Typically, it is optimal for the client to use the maximum packet size that the database server allows. This reduces the total number of round trips required to return data to the client, thus improving performance. Therefore, performance can be improved if the PacketSize attribute is set to the maximum packet size of the Sybase ASE server.

**Prepare Method (OptimizePrepare):** If your application executes the same SQL statements multiple times, performance can be improved by creating a stored procedure on the server at prepare time. If your application executes one of these prepared statements multiple times, performance will improve because the driver created a stored procedure and executing a stored procedure is faster than executing a single SQL statement; however, if a prepared statement is only executed once or is never executed, performance can decrease. If your application executes the same SQL statements multiple times, the Prepare Method option should be set to 1.

**Select Method (SelectMethod):** If your application often executes a SQL statement before processing or closing the previous result set, then it uses multiple active statements per connection. The default setting (0) of this option causes the driver to use database cursors for Select statements and allows an application to process multiple active statements per connection. An active statement is defined as a statement where all the result rows or result sets have not been fetched. This can cause high overhead on the server. If your application does not use multiple active statements, however, setting Select Method to 1 will increase performance of Select statements by allowing the server to return results without using a database cursor. If this option is set to 0, it should be used in conjunction with Fetch Array Size (ArraySize). If this option is set to 1, Fetch Array Size (ArraySize) has no effect.

# Data Types

Table 12-2 shows how the Sybase data types are mapped to the standard ODBC data types. "Unicode Support" on page 468 lists Sybase to Unicode data type mappings.

**Table 12-2. Sybase Data Type Mapping**

Sybase Data Type...	Maps to ODBC Data Type
BIGDATETIME <sup>1</sup>	SQL_DATETIME
BIGINT <sup>2</sup>	SQL_BIGINT
BIGTIME <sup>1</sup>	SQL_DATETIME
BINARY	SQL_BINARY
BIT	SQL_BIT
CHAR	SQL_CHAR
DATE <sup>3</sup>	SQL_TYPE_DATE
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
TIME <sup>3</sup>	SQL_TYPE_TIME
TIMESTAMP	SQL_VARBINARY
TINYINT	SQL_TINYINT
UNSIGNED BIGINT <sup>2</sup>	SQL_BIGINT
UNSIGNED INT <sup>2</sup>	SQL_INTEGER
UNSIGNED SMALLINT <sup>2</sup>	SQL_SMALLINT

**Table 12-2. Sybase Data Type Mapping** (cont.)

Sybase Data Type...	Maps to ODBC Data Type
VARBINARY	SQL_VARBINARY
VARCHAR	SQL_VARCHAR

- 1. Sybase 15.5 and higher only.
- 2. Sybase 15 and higher only.
- 3. Sybase 12.5.1 and higher only.

NOTE FOR USERS OF SYBASE 12.5 AND HIGHER: The Sybase Wire Protocol driver supports extended new limits (XNL) for character and binary columns—columns with lengths greater than 255.

See ["Retrieving Data Type Information" on page 59](#) for information about retrieving data types.

## Unicode Support

When connected to a Unicode database, the Sybase Wire Protocol driver supports Unicode data types listed in the following table, in addition to standard ODBC data types listed in [Table 12-2 on page 467](#).

Sybase Data Type	Mapped to . . .
CHAR <sup>1</sup>	SQL_WCHAR
SYSNAME <sup>1</sup>	SQL_VARCHAR
TEXT <sup>1</sup>	SQL_WLONGVARCHAR
UNICHAR <sup>2</sup>	SQL_WCHAR
UNITEXT <sup>3</sup>	SQL_WLONGVARCHAR
UNIVARCHAR <sup>2</sup>	SQL_WVARCHAR
VARCHAR <sup>1</sup>	SQL_WVARCHAR

- 1. This data type is available only if the data source is configured to use the UTF-8 character set.
- 2. On Sybase 12.5 servers, this data type is available only if the data source is configured to use the UTF-8 character set. On Sybase 12.5.1 and higher servers, this data type is always available, even if the data source is not configured to use the UTF-8 character set.
- 3. This data type is available on Sybase 15 and higher servers only.

For data types that require the UTF-8 character set, set the Charset connection string attribute. See ["Charset" on page 443](#) for information about using this connection string attribute.

The driver supports the Unicode ODBC W (Wide) function calls, such as SQLConnectW. This allows the Driver Manager to transmit these calls directly to the driver. Otherwise, the

Driver Manager would incur the additional overhead of converting the W calls to ANSI function calls, and vice versa.

See ["UTF-16 Applications on UNIX and Linux" on page 125](#) for related details. Also, refer to [Chapter 4 "Internationalization, Localization, and Unicode"](#) in the *DataDirect Connect Series for ODBC Reference* for a more detailed explanation of Unicode.

---

## Advanced Features

The driver supports the following advanced features:

- Failover
- Security
- Connection Pooling
- DataDirect Bulk Load

### Failover

The driver supports failover and its related connection options. Failover connection options are located on the [Failover tab](#) of the driver Setup dialog box. See ["Using Failover" on page 65](#) for a general description of failover and its implementation.

### Security

The driver supports authentication and encryption. Security connection options are located on the [Security tab](#) of the driver Setup dialog box. See ["Using Security" on page 77](#) for a general description of security and its implementation. The following security information is specific to the Sybase Wire Protocol Driver.

### Authentication

If you are using Kerberos, verify that your environment meets the requirements listed in [Table 12-3](#) before you configure the driver for Kerberos authentication.

**Table 12-3. Kerberos Authentication Requirements for the Sybase Wire Protocol Driver**

Component	Requirements
Database server	The database server must be administered by the same domain controller that administers the client and must be running Sybase 12.0 or higher. In addition, the Sybase Security and directory services package, ASE_SECDIR, is required.
Kerberos server	<p>The Kerberos server is the machine where the user IDs for authentication are administered. The Kerberos server is also the location of the Kerberos KDC.</p> <p>Network authentication must be provided by one of the following methods:</p> <ul style="list-style-type: none"> <li>■ Windows Active Directory on one of the following operating systems: Windows Server 2003 or Windows 2000 Server Service Pack 3 or higher</li> <li>■ MIT Kerberos 1.4.2 or higher</li> </ul>
Client	The client must be administered by the same domain controller that administers the database server.

## Connection Pooling

The driver supports connection pooling and its related connection options. Connection pooling connection options are located on the [Pooling tab](#) of the driver Setup dialog box. See ["Using DataDirect Connection Pooling" on page 82](#) for a general description of connection pooling and its implementation.

## DataDirect Bulk Load

The driver supports DataDirect bulk load and its related connection options. Bulk load connection options are located on the [Bulk tab](#) of the driver Setup dialog box. See ["Using DataDirect Bulk Load" on page 85](#) for a general description of DataDirect bulk load and its implementation.

For Sybase, some additional database configuration is required when the destination table for a bulk load operation does not have an index defined. If you are using a destination table that does not have an index defined, you can ask the database operator to execute the following commands:

```
use master
sp_dboption test, "select into/bulkcopy/pllsort", true
```

This option is required to perform operations that do not keep a complete record of the transaction in the log. For more information, refer to the Sybase documentation.

Alternatively, you can define an index on the destination table.

Failure to properly configure the database results in errors such as the following:

```
"You cannot run the non-logged version of bulk copy in this database. Please
check with the DBO."
```

Bulk Copy Operations and Transactions

Sybase does not support a bulk insert within a transaction, and returns an error if a bulk copy operation is attempted in the scope of an existing transaction.

The Sybase server treats each batch of the bulk copy operation as a single transaction. If any rows in the batch are rejected, the entire transaction is rolled back.

Performance Considerations

Sybase defines two bulk copy modes, described in [Table 12-4](#). Sybase automatically selects the appropriate mode at run time. For more information, refer to your Sybase documentation.

Table 12-4. Summary of Fast and Slow Bulk Copy Mode Characteristics		
Characteristic	Fast Bulk Copy Mode	Slow Bulk Copy Mode
Destination Table Characteristics	No indexes or triggers on destination table	One or more indexes or triggers
Database Configuration Required	The into/bulkcopy/pllsort dboption must be set to true.	None
Logging Performed	Page allocations are logged, but row inserts are not	Row inserts are logged
Transaction Log Handling	You must dump the database before backing up (dumping) the transaction log.	The transaction log can become very large. After the bulk copy completes, back up your database with dump database, then truncate the log with dump transaction.

Unexpected Characters

Users are sometimes surprised when they insert a character into a database, only to have a different character displayed when they fetch it from the database. There are many reasons this can happen, but it most often involves code page issues, not driver errors.

Client and server machines in a database system each use code pages, which can be identified by a name or a number, such as Shift\_JIS (Japanese) or cp1252 (Windows English). A code page is a mapping that associates a sequence of bits, called a code point, with a specific character. Code pages include the characters and symbols of one or more languages. Regardless of geographical location, a machine can be configured to use a specific code page. Most of the time, a client and database server would use similar, if not identical, code pages. For example, a client and server might use two different Japanese code pages, such as Shift\_JIS and EUC\_JP, but they would still share many Japanese characters in common. These characters might, however, be represented by different code points in each code page. This introduces the need to convert between code pages to maintain data integrity. In some cases, no one-to-one character correspondence exists

between the two code points. This causes a substitution character to be used, which can result in displaying an unexpected character on a fetch.

When the driver on the client machine opens a connection with the database server, the driver determines the code pages being used on the client and the server. This is determined from the Active Code Page on a Windows-based machine. If the client machine is UNIX-based, the driver checks the IANAAppCodePage attribute (see ["IANAAppCodePage" on page 454](#)). If it does not find a specific setting for IACP, it defaults to a value of ISO\_8859\_1.

If the client and server code pages are compatible, the driver transmits data in the code page of the server. Even though the pages are compatible, a one-to-one correspondence for every character may not exist. If the client and server code pages are completely dissimilar, for example, Russian and Japanese, then many substitutions occur because very few, if any, of the characters are mapped between the two code pages.

The following is a specific example of an unexpected character:

- The Windows client machine is running code page cp1252.
- The Sybase server is running code page cp850.
- You insert decimal literals for character data. You think you are inserting LATIN SMALL LETTER I WITH ACUTE (i) and BOX DRAWINGS DOUBLE VERTICAL (||) in the database. When you fetch the data, you see INVERTED EXCLAMATION MARK (¡) and MASCULINE ORDINAL INDICATOR (º) displayed on the client instead.

This occurs because the code points do not correspond in the two code pages. An example of syntax you would use to insert the decimal literals is:

```
CREATE table cp850chars(val text )
INSERT INTO cp850chars values( CHAR(161)+CHAR(186))
```

This effectively inserts the hexadecimal bytes for the numbers 161 (0xA1) and 186 (0xBA) into the text column. Each of these hexadecimal bytes is treated as the single byte code point for the character it represents. The problem is that the character representation for these two particular hexadecimal values is different from code page cp850 to code page cp1252. On cp850, these hexadecimal values represent i (0xA1) and || (0xBA), which is what you thought you were inserting by using the previously described syntax. When you fetch these hexadecimal values, however, the characters displayed on your client machine are ¡ (0xA1) and º (0xBA), because that is what the hexadecimal values represent in code page cp1252. This is not a matter of data corruption or substitution; these hexadecimal values simply represent different values in the two different code pages.

This is not a driver error. It occurs because the code points map differently and because some characters do not exist in a code page. The best way to avoid these problems is to use the same code page on both the client and server machines.



---

## MTS Support

On Windows, the driver can take advantage of Microsoft Transaction Server (MTS) capabilities, specifically, the Distributed Transaction Coordinator (DTC) using the XA Protocol. For a general discussion of MTS and DTC, refer to the help file of the Microsoft Transaction Server SDK.

NOTE: The DataDirect Connect *for* ODBC 32-bit drivers can operate in a 64-bit Windows environment; however, they do not support DTC in this environment. Only the DataDirect Connect64 *for* ODBC 64-bit drivers support DTC in a 64-bit Windows environment.

To enable DTC support, you must be accessing Sybase Adaptive Server Enterprise 12.0 or higher. You can choose either Native OLE and XA protocol distributed transactions. See the Distributed Transaction Model option documented in ["Configuring and Connecting to Data Sources" on page 421](#) for details.

### To enable distributed transaction in the Sybase server:

- 1 Assign the dtm\_tm\_role to each user who will participate in distributed transactions (who will log in to Adaptive Server). You can do this using the sp\_role command. For example:

```
sp_role "grant", dtm_tm_role, user_name
```

In the open string for resource managers, the specified username must have the dtm\_tm\_role.

- 2 Specify a default database other than the master for each user. Sybase cannot start distributed transactions in a master database.

---

## NULL Values

When the Sybase Wire Protocol driver establishes a connection, the driver sets the Sybase database option ansinull to on. Setting ansinull to on ensures that the driver is compliant with the ANSI SQL standard, which makes developing cross-database applications easier.

By default, Sybase does not evaluate NULL values in SQL equality (=), inequity (<>), or aggregate function comparisons in an ANSI SQL-compliant manner. For example, the ANSI SQL specification defines that `col1=NULL` always evaluates to false:

```
SELECT * FROM table WHERE col1 = NULL
```

Using the default database setting (ansinull=off), the same comparison evaluates to true instead of false.

Setting ansinull to on changes the default database behavior so that SQL statements must use `IS NULL` instead of `=NULL`. For example, using the Sybase Wire Protocol driver, if the value of col1 in the following statement is NULL, the comparison evaluates to true:

```
SELECT * FROM table WHERE col1 IS NULL
```

In your application, you can restore the default Sybase behavior for a connection in the following ways:

- Use the Initialization String option to specify the SQL command `set ANSI_NULL off`. For example, the following connection string ensures that the handling of NULL values is restored to the Sybase default for the current connection:

```
DSN=SYB TABLES;DB=PAYROLL;IS=set ANSI_NULL off
```

- Explicitly execute the following statement after the connection is established:

```
SET ANSI_NULL OFF
```

---

## Persisting a Result Set as an XML Data File

The driver allows you to persist a result as an XML data file with embedded schema. See ["Persisting a Result Set as an XML Data File" on page 60](#) for details about implementation.

---

## Isolation and Lock Levels Supported

The Sybase database system supports isolation levels 0 (read uncommitted), 1 (read committed, the default), 2 (repeatable read), and 3 (serializable). It supports page-level locking.

Refer to [Chapter 7 "Locking and Isolation Levels"](#) in the *DataDirect Connect Series for ODBC Reference* for details.

---

## SQL Grammar Support

The driver supports the minimum SQL grammar.

---

## ODBC Conformance Level

The driver is Level 1 compliant, that is, it supports all ODBC Core and Level 1 functions. In addition, the following functions are supported:

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

Refer to [Chapter 2 “ODBC API and Scalar Functions”](#) in the *DataDirect Connect Series for ODBC Reference* for a list of the API functions supported by the Sybase Wire Protocol driver.

---

## Number of Connections and Statements Supported

The Sybase database system supports multiple connections and multiple statements per connection. If the Select Method option on the Performance tab or the connection string attribute SelectMethod is set to 1 (Direct), Sybase data sources are limited to one active statement in manual commit mode.

---

## Using Arrays of Parameters

When designing an application, using parameter arrays for bulk inserts or updates, for example, can improve performance. Refer to [Chapter 5 “Designing ODBC Applications for Performance Optimization”](#) in the *DataDirect Connect Series for ODBC Reference* for more information about using arrays of parameters to improve performance.

Because Sybase databases do not support parameter arrays natively, the Sybase Wire Protocol driver emulates them by sending T-SQL batches of Insert or Update statements to the database, which will improve performance.



# 13 The Oracle Driver

The DataDirect Connect *for* ODBC and DataDirect Connect64 *for* ODBC Oracle driver (the Oracle driver) each support the following Oracle database servers when using the appropriate client software:

- Oracle 11g R1, R2 (11.1, 11.2)
- Oracle 10g R1, R2 (10.1, 10.2)
- Oracle 9i R1, R2 (9.0.1, 9.2)
- Oracle 8i R3 (8.1.7)

The Oracle driver is supported in the Windows, UNIX, and Linux environments. See ["Environment-Specific Information" on page 45](#) for detailed information about the environments supported by this driver.

Refer to the readme file shipped with your DataDirect Connect product for the file name of the Oracle driver.

NOTE: The Oracle driver requires Oracle client software. Progress DataDirect also provides an Oracle driver that does not require any client software to access Oracle databases. See ["The Oracle Wire Protocol Driver" on page 241](#) for details.

---

## Driver Requirements

This section provides the system requirements for using the Oracle driver on Windows, UNIX, and Linux.

IMPORTANT: You must have all components of the Oracle client software installed; otherwise, the driver will not operate properly. You must have the appropriate DLLs or shared libraries and objects on your path.

Although an earlier version of a client can access a later version of a database, for example, client 9i to server 10g, to ensure that you have access to all of the features of a particular database, you should use the client that matches the database version, for example, client 10g to server 10g.

NOTE: The Oracle driver supports Oracle 10g clients; however, the clients are not available for all operating systems supported by the driver. Consult the Oracle Web site for current client availability.

## Windows

For 32-bit drivers, Oracle Net8 Client 9.2 or higher is required.

For 64-bit drivers, Oracle client software 10.1 or higher is required on x64.

## UNIX and Linux

For 32-bit drivers, Oracle Net8 Client 9.2 or higher is required.

For 64-bit drivers, Oracle client software 9i R2 or higher is required on Linux for Itanium II and UNIX. Oracle client software 10.1 or higher is required for Linux on x64.

Before you can use the Oracle driver, you must have a supported Oracle client installed on your workstation in the \$ORACLE\_HOME source tree. ORACLE\_HOME is an environment variable created by the Oracle installation process that identifies the location of your Oracle client components.

Set the environment variable ORACLE\_HOME to the directory where you installed the Oracle client. For example, for C-shell users, the following syntax is valid:

```
setenv ORACLE_HOME /databases/oracle
```

For Bourne- or Korn-shell users, the following syntax is valid:

```
ORACLE_HOME=/databases/oracle;export ORACLE_HOME
```

### 32-bit drivers—Building the Required Oracle Net8 Shared Library on HP-UX 11

You must build a replacement shared library for Oracle Net8 Client 9.2 on HP-UX 11. This shared library, libclntsh.sl, contains your unique Oracle Net8 configuration, which is used by the Oracle driver to access local and remote Oracle databases.

The shared library libclntsh.sl is built by the Oracle script genclntsh. The genclntsh script provided by Oracle causes errors resulting from undefined symbols. Run the genclntsh92 script provided by Progress DataDirect to build a replacement libclntsh.sl. This script, in the src/oracle directory, places the new libclntsh.sl in ../../lib, which is your \$ODBC\_HOME/lib directory; it does not overwrite the original libclntsh.sl in the \$ORACLE\_HOME/lib directory.

Before you build the Oracle Net8 shared library, install Oracle and set the environment variable ORACLE\_HOME to the directory where you installed Oracle.

For Oracle Net8 Client 9.2 on HP-UX 11, the following commands build the Oracle Net8 shared library:

```
cd ${ODBC_HOME}/src/oracle
genclntsh92
```

**WARNING:** The \$ODBC\_HOME/lib directory, containing the correct libclntsh library, *must* be on the SHLIB\_PATH *before* \$ORACLE\_HOME/lib. Otherwise, the original Oracle library will be loaded, resulting in the unresolved symbol error.

### Connecting to Oracle 9.2 from HP-UX

To connect to Oracle 9.2 from HP-UX, you must have the HP patch PHSS\_22514 installed on the operating system, and you must set the LD\_PRELOAD system variable to the absolute path of the libjava.sl library.

---

## Configuring and Connecting to Data Sources

After you install the driver, you configure data sources to connect to the database. See [Chapter 1 “Quick Start Connect” on page 35](#) for an explanation of different types of data sources. The data source contains connection options that allow you to tune the driver for specific performance. If you want to use a data source but need to change some of its values, you can either modify the data source or override its values at connection time through a connection string.

If you choose to use a connection string, you must use specific connection string attributes. See [“Using a Connection String” on page 487](#) and [Table 13-1 on page 489](#) for an alphabetical list of driver connection string attributes and their initial default values.

### Data Source Configuration in the UNIX/Linux odbc.ini File

On UNIX and Linux, you must set up the proper ODBC environment before configuring data sources. See [“Environment Configuration” on page 37](#) for basic setup information and [“Environment Variables” on page 97](#) for more detail about this procedure.

Data sources for UNIX and Linux are stored in the system information file (by default, odbc.ini). If you have a Motif GUI environment on Linux, you can configure and modify data sources through the DataDirect ODBC Data Source Administrator for Linux (the Linux ODBC Administrator) using a driver Setup dialog box. (See [“Configuration Through the Administrator” on page 100](#) for a detailed explanation of the Administrator.)

If you do not have a GUI environment, you can configure and modify data sources directly by editing the odbc.ini file and storing default connection values there. See [“Configuration Through the System Information \(odbc.ini\) File” on page 103](#) for detailed information about the specific steps necessary to configure a data source.

[Table 13-1 on page 489](#) lists driver connection string attributes that must be used in the odbc.ini file to set the value of the attributes. Note that only the long name of the attribute can be used in the file. The default listed in the table is the initial default value when the driver is installed.

### Data Source Configuration through a GUI

On Windows, data sources are stored in the Windows Registry. You can configure and modify data sources through the ODBC Administrator using a driver Setup dialog box, as described in this section.

On UNIX and Linux, data sources are stored in the `odbc.ini` file. On Linux, you can configure and modify data sources through the Linux ODBC Administrator using a driver Setup dialog box, as described in this section.

NOTE: This book shows dialog box images that are specific to Windows. If you are using the drivers in the Linux environment, the dialog box that you see may differ slightly from the Windows version. Windows-only and UNIX-only connection options are specifically noted by icons in the Setup dialog box descriptions.

When the driver is first installed, the values of its connection options are set by default. These values appear on the driver Setup dialog box tabs when you create a new data source. You can change these default values by modifying the data source. In the following procedure, the description of each tab is followed by a table that lists the connection options for that tab and their initial default values. This table links you to a complete description of the options and their connection string attribute equivalents. The connection string attributes are used to override the default values of the data source if you want to change these values at connection time.

#### To configure an Oracle data source:

##### 1 Start the ODBC Administrator:

- On Windows, start the ODBC Administrator by selecting its icon from the DataDirect Connect program group.
- On Linux, change to the `install_dir/tools` directory and, at a command prompt, enter:

```
odbcadmin
```

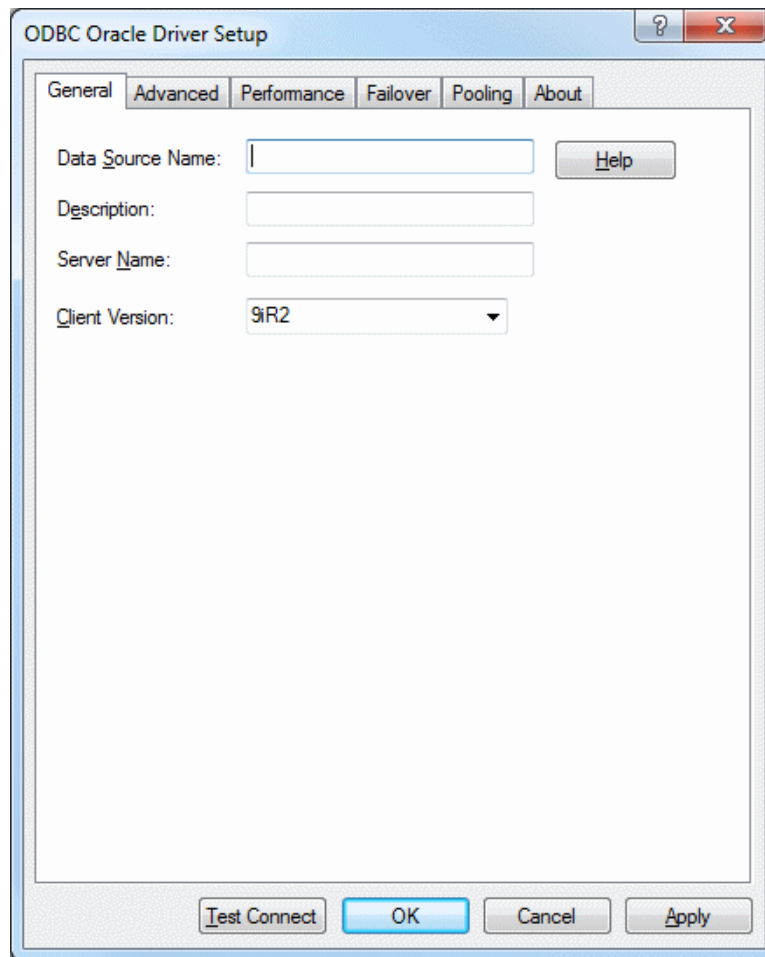
where `install_dir` is the path to the product installation directory.

##### 2 Select a tab:

- **User DSN:** If you are configuring an existing user data source, select the data source name and click **Configure** to display the driver Setup dialog box.  
  
If you are configuring a new user data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.
- **System DSN:** If you are configuring an existing system data source, select the data source name and click **Configure** to display the driver Setup dialog box.  
  
If you are configuring a new system data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.
- **File DSN:** If you are configuring an existing file data source, select the data source file and click **Configure** to display the driver Setup dialog box.  
  
If you are configuring a new file data source, click **Add** to display a list of installed drivers; then, select a driver. Click **Advanced** if you want to specify attributes; otherwise, click **Next** to proceed. Specify a name for the data source and click **Next**. Verify the data source information; then, click **Finish** to display the driver Setup dialog box.



The General tab of the Setup dialog box appears by default.

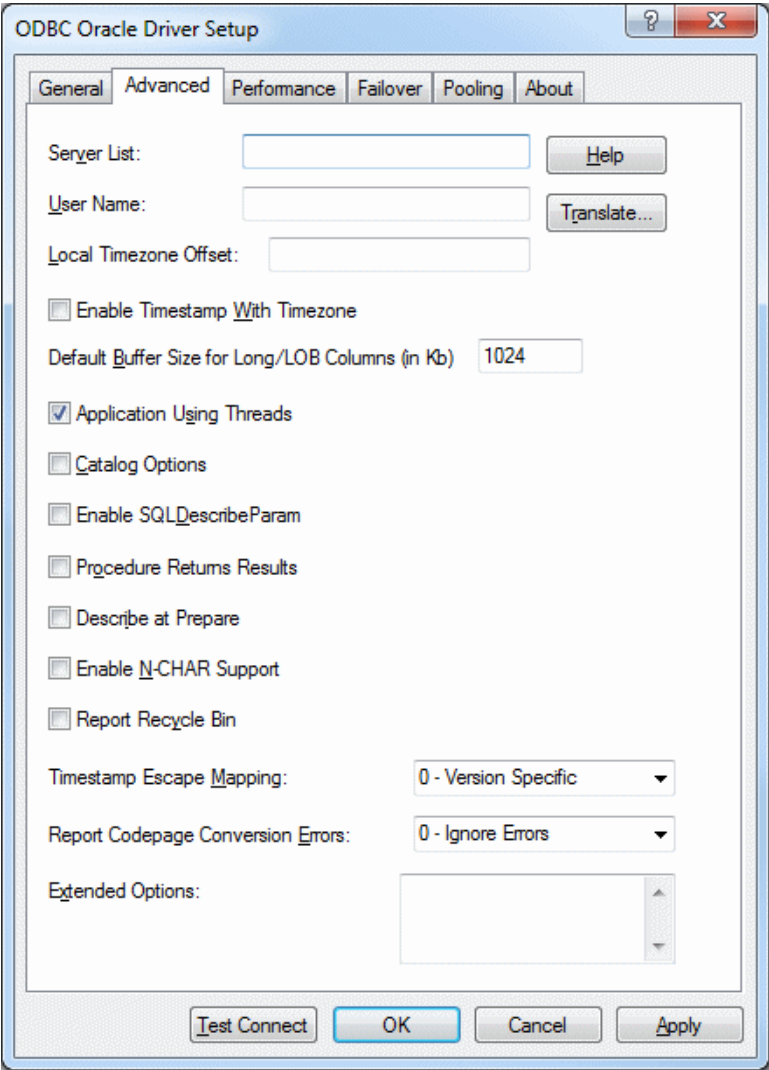


NOTE: The General tab displays only fields that are required for creating a data source. The fields on all other tabs are optional, unless noted otherwise.

- 3 On this tab, provide values for the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: General	Default
<a href="#">Data Source Name (see page 494)</a>	None
<a href="#">Description (see page 495)</a>	None
<a href="#">Server Name (see page 503)</a>	None
<a href="#">Client Version (see page 492)</a>	9iR2

- 4 Optionally, click the **Advanced** tab to specify additional data source settings.



On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Advanced	Default
<a href="#">Server List (see page 502)</a>	None
<a href="#">User Name (see page 504)</a>	None
<a href="#">Local Timezone Offset (see page 499)</a>	None
<a href="#">Enable Timestamp with Timezone (see page 497)</a>	Disabled
<a href="#">Default Buffer Size for Long/LOB Columns (in Kb) (see page 494)</a>	1024
<a href="#">Application Using Threads (see page 491)</a>	Enabled
<a href="#">Catalog Options (see page 492)</a>	Disabled
<a href="#">Enable SQLDescribeParam (see page 496)</a>	Disabled
<a href="#">Procedure Returns Results (see page 501)</a>	Disabled
<a href="#">Describe At Prepare (see page 495)</a>	Disabled

**Connection Options: Advanced**

	<b>Default</b>
<a href="#">Enable N-CHAR Support (see page 495)</a>	Disabled
<a href="#">Report Recycle Bin (see page 502)</a>	Disabled
<a href="#">Timestamp Escape Mapping (see page 503)</a>	0 - Version Specific
<a href="#">Report Codepage Conversion Errors (see page 501)</a>	0 - Ignore Errors
<a href="#">IANAAppCodePage (see page 497)</a>	4 (ISO 8559-1 Latin-1)
UNIX ONLY	

**Extended Options:** Type a semi-colon separated list of connection options and their values. Use this configuration option to set the value of undocumented connection options that are provided by Progress DataDirect customer support. You can include any valid connection option in the Extended Options string, for example:

```
Database=Server1;UndocumentedOption1=value[;UndocumentedOption2=value;]
```

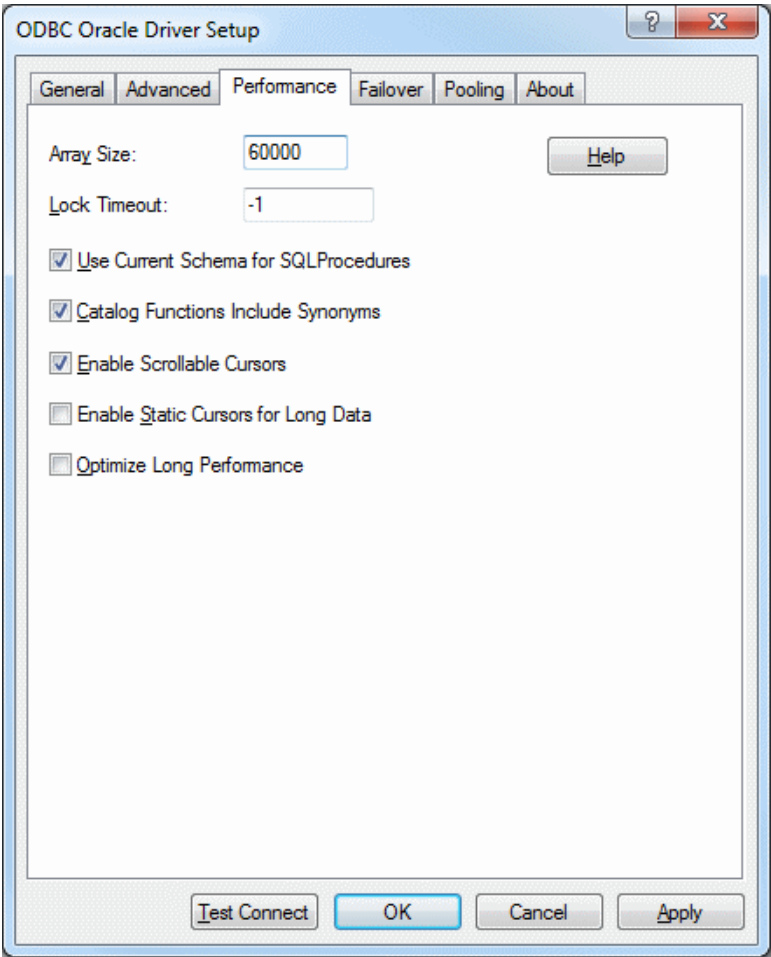
If the Extended Options string contains option values that are also set in the setup dialog or data source, the values of the options specified in the Extended Options string take precedence. However, connection options that are specified on a connection string override any option value specified in the Extended Options string.

**NOTE:** Do not specify the Extended Options configuration option in a connection string, or the driver will return an error. Instead, applications should specify the individual undocumented connection options in the connection string.

**Translate:** Click **Translate** to display the Select Translator dialog box, which lists the translators specified in the ODBC Translators section of the Registry. Progress 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.

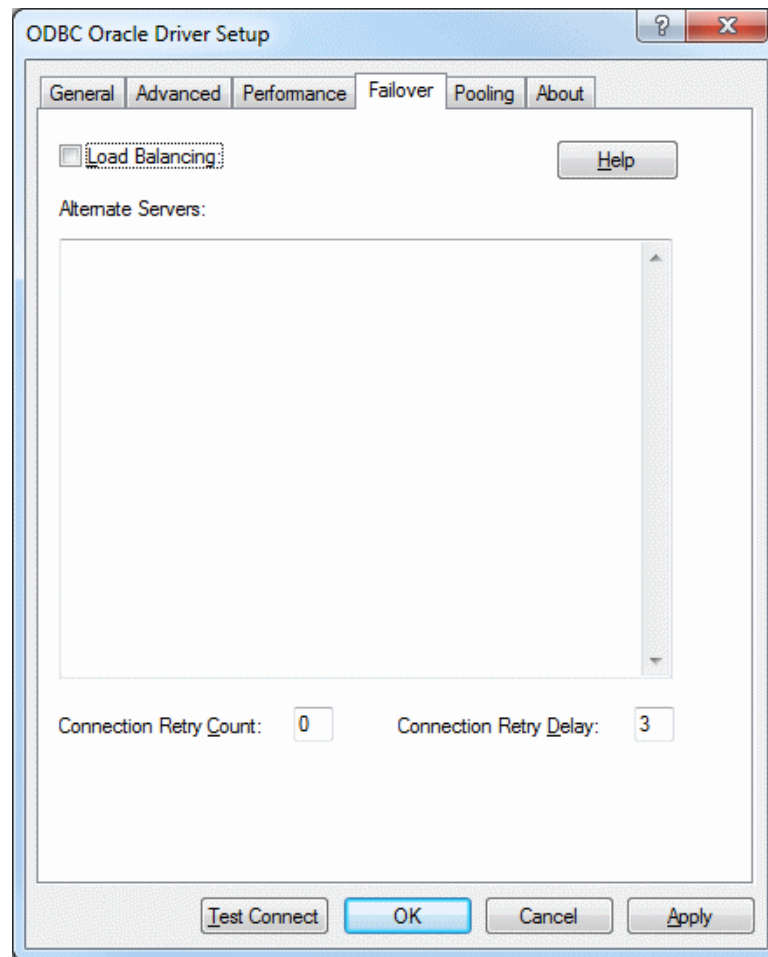
- 5 Optionally, click the **Performance** tab to specify performance data source settings.



On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Performance	Default
<a href="#">Array Size (see page 491)</a>	60000
<a href="#">Lock Timeout (see page 499)</a>	-1
<a href="#">Use Current Schema for SQLProcedures (see page 504)</a>	Enabled
<a href="#">Catalog Functions Include Synonyms (see page 491)</a>	Enabled
<a href="#">Enable Scrollable Cursors (see page 496)</a>	Enabled
<a href="#">Enable Static Cursors for Long Data (see page 497)</a>	Disabled
<a href="#">Optimize Long Performance (see page 500)</a>	Disabled

- 6 Optionally, click the **Failover** tab to specify failover data source settings.

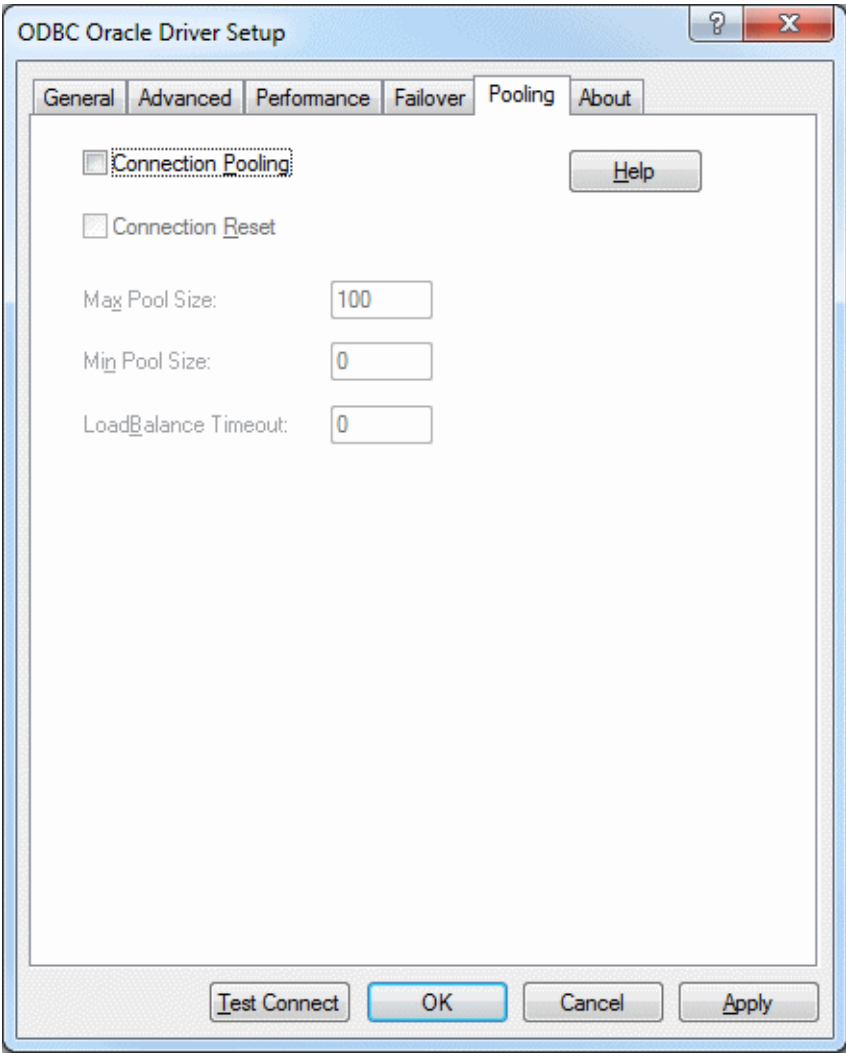


See ["Using Failover" on page 65](#) for a general description of failover and its related connection options.

On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Failover	Default
<a href="#">Load Balancing (see page 498)</a>	Disabled
<a href="#">Alternate Servers (see page 490)</a>	None
<a href="#">Connection Pooling (see page 493)</a>	0
<a href="#">Connection Retry Delay (see page 494)</a>	3

- 7 Optionally, click the **Pooling** tab to specify connection pooling data source settings.



See ["Using DataDirect Connection Pooling" on page 82](#) for a general description of connection pooling.

On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Pooling	Default
<a href="#">Connection Pooling (see page 493)</a>	Disabled
<a href="#">Connection Reset (see page 493)</a>	Disabled
<a href="#">Max Pool Size (see page 500)</a>	100
<a href="#">Min Pool Size (see page 500)</a>	0
<a href="#">Load Balance Timeout (see page 498)</a>	0

- 8 At any point during the configuration process, you can click **Test Connect** to attempt to connect to the data source using the connection options specified in the driver Setup dialog box. A logon dialog box appears (see ["Using a Logon Dialog Box" on page 488](#) for details). 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 incorrect environment or connection value, it displays an appropriate error message. Click **OK**.

NOTE: If you are configuring alternate servers for use with the connection failover feature, be aware that the Test Connect button tests only the primary server, not the alternate servers.

- 9 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.

## Using a Connection String

If you want to use a connection string for connecting to a database, or if your application requires it, you must specify either a DSN (data source name), a File DSN, or a DSN-less connection in the string. The difference is whether you use the `DSN=`, `FILEDSN=`, or the `DRIVER=` keyword in the connection string, as described in the ODBC specification. A DSN or FILEDSN connection string tells the driver where to find the default connection information. Optionally, you may specify *attribute=value* pairs in the connection string to override the default values stored in the data source.

The DSN connection string has the form:

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

The FILEDSN connection string has the form:

```
FILEDSN=filename.dsn[;attribute=value[;attribute=value]...]
```

The DSN-less connection string specifies a driver instead of a data source. All connection information must be entered in the connection string because the information is not stored in a data source.

The DSN-less connection string has the form:

```
DRIVER=[{driver_name}] [;attribute=value[;attribute=value]...]
```

Table 13-1 lists the long and short names for each attribute, as well as the initial default value when the driver is first installed. You can specify either long or short names in the connection string.

An example of a DSN connection string with overriding attribute values for Oracle Wire Protocol is:

```
DSN=Accounting;SRVR=QESVR;UID=JOHN;PWD=XYZZY
```

A FILEDSN connection string is similar except for the initial keyword:

```
FILEDSN=Oracle.dsn;SRVR=QESVR;UID=JOHN;PWD=XYZZY
```

A DSN-less connection string must provide all necessary connection information:

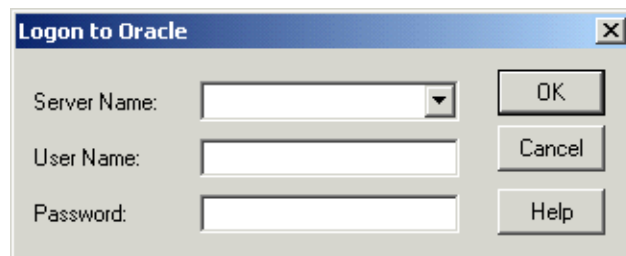
```
DRIVER=DataDirect 7.1 Oracle;SRVR=QESVR;CV=10GR1;UID=JOHN;PWD=XYZZY
```

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

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

## 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, provide the following information:

- 1 In the Server Name field, type the client connection string of the computer containing the Oracle database tables you want to access. Or, select the string from the Server Name drop-down list, which displays the names you specified in the ODBC Oracle driver Setup dialog box.

For local servers, use the SQL\*Net connection string. If the SQL\*Net connection string contains semicolons, enclose it in quotation marks. Refer to your SQL\*Net documentation for more information.

For remote servers, the Oracle TNS Client connection string is the alias name of the Oracle Listener on your network.

- 2 If required, type your Oracle user name.



- 3 If required, type your Oracle password.
- 4 Click **OK** to log on to the Oracle database installed on the server you specified and to update the values in the Registry.

NOTE: You can also use OS Authentication to connect to an Oracle database. See ["OS Authentication" on page 511](#) for details.

## Connection Option Descriptions

The following connection option descriptions are listed alphabetically by the GUI name that appears on the driver Setup dialog box. The connection string attribute name, along with its short name, is listed immediately underneath the GUI name. For example:

### Application Using Threads

Attribute ApplicationUsingThreads (AUT)

In most cases, the GUI name and the attribute name are the same; however, some exceptions exist. If you need to look up an option by its connection string attribute name, please refer to the alphabetical table of connection string attribute names.

Also, a few connection string attributes, for example, Password, do not have equivalent options that appear on the GUI. They are in the list of descriptions alphabetically by their attribute names.

[Table 13-1](#) lists the connection string attributes supported by the Oracle driver.

**Table 13-1. Oracle Attribute Names**

Attribute (Short Name)	Default
<a href="#">AlternateServers (ASRV)</a>	None
<a href="#">ApplicationUsingThreads (AUT)</a>	1 (Enabled)
<a href="#">ArraySize (AS)</a>	60000
<a href="#">CatalogIncludesSynonyms (CIS)</a>	1 (Enabled)
<a href="#">CatalogOptions (CO)</a>	0 (Disabled)
<a href="#">ClientVersion (CV)</a>	9iR2
<a href="#">ConnectionReset (CR)</a>	0 (Disabled)
<a href="#">ConnectionRetryCount (CRC)</a>	0
<a href="#">ConnectionRetryDelay (CRD)</a>	3
<a href="#">DataSourceName (DSN)</a>	None
<a href="#">DefaultLongDataBuffLen (DLDBL)</a>	1024
<a href="#">DescribeAtPrepare (DAP)</a>	0 (Disabled)
<a href="#">Description (n/a)</a>	Empty
<a href="#">EnableDescribeParam (EDP)</a>	0 (Disabled)
<a href="#">EnableNcharSupport (ENS)</a>	0 (Disabled)

**Table 13-1. Oracle Attribute Names** (cont.)

Attribute (Short Name)	Default
<a href="#">EnableScrollableCursors (ESC)</a>	1 (Enabled)
<a href="#">EnableStaticCursorsForLongData (ESCLD)</a>	0 (Disabled)
<a href="#">EnableTimestampwithTimezone (ETWT)</a>	0 (Disabled)
<a href="#">IANAAppCodePage (IACP)</a>	4 (ISO 8559-1 Latin-1)
UNIX ONLY	
<a href="#">LoadBalanceTimeout (LBT)</a>	0 (Disabled)
<a href="#">LoadBalancing (LB)</a>	0 (Disabled)
<a href="#">LocalTimezoneOffset (LTZO)</a>	None
<a href="#">LockTimeout (LTO)</a> (see page 499)	-1
<a href="#">LogonID (UID)</a>	None
<a href="#">MaxPoolSize (MXPS)</a>	100
<a href="#">MinPoolSize (MNPS)</a>	0
<a href="#">OptimizeLongPerformance (OLP)</a>	0 (Disabled)
<a href="#">Password (PWD)</a>	Empty
<a href="#">Pooling (POOL)</a>	0 (Disabled)
<a href="#">ProcedureRetResults (PRR)</a>	0 (Disabled)
<a href="#">ReportCodepageConversionErrors (RCCE)</a>	0 (Ignore Errors)
<a href="#">ReportRecycleBin (RRB)</a>	0 (Disabled)
<a href="#">ServerList</a>	None
<a href="#">ServerName (SRVR)</a>	None
<a href="#">TimestampEscapeMapping (TEM)</a>	0 (Version Specific)
<a href="#">UseCurrentSchema (UCS)</a>	1 (Enabled)

## Alternate Servers

Attribute	AlternateServers (ASRV)
Description	A list of alternate database servers to which the driver tries to connect if the primary database server is unavailable. Specifying a value for this option enables connection failover for the driver. The value you specify must be in the form of a string that defines the physical location of each alternate server. All of the other required connection information for each alternate server is the same as what is defined for the primary server connection.
Valid Values	<p>(ServerName=servervalue[, . . .])</p> <p>You must specify the server name of each alternate server.</p>
Example	<p>The following Alternate Servers value defines two alternate database servers for connection failover:</p> <pre>(ServerName=AcctBackup1, ServerName=AcctBackup2)</pre>

Default None  
 GUI tab [Failover tab](#) on page 485

## Application Using Threads

Attribute ApplicationUsingThreads (AUT)  
 Description Determines whether the driver works with applications using multiple ODBC threads.  
 This connection option can affect performance. See ["Performance Considerations" on page 504](#) for details.  
 Valid Values 0 | 1  
 If set to 1 (Enabled), the driver works with single-threaded and multi-threaded applications.  
 If set to 0 (Disabled), the driver does not work with multi-threaded applications. If using the driver with single-threaded applications, this value avoids additional processing required for ODBC thread-safety standards.  
 Default 1 (Enabled)  
 GUI tab [Advanced tab](#) on page 482

## Array Size

Attribute ArraySize (AS)  
 Description The number of bytes the driver can fetch in a single network round trip. 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.  
 This connection option can affect performance. See ["Performance Considerations" on page 504](#) for details.  
 Valid Values An integer from 1 to 4,294,967,296 (4 GB)  
 The value 1 does not define the number of bytes but, instead, causes the driver to allocate space for exactly one row of data.  
 Default 60000  
 GUI Tab [Performance tab](#) on page 483

## Catalog Functions Include Synonyms

Attribute CatalogIncludesSynonyms (CIS)  
 Description Determines whether synonyms are included in calls to SQLProcedures, SQLStatistics, and SQLProcedureColumns.  
 This connection option can affect performance. See ["Performance Considerations" on page 504](#) for details.

Valid Values 0 | 1

If set to 1 (Enabled), synonyms are included in calls to SQLProcedures, SQLStatistics, and SQLProcedureColumns.

If set to 0 (Disabled), synonyms are excluded (a non-standard behavior) and performance is thereby improved.

Default 1 (Enabled)

GUI Tab [Performance tab](#) on page 483

## Catalog Options

Attribute CatalogOptions (CO)

Description Determines whether SQL\_NULL\_DATA is returned for the result columns REMARKS and COLUMN\_DEF.

This connection option can affect performance. See ["Performance Considerations" on page 504](#) for details.

Valid Values 0 | 1

If set to 1 (Enabled), the result column REMARKS (for the catalog functions SQLTables and SQLColumns) and the result column COLUMN\_DEF (for the catalog function SQLColumns) return actual values. Enabling this option reduces the performance of your catalog (SQLColumns and SQLTables) queries.

If set to 0 (Disabled), SQL\_NULL\_DATA is returned for the result columns REMARKS and COLUMN\_DEF.

Default 0 (Disabled)

GUI Tab [Advanced tab](#) on page 482

## Client Version

Attribute ClientVersion (CV)

Description A value to specify the Oracle client software version. The driver assumes that it is using the version of Oracle client software specified by this option to connect to an Oracle server.

Valid Values 8i | 9iR1 | 9iR2 | 10gR1

When set to 10gR1 and later, the driver binds all non-integer numerics as BINARY FLOAT and BINARY DOUBLE. When set to any Oracle version previous to Oracle10g R1, the driver binds non-integer numerics as if connected to an Oracle 9i R2 or earlier version of the server (regardless of the actual version of the server to which it is connected). When connecting to an Oracle 10g server with a pre-10g client, this attribute must be set to the same version as the actual Oracle client software in use; otherwise, numeric parameter bindings may fail. Versions of the Oracle client software prior to 10g R1 do not fully support the new features of the Oracle 10g database server.

Default 9iR2

GUI Tab [General tab](#) on page 481

## Connection Pooling

Attribute	Pooling (POOL)
Description	Specifies whether to use the driver's connection pooling.  NOTE: The application must be thread-enabled to use connection pooling.  This connection option can affect performance. See <a href="#">"Performance Considerations" on page 504</a> for details.
Valid Values	0   1  If set to 1 (Enabled), the driver uses connection pooling.  If set to 0 (Disabled), the driver does not use connection pooling.
Default	0 (Disabled)
GUI Tab	<a href="#">Pooling tab</a> on page 486

## Connection Reset

Attribute	ConnectionReset (CR)
Description	Determines whether the state of connections that are removed from the connection pool for reuse by the application is reset to the initial configuration of the connection.  This connection option can affect performance. See <a href="#">"Performance Considerations" on page 504</a> for details.
Valid Values	0   1  If set to 1 (Enabled), the state of connections removed from the connection pool for reuse by an application is reset to the initial configuration of the connection. Resetting the state can negatively impact performance because additional commands must be sent over the network to the server to reset the state of the connection.  If set to 0 (Disabled), the state of connections is not reset.
Default	0 (Disabled)
GUI Tab	<a href="#">Pooling tab</a> on page 486

## Connection Retry Count

Attribute	ConnectionRetryCount (CRC)
Description	The number of times the driver retries connection attempts to the primary database server, and if specified, alternate servers until a successful connection is established.  This option and the Connection Retry Delay connection option, which specifies the wait interval between attempts, can be used in conjunction with failover.
Valid Values	0   <i>x</i>  where <i>x</i> is a positive integer from 1 to 65535.  If set to 0, the driver does not try to connect after the initial unsuccessful attempt.

If set to  $x$ , the driver retries connection attempts the specified number of times. If a connection is not established during the retry attempts, the driver returns an error that is generated by the last server to which it tried to connect.

Default 0

GUI Tab [Failover tab](#) on page 485

## Connection Retry Delay

Attribute ConnectionRetryDelay (CRD)

Description The number of seconds the driver waits between connection retry attempts when Connection Retry Count is set to a positive integer.

This option and the Connection Retry Count connection option can be used in conjunction with failover.

Valid Values 0 |  $x$

where  $x$  is a positive integer from 1 to 65535.

If set to 0, there is no delay between retries.

If set to  $x$ , the driver waits the specified number of seconds between connection retry attempts.

Default 3

GUI Tab [Failover tab](#) on page 485

## Data Source Name

Attribute DataSourceName (DSN)

Description The name of a data source in your Windows Registry or odbc.ini file.

Valid Values *string*

where *string* is the name of a data source.

Default None

GUI Tab [General tab](#) on page 481

## Default Buffer Size for Long/LOB Columns (in Kb)

Attribute DefaultLongDataBuffLen (DLDBL)

Description The maximum length of data (in KB) the driver can fetch from long columns in a single round trip and the maximum length of data that the driver can send using the SQL\_DATA\_AT\_EXEC parameter.

NOTE: If this option is enabled, the Optimize Long Performance option is ignored.

This connection option can affect performance. See ["Performance Considerations" on page 504](#) for details.

Valid Values	An integer in multiples of 1024
	The value must be in multiples of 1024 (for example, 1024, 2048). You need to increase the default value if the total size of any Long data exceeds 1 MB. This value is multiplied by 1024 to determine the total maximum length of fetched data. For example, if you enter a value of 2048, the maximum length of data would be 1024 x 2048, or 2097152 (2 MB).
Default	1024
GUI Tab	<a href="#">Advanced tab</a> on page 482

## Describe At Prepare

Attribute	DescribeAtPrepare (DAP)
Description	Determines whether the driver describes the SQL statement at prepare time.
	This connection option can affect performance. See <a href="#">"Performance Considerations" on page 504</a> for details.
Valid Values	0   1
	If set to 1 (Enabled), the driver describes the SQL statement at prepare time.
	If set to 0 (Disabled), the driver does not describe the SQL statement at prepare time.
Default	0 (Disabled)
GUI Tab	<a href="#">Advanced tab</a> on page 482

## Description

Attribute	Description (n/a)
Description	An optional long description of a data source. This description is not used as a runtime connection attribute, but does appear in the ODBC.INI section of the Registry and in the odbc.ini file.
Valid Values	<i>string</i>
	where <i>string</i> is a description of a data source.
Default	None
GUI Tab	<a href="#">General tab</a> on page 481

## Enable N-CHAR Support

Attribute	EnableNcharSupport (ENS)
Description	Determines whether the driver provides support for the N-types NCHAR, NVARCHAR2, and NCLOB. These types are described as SQL_WCHAR, SQL_WVARCHAR, and SQL_WLONGVARCHAR, and are returned as supported by SQLGetTypeInfo. In addition, the "normal" char types (char, varchar2, long, clob) are described as SQL_CHAR, SQL_VARCHAR, and SQL_LONGVARCHAR regardless of the character set on the Oracle server.

See ["Unicode Support" on page 509](#) for details.

NOTE: Valid only on Oracle 9i and higher.

Valid Values 0 | 1

If set to 1 (Enabled), the driver provides support for the N-types NCHAR, NVARCHAR2, and NCLOB.

If set to 0 (Disabled), the driver does not provide support for the N-types NCHAR, NVARCHAR2, and NCLOB.

Default 0 (Disabled)

GUI Tab [Advanced tab](#) on page 482

## Enable Scrollable Cursors

Attribute EnableScrollableCursors (ESC)

Description Determines whether scrollable cursors, both Keyset and Static, are enabled for the data source.

This connection option can affect performance. See ["Performance Considerations" on page 504](#) for details.

Valid Values 0 | 1

If set to 1 (Enabled), scrollable cursors are enabled for the data source.

If set to 0 (Disabled), scrollable cursors are not enabled.

Default 1 (Enabled)

GUI Tab [Performance tab](#) on page 483

## Enable SQLDescribeParam

Attribute EnableDescribeParam (EDP)

Description Determines whether the SQLDescribeParam function describes all parameters with a data type of SQL\_VARCHAR for Select statements. For Insert/Update/Delete statements and for stored procedures, the parameters are described as the actual Oracle data types on the Oracle server. This option must be enabled to access data when using Microsoft Remote Data Objects (RDO).

Valid Values 0 | 1

If set to 1 (Enabled), the SQLDescribeParam function describes all parameters with a data type of SQL\_VARCHAR for Select statements.

If set to 0 (Disabled), the SQLDescribeParam function does not describe all parameters with a data type of SQL\_VARCHAR for Select statements.

Default 0 (Disabled)

GUI Tab [Advanced tab](#) on page 482



## Enable Static Cursors for Long Data

Attribute	EnableStaticCursorsForLongData (ESCLD)
Description	Determines whether the driver supports Long columns when using a static cursor. Enabling this option causes a performance penalty at the time of execution when reading Long data.  This connection option can affect performance. See <a href="#">"Performance Considerations" on page 504</a> for details.
Valid Values	0   1  If set to 1 (Enabled), the driver supports Long columns when using a static cursor.  If set to 0 (Disabled), the driver does not support Long columns when using a static cursor.  NOTE: You must enable this option if you want to persist a result set that contains Long data into an XML data file.
Default	0 (Disabled)
GUI Tab	<a href="#">Performance tab</a> on page 483

## Enable Timestamp with Timezone

Attribute	EnableTimestampwithTimezone (ETWT)
Description	Determines whether the driver exposes timestamps with timezones to the application.
Valid Values	0   1  If set to 1 (Enabled), the driver exposes timestamps with timezones to the application. The driver issues an ALTER SESSION at connection time to modify NLS_TIMESTAMP_TZ_FORMAT. NLS_TIMESTAMP_TZ_FORMAT is changed to the ODBC definition of a timestamp literal with the addition of the timezone literal: 'YYYY-MM-DD HH24:MI:SSXFF TZR'.  If set to 0 (Disabled), timestamps with timezones are not exposed to the application.
Default	0 (Disabled)
GUI Tab	<a href="#">Advanced tab</a> on page 482

## IANAAppCodePage

Attribute	IANAAppCodePage (IACP)
Description	An Internet Assigned Numbers Authority (IANA) value. You must specify a value for this option if your application is not Unicode-enabled or if your database character set is not Unicode. Refer to <a href="#">Chapter 4 "Internationalization, Localization, and Unicode"</a> in the <i>DataDirect Connect Series for ODBC Reference</i> for details.  The driver uses the specified IANA code page to convert "W" (wide) functions to ANSI.

The driver and Driver Manager both check for the value of IANAAppCodePage in the following order:

- In the connection string
- In the Data Source section of the system information file (odbc.ini)
- In the ODBC section of the system information file (odbc.ini)

If the driver does not find an IANAAppCodePage value, the driver uses the default value of 4 (ISO 8859-1 Latin-1).

Valid Values *IANA\_code\_page*

where *IANA\_code\_page* is one of the valid values listed in [Chapter 1 “Values for the Attribute IANAAppCodePage”](#) in the *DataDirect Connect Series for ODBC Reference*. The value must match the database character encoding and the system locale.

Default 4 (ISO 8559-1 Latin-1)

GUI Tab [Advanced tab](#) on page 482

**Load Balance Timeout**

Attribute LoadBalanceTimeout (LBT)

Description The number of seconds to keep inactive connections open in a connection pool. An inactive connection is a database session that is not associated with an ODBC connection handle, that is, a connection in the pool that is not in use by an application.

NOTE: The Min Pool Size option may cause some connections to ignore this value.

This connection option can affect performance. See ["Performance Considerations" on page 504](#) for details.

Valid Values 0 | *x*

where *x* is a positive integer that specifies a number of seconds.

If set to 0, inactive connections are kept open.

If set to *x*, inactive connections are closed after the specified number of seconds passes.

Default 0 (Disabled)

GUI Tab [Pooling tab](#) on page 486

**Load Balancing**

Attribute LoadBalancing (LB)

Description Determines whether the driver uses client load balancing in its attempts to connect to the database servers (primary and alternate). You can specify one or multiple alternate servers by setting the Alternate Servers option.

Valid Values 0 | 1

If set to 1 (Enabled), the driver uses client load balancing and attempts to connect to the database servers (primary and alternate servers) in random order.

If set to 0 (Disabled), the driver does not use client load balancing and connects to each server based on their sequential order (primary server first, then, alternate servers in the order they are specified).

NOTE: This option has no effect unless alternate servers are defined for the Alternate Servers connection option.

Default 0 (Disabled)

GUI Tab [Failover tab](#) on page 485

## Local Timezone Offset

Attribute LocalTimezoneOffset (LTZO)

Description A value to alter local time zone information. The default is "" (empty string), which means that the driver determines local time zone information from the operating system. If it is not available from the operating system, the driver defaults to using the setting on the Oracle server.

Valid Values Valid values are specified as offsets from GMT as follows:  $(-)\text{HH}:\text{MM}$ . For example,  $-08:00$  equals GMT minus 8 hours.

The driver uses the value of this option to issue an ALTER SESSION for local time zone at connection time.

Default "" (Empty String)

GUI Tab [Advanced tab](#) on page 482

## Lock Timeout

Attribute LockTimeout (LTO)

Description Specifies the amount of time, in seconds, the Oracle server waits for a lock to be released before generating an error when processing a Select...For Update statement on an Oracle 9i or higher server.

This connection option can affect performance. See ["Performance Considerations" on page 504](#) for details.

Valid Values  $-1 \mid 0 \mid x$

where  $x$  is an integer that specifies a number of seconds.

If set to -1, the server waits indefinitely for the lock to be released.

If set to 0, the server generates an error immediately and does not wait for the lock to time out.

If set to  $x$ , the server waits for the specified number of seconds for the lock to be released.

NOTE: If you are connected to an Oracle 8i server, any value greater than 0 is equivalent to the value -1.

Default -1

GUI Tab [Performance tab](#) on page 483

## Max Pool Size

Attribute	MaxPoolSize (MXPS)
Description	<p>The maximum number of connections allowed within a single connection pool. When the maximum number of connections is reached, no additional connections can be created in the connection pool.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 504</a> for details.</p>
Valid Values	<p>An integer from 1 to 65535</p> <p>For example, if set to 20, the maximum number of connections allowed in the pool is 20.</p>
Default	100
GUI Tab	<a href="#">Pooling tab</a> on page 486

## Min Pool Size

Attribute	MinPoolSize (MNPS)
Description	<p>The minimum number of connections that are opened and placed in a connection pool, in addition to the active connection, when the pool is created. The connection pool retains this number of connections, even when some connections exceed their Load Balance Timeout value.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 504</a> for details.</p>
Valid Values	<p>0   <math>x</math></p> <p>where <math>x</math> is an integer from 1 to 65535.</p> <p>For example, if set to 5, the start-up number of connections in the pool is 5 in addition to the current existing connection.</p> <p>If set to 0, no connections are opened in addition to the current existing connection.</p>
Default	0
GUI Tab	<a href="#">Pooling tab</a> on page 486

## Optimize Long Performance

Attribute	OptimizeLongPerformance (OLP)
Description	<p>Allows the driver to fetch Long data directly into the application's buffers rather than allocating buffers and making a copy. This option decreases fetch times on Long data; however, it can cause the application to be limited to one active statement per connection.</p> <p>NOTE: If this option is enabled, the Default Buffer Size for Long/LOB Columns option is ignored.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 504</a> for details.</p>

Valid Values	0   1
	If set to 1 (Enabled), the driver fetches Long data directly into the application's buffers rather than allocating buffers and making a copy.
	If set to 0 (Disabled), the driver does not fetch Long data directly into the application's buffers.
Default	0 (Disabled)
GUI Tab	<a href="#">Performance tab</a> on page 483

## Password

Attribute	Password (PWD)
Description	The password that the application uses to connect to your database. The Password option cannot be specified through the driver Setup dialog box and should not be stored in a data source. It is specified through the Logon dialog box or a connection string.
Valid Values	<i>pwd</i>
	where <i>pwd</i> is a valid password.
Default	None
GUI Tab	n/a

## Procedure Returns Results

Attribute	ProcedureRetResults (PRR)
Description	Determines whether the driver returns result sets from stored procedures/functions.  See <a href="#">"MTS Support" on page 511</a> for details.
Valid Values	0   1
	If set to 1 (Enabled), the driver returns result sets from stored procedures/functions. When set to 1 and you execute a stored procedure that does not return result sets, you will incur a small performance penalty.
	If set to 0 (Disabled), the driver does not return result sets from stored procedures.
Default	0 (Disabled)
GUI Tab	<a href="#">Advanced tab</a> on page 482

## Report Codepage Conversion Errors

Attribute	ReportCodepageConversionErrors (RCCE)
Description	Specifies how the driver handles code page conversion errors that occur when a character cannot be converted from one character set to another.

An error message or warning can occur if an ODBC call causes a conversion error, or if an error occurs during code page conversions to and from the database or to and from the application. The error or warning generated is `Code page conversion error encountered`. In the case of parameter data conversion errors, the driver adds the following sentence: `Error in parameter x`, where `x` is the parameter number. The standard rules for returning specific row and column errors for bulk operations apply.

Valid Values 0 | 1 | 2

If set to 0 (Ignore Errors), the driver substitutes 0x1A for each character that cannot be converted and does not return a warning or error.

If set to 1 (Return Error), the driver returns an error instead of substituting 0x1A for unconverted characters.

If set to 2 (Return Warning), the driver substitutes 0x1A for each character that cannot be converted and returns a warning.

Default 0 (Ignore Errors)

GUI Tab [Advanced tab](#) on page 482

## Report Recycle Bin

Attribute ReportRecycleBin (RRB)

Description Determines whether support is provided for reporting objects that are in the Oracle Recycle Bin.

On Oracle 10g R1 and higher, when a table is dropped, it is not actually removed from the database, but placed in the recycle bin instead.

Valid Values 0 | 1

If set to 1 (Enabled), support is provided for reporting objects that are in the Oracle Recycle Bin.

If set to 0 (Disabled), the driver does not return tables contained in the recycle bin in the result sets returned from SQLTables and SQLColumns. Functionally, this means that the driver filters out any results whose Table name begins with BIN\$.

Default 0 (Disabled)

GUI Tab [Advanced tab](#) on page 482

## Server List

Attribute ServerList

Description A list of client connection strings that appear in the logon dialog box. This option applies to GUIs only and is not a runtime connection string attribute.

See ["Performance Considerations" on page 504](#) for details about the logon dialog box.

Valid Values *string*

where *string* is a list of valid client connection strings. Separate the strings with commas. If the client connection string contains a comma, enclose it in quotation marks, for example, "Serv,1", "Serv,2", "Serv,3".

Default None

GUI Tab [Advanced tab](#) on page 482

## Server Name

Attribute ServerName (SRVR)

Description The client connection string of the computer containing the Oracle database tables you want to access.

Valid Values *string*

where *string* is a valid client connection string.

For local servers, use the SQL\*Net connection string. If the SQL\*Net connection string contains semicolons, enclose it in quotation marks. Refer to your SQL\*Net documentation for more information.

For remote servers, the Oracle TNS Client connection string is the alias name of the Oracle Listener on your network.

Default None

GUI Tab [General tab](#) on page 481

## Timestamp Escape Mapping

Attribute TimestampEscapeMapping (TEM)

Description Determines how the driver maps Date, Time, and Timestamp literals.

Valid Values 0 | 1

If set to 0 (Oracle Version Specific), the driver determines whether to use the TO\_DATE or TO\_TIMESTAMP function based on the version of the Oracle server to which it is connected. If the driver is connected to an 8.x server, it maps the Date, Time, and Timestamp literals to the TO\_DATE function. If the driver is connected to a 9.x or higher server, it maps these escapes to the TO\_TIMESTAMP function.

If set to 1 (Oracle 8x Compatible), the driver always uses the Oracle 8.x TO\_DATE function as if connected to an Oracle 8.x server.

Default 0 (Oracle Version Specific)

GUI Tab [Advanced tab](#) on page 482

## Use Current Schema for SQLProcedures

Attribute	UseCurrentSchema (UCS)
Description	Determines whether the driver returns only procedures owned by the current user when executing SQLProcedures.  This connection option can affect performance. See <a href="#">"Performance Considerations" on page 504</a> for details.
Valid Values	0   1  When set to 1 (Enabled), the call for SQLProcedures is optimized, but only procedures owned by the user are returned.  When set to 0 (Disabled), the driver does not specify only the current user.
Default	1 (Enabled)
GUI Tab	<a href="#">Performance tab</a> on page 483

## User Name

Attribute	LogonID (UID)
Description	The default user ID that is used to connect to your database. Your ODBC application may override this value or you may override it in the logon dialog box or connection string.  You can also use OS Authentication to connect to your Oracle database. See <a href="#">"OS Authentication" on page 511</a> for details.
Valid Values	<i>userid</i>  where <i>userid</i> is a valid user ID with permissions to access the database.
Default	None
GUI Tab	<a href="#">Advanced tab</a> on page 482

---

## Performance Considerations

The following connection options can enhance driver performance. You can also enhance performance through efficient application design. Refer to [Chapter 5 "Designing ODBC Applications for Performance Optimization"](#) in the *DataDirect Connect Series for ODBC Reference* for details.

**Application Using Threads (ApplicationUsingThreads):** The driver coordinates concurrent database operations (operations from different threads) by acquiring locks. Although locking prevents errors in the driver, it also decreases performance. If your application does not make ODBC calls from different threads, the driver has no reason to coordinate operations. In this case, the ApplicationUsingThreads attribute should be disabled (set to 0).

NOTE: If you are using a multi-threaded application, you must enable the Application Using Threads option.



**Array Size (ArraySize):** If this connection string attribute is set appropriately, the driver can improve performance of your application by reducing the number of round trips on the network. For example, if your application normally retrieves 200 rows, it is more efficient for the driver to retrieve 200 rows at one time over the network than to retrieve 50 rows at a time during four round trips over the network.

**Catalog Functions Include Synonyms (CatalogIncludesSynonyms):** Standard ODBC behavior is to include synonyms in the result set of calls to the following catalog functions: SQLProcedures, SQLStatistics and SQLProcedureColumns. Retrieving this synonym information degrades performance. If your ODBC application does not need to return synonyms when using these catalog functions, the driver can improve performance if the CatalogIncludesSynonyms attribute is disabled (set to 0).

**Catalog Options (CatalogOptions):** If your application does not need to access the comments/remarks for database tables, performance of your application can be improved. In this case, the CatalogOptions attribute should be disabled (set to 0) because retrieving comments/remarks degrades performance. If this attribute is enabled (set to 1), result column REMARKS (for the catalog functions SQLTables and SQLColumns) and the result column COLUMN\_DEF (for the catalog function SQLColumns) return actual values.

**Default Buffer Size for Long/LOB Columns (DefaultLongDataBuffLen):** To improve performance when your application fetches images, pictures, or long text or binary data, a buffer size can be set to accommodate the maximum size of the data. The buffer size should only be large enough to accommodate the maximum amount of data retrieved; otherwise, performance is reduced by transferring large amounts of data into an oversized buffer. If your application retrieves more than 1 MB of data, the buffer size should be increased accordingly.

**Describe At Prepare (DescribeAtPrepare):** When enabled, this option requires extra network traffic. If your application does not require result set information at prepare time (for instance, you request information about the result set using SQLColAttribute(s), SQLDescribeCol, SQLNumResultCols, and so forth, before calling SQLExecute on a prepared statement), you can increase performance by disabling this option.

**Enable Scrollable Cursors (EnableScrollableCursors) and Enable Static Cursors for Long Data (EnableStaticCursorsForLongData):** When your application uses Static or Keyset (Scrollable) cursors, the EnableScrollableCursors attribute must be enabled (set to 1). Also, if your application retrieves images, pictures, long text or binary data while using Static cursors, the EnableStaticCursorsForLongData attribute must be enabled (set to 1). However, this can degrade performance when retrieving long data with Static cursors as the entire result set is stored on the client. To improve performance, you might consider designing your application to retrieve long data through forward-only cursors.

**Lock Timeout (LockTimeOut):** Sometimes users attempt to select data that is locked by another user. Oracle provides three options when accessing locked data with SELECT ... FOR UPDATE statements:

- Wait indefinitely for the lock to be released (-1)
- Return an error immediately (0)
- Return an error if the lock has not been released within a specific number of seconds (*n* seconds)

NOTE: This option is not available with Oracle 8.

Some applications may benefit by not waiting indefinitely and continuing execution; this keeps the application from hanging. The application, however, needs to handle lock timeouts properly with an appropriate timeout value; otherwise, processing time could be wasted handling lock timeouts, and deadlocks could go undetected.

To improve performance, either enter a number of seconds or enter 0 as the value for this option.

**Optimize Long Performance (OptimizeLongPerformance):** When enabled, this option fetches Long data directly into the application's buffers rather than allocating buffers and making a copy. Also, when enabled, this option decreases fetch times on Long data; however, it can cause the application to be limited to one active statement per connection.

**Procedure Returns Results (ProcedureRetResults):** The driver can be tuned for improved performance if your application's stored procedures do not return results. In this case, the ProcedureRetResults attribute should be disabled (set to 0).

**Use Current Schema for SQLProcedures (UseCurrentSchema):** If your application needs to access database objects owned only by the current user, performance of your application can be improved. In this case, the UseCurrentSchema attribute should be enabled (set to 1). When this attribute is enabled, the driver returns only database objects owned by the current user when executing catalog functions. Calls to catalog functions are optimized by grouping queries. Enabling this attribute is equivalent to passing the Logon ID used on the connection as the SchemaName argument to the catalog functions.

# Data Types

Table 13-2 shows how the Oracle data types are mapped to the standard ODBC data types. "Unicode Support" on page 509 lists Oracle to Unicode data type mappings.

Table 13-2. Oracle Data Types	
Oracle	ODBC
BFILE <sup>1</sup>	SQL_LONGVARIABLE
BINARY DOUBLE <sup>2</sup>	SQL_REAL
BINARY FLOAT <sup>2</sup>	SQL_DOUBLE
BLOB <sup>3</sup>	SQL_LONGVARIABLE
CHAR	SQL_CHAR
CLOB <sup>3</sup>	SQL_LONGVARIABLE
DATE	SQL_TYPE_TIMESTAMP
LONG	SQL_LONGVARIABLE
LONG RAW	SQL_LONGVARIABLE
NUMBER	SQL_DOUBLE
NUMBER (p,s)	SQL_DECIMAL
RAW	SQL_VARIABLE

**Table 13-2. Oracle Data Types** (cont.)

Oracle	ODBC
TIMESTAMP <sup>4</sup>	SQL_TIMESTAMP
TIMESTAMP WITH LOCAL TIMEZONE <sup>4</sup>	SQL_TIMESTAMP
TIMESTAMP WITH TIMEZONE <sup>4</sup>	SQL_VARCHAR
VARCHAR2	SQL_VARCHAR
XMLType <sup>5</sup>	SQL_LONGVARCHAR

1. Read-Only

2. Supported only on Oracle 10g and higher.

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

4. Supported only on Oracle 9i and higher.

5. Supported only on Oracle 9i R2 and higher.

The Oracle driver does not support any object types (also known as abstract data types). When the driver encounters an object type during data retrieval, it will return an Unknown Data Type error (SQL State HY000).

See ["Retrieving Data Type Information" on page 59](#) for more information about data types.

## XMLType

Oracle 9i R2 and higher supports the XMLType data type. The driver supports tables containing columns whose data type is specified as XMLType.

When inserting or updating XMLType columns, the data to be inserted or updated must be in the form of an XMLType data type. The database provides functions to construct XMLType data. The `xmlData` argument to `xmltype( )` may be specified as a string literal.

## Examples

If the XMLType column is created with the CLOB storage type, then the driver returns it without use of the special `getClobVal` function, that is, you can use:

```
SELECT XML_col FROM table_name...
```

instead of

```
SELECT XML_col.getClobVal()...
```

The following example illustrates using the CLOB storage type:

```
CREATE TABLE po_xml_tab(
  poid NUMBER(10),
  poDoc XMLTYPE
)
XMLType COLUMN poDoc
STORE AS CLOB (
  TABLESPACE lob_seg_ts
  STORAGE (INITIAL 4096 NEXT 4096)
```

```

        CHUNK 4096 NOCACHE LOGGING
    );

```

The next example illustrates how to create a table, insert data, and retrieve data when not using the CLOB storage type:

```
CREATE TABLE PURCHASEORDER (PODOCUMENT sys.XMLTYPE);
```

The PURCHASEORDER table contains one column—PODOCUMENT—with a data type of XMLType (sys.XMLTYPE). The next step is to insert one purchase order, created by the static function sys.XMLTYPE.createXML:

```

INSERT INTO PURCHASEORDER (PODOCUMENT) values (
    sys.XMLTYPE.createXML(
        '
        <PurchaseOrder>
          <Reference>BLAKE-2001062514034298PDT</Reference>

          <Actions>
            <Action>
              <User>KING</User>
              <Date/>
            </Action>
          </Actions>
          <Reject/>

          <Requester>David E. Blake</Requester>
          <User>BLAKE</User>
          <CostCenter>S30</CostCenter>
          <ShippingInstructions>
            <name>David E. Blake</name>
            <address>400 Oracle Parkway Redwood Shores, CA, 94065 USA</address>
            <telephone>650 999 9999</telephone>
          </ShippingInstructions>

          <SpecialInstructions>Air Mail</SpecialInstructions>
          <LineItems>
            <LineItem ItemNumber="1">
              <Description>The Birth of a Nation</Description>
              <Part Id="EE888" UnitPrice="65.39" Quantity="31"/>
            </LineItem>
          </LineItems>
        </PurchaseOrder>
        ');

```

Use the getClobVal function to retrieve the data:

```
SELECT p.podocument.getClobVal() FROM PURCHASEORDER p;
```

# Unicode Support

The Oracle driver uses the NLS\_LANG environment variable setting of the Oracle client to determine how to transmit data to the client.

On Windows, UNIX, and Linux, a Unicode setting is determined if the NLS\_LANG environment variable is set to:

*LANGUAGE\_TERRITORY.CHARSET*

where *CHARSET* is either UTF8, AL24UTF8, or AL32UTF8. For example:

AMERICAN\_AMERICA.UTF8

Alternatively, on Windows, instead of the NLS\_LANG environment variable, the value of the HKEY\_LOCAL\_MACHINE\SOFTWARE\ORACLE\oracle\_home\_key registry key can be set to:

*LANGUAGE\_TERRITORY.CHARSET*

where *oracle\_home\_key* is HOME0 for Oracle 9i R2 and earlier, and is the Oracle home name used at the time of client installation for Oracle 10g.

If the *CHARSET* is a Unicode setting and a Unicode application is accessing the driver, then no data conversion is necessary. If an ANSI application is accessing the driver, then the driver must convert the data from the application from ANSI to Unicode (UTF-8) for the client.

If the *CHARSET* is ANSI and an ANSI application is accessing the driver, then no data conversion is necessary. If a Unicode application is accessing the driver, then the driver must convert the data from the application from Unicode to ANSI for the client.

If NLS\_LANG is set to UTF-8, the Oracle driver maps the Oracle data types to Unicode data types as shown in the following table:

Oracle Data Type	Mapped to . . .
CHAR	SQL_WCHAR
CLOB	SQL_WLONGVARCHAR
VARCHAR2	SQL_WVARCHAR
LONG	SQL_WLONGVARCHAR

The driver also continues to map these Oracle data types to the normal character data types. See ["Data Types" on page 506](#) for these mappings.

The driver supports the Unicode ODBC W (Wide) function calls, such as SQLConnectW. This allows the Driver Manager to transmit these calls directly to the driver. Otherwise, the Driver Manager would incur the additional overhead of converting the W calls to ANSI function calls, and vice versa.

See ["UTF-16 Applications on UNIX and Linux" on page 125](#) for related details. Also, refer to [Chapter 4 "Internationalization, Localization, and Unicode"](#) in the *DataDirect Connect Series for ODBC Reference* for a more detailed explanation of Unicode.

---

## Advanced Features

The driver supports failover and its related connection options. Failover connection options are located on the [Failover tab](#) of the driver Setup dialog box. See ["Using Failover" on page 65](#) for a general description of failover and its implementation.

---

## Unexpected Characters

Users are sometimes surprised when they insert a character into a database, only to have a different character displayed when they fetch it from the database. There are many reasons this can happen, but it most often involves code page issues, not driver errors.

Client and server machines in a database system each use code pages, which can be identified by a name or a number, such as Shift\_JIS (Japanese) or cp1252 (Windows English). A code page is a mapping that associates a sequence of bits, called a code point, with a specific character. Code pages include the characters and symbols of one or more languages. Regardless of geographical location, a machine can be configured to use a specific code page. Most of the time, a client and database server would use similar, if not identical, code pages. For example, a client and server might use two different Japanese code pages, such as Shift\_JIS and EUC\_JP, but they would still share many Japanese characters in common. These characters might, however, be represented by different code points in each code page. This introduces the need to convert between code pages to maintain data integrity. In some cases, no one-to-one character correspondence exists between the two code points. This causes a substitution character to be used, which can result in displaying an unexpected character on a fetch.

When the driver on the client machine opens a connection with the database server, the driver determines the code pages being used on the client and the server. This is determined from the Active Code Page on a Windows-based machine. If the client machine is UNIX-based, the driver checks the IANAAppCodePage option. If it does not find a specific setting for IACP, it defaults to a value of ISO\_8859\_1.

If the client and server code pages are compatible, the driver transmits data in the code page of the server. Even though the pages are compatible, a one-to-one correspondence for every character may not exist. If the client and server code pages are completely dissimilar, for example, Russian and Japanese, then many substitutions occur because very few, if any, of the characters are mapped between the two code pages.

The following is a specific example of an unexpected character:

- The Windows client machine is running code page cp1252.
- The Oracle server is running code page ISO-8859-P1.
- When you insert a Euro character (€) from the Windows client and then fetch it back, an upside down question mark (¿) is displayed on the client instead of the Euro symbol.

This substitution occurs because the Euro character does not exist within the characters defined by the ISO-8859-P1 character set on the Oracle server. The Oracle server records the code point for its substitution character in the table instead of the code point for the Euro. This code point is an upside down question mark in the Windows cp1252 code page.

This is not a driver error. The code page of the Oracle database could not recognize the Euro code point and used its substitution character in the table. The best way to avoid these problems is to use the same code page on both the client and server machines.

You can check the native code point stored in the Oracle database using SQL\*Plus with a SQL statement similar to the following:

```
SELECT dump(columnname, 1016) FROM yourtable;
```

Check the returned hexadecimal values to verify whether the data you intended to reside in the table is there. If it appears that Oracle substituted a different code point, then check the Oracle database code page to see if your intended character exists. If your character does not exist in the code page, then no error is involved; Oracle simply does not recognize the original character, and uses its substitution character instead.

---

## MTS Support

On Windows, the driver can take advantage of Microsoft Transaction Server (MTS) capabilities, specifically, the Distributed Transaction Coordinator (DTC) using the XA Protocol. For a general discussion of MTS and DTC, refer to the help file of the Microsoft Transaction Server SDK.

NOTE: The DataDirect Connect *for* ODBC 32-bit drivers can operate in a 64-bit Windows environment; however, they do not support DTC in this environment. Only the DataDirect Connect64 *for* ODBC 64-bit drivers support DTC in a 64-bit Windows environment.

To enable DTC support, you must be accessing Oracle 8.0.5 or higher servers using Oracle Net8 Client 8.1.6 or higher.

---

## OS Authentication

On Windows, UNIX, and Linux, Oracle has a feature called OS Authentication that allows you to connect to an Oracle database via the operating system user name and password. To connect, use a forward slash ( / ) for the user name and leave the password blank. To configure the Oracle server, refer to the Oracle server documentation. This feature is valid when connecting from a data source, a connection string, or a logon dialog box.

---

## Support for Oracle RAC

Oracle introduced Real Application Clusters (RAC) with Oracle 9i, and RAC is also a key feature of Oracle 10g. Oracle RAC allows a single physical Oracle database to be accessed by concurrent instances of Oracle running across several different CPUs.

An Oracle RAC is composed of a group of independent servers, or nodes, that cooperate as a single system. A cluster architecture such as this provides applications access to more

computing power when needed, while allowing computing resources to be used for other applications when database resources are not as heavily required. For example, in the event of a sudden increase in network traffic, an Oracle RAC can distribute the load over many nodes, a feature referred to as *server load balancing*. Oracle RAC features are available to you simply by connecting to an Oracle RAC system with a DataDirect Connect Series *for* ODBC driver. There is no additional configuration required.

*Connection failover* and *client load balancing* can be used in conjunction with an Oracle RAC system, but they are not specifically part of Oracle RAC. See ["Using Failover" on page 65](#) for details about how these features work in DataDirect Connect Series *for* ODBC drivers.

---

## Support of Materialized Views

When connected to an Oracle 9i or higher server, the Oracle driver supports the creation of materialized views. Materialized views are like any other database view with the following additions: the results are stored as a database object and the results can be updated on a schedule determined by the Create View statement.

Materialized views improve performance for data warehousing and replication. Refer to the Oracle documentation for more information about materialized views.

---

## Stored Procedure Results

When you enable the Procedure Returns Results connection option, the driver returns result sets from stored procedures/functions. In addition, SQLGetInfo(SQL\_MULT\_RESULTS\_SETS) returns Y and SQLGetInfo(SQL\_BATCH\_SUPPORT) returns SQL\_BS\_SELECT\_PROC. If this option is enabled and you execute a stored procedure that does not return result sets, you 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 G1%rowtype;
end GEN_PACKAGE;
Create or replace procedure GEN_PROCEDURE1 (
  rset IN OUT GEN_PACKAGE.GTABLE2CHARCOL, icol INTEGER) as
begin
  open rset for select CHARCOL from GTABLE2
  where INTEGERCOL <= icol order by INTEGERCOL;
end;
```



When executing the stored procedures with result sets, do not include the result set arguments (Oracle ref cursors) in the list of procedure parameters. The result set returned through the ref cursor is returned as a normal ODBC result set.

```
{call GEN_PROCEDURE1 (?)}
```

where ? is the parameter for the icol argument.

For more information, refer to your Oracle SQL documentation.

---

## Persisting a Result Set as an XML Data File

The driver allows you to persist a result as an XML data file with embedded schema. See ["Persisting a Result Set as an XML Data File" on page 60](#) for details about implementation.

NOTE: If you are persisting a result set that contains Long data, you must enable the EnableStaticCursorsforLongData connection string attribute.

---

## Isolation and Lock Levels Supported

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

Refer to [Chapter 7 "Locking and Isolation Levels"](#) in the *DataDirect Connect Series for ODBC Reference* for details.

---

## SQL Support

The driver supports the core SQL grammar.

---

## ODBC Conformance Level

The driver is Level 1 compliant, that is, it supports all ODBC Core and Level 1 functions. In addition, the following functions are supported:

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

Refer to [Chapter 2 “ODBC API and Scalar Functions”](#) in the *DataDirect Connect Series for ODBC Reference* for a list of the API functions supported by the Oracle driver.

---

## Number of Connections and Statements Supported

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

---

## Using Arrays of Parameters

Oracle 8i and higher databases natively support parameter arrays, and the Oracle driver, in turn, supports them when connected to these versions of Oracle databases. When designing an application for performance, using native parameter arrays for bulk inserts or updates, for example, can improve performance. Refer to [Chapter 5 “Designing ODBC Applications for Performance Optimization”](#) in the *DataDirect Connect Series for ODBC Reference* for more information about using arrays of parameters to improve performance.

If the database does not support parameter arrays, the Oracle driver emulates them so that you can design your applications to use arrays of parameters and take advantage of the performance improvements where applicable. The driver emulates parameter arrays by sending individual rows to the database.

# 14 The SQL Server Legacy Wire Protocol Driver

The DataDirect Connect *for* ODBC and DataDirect Connect64 *for* ODBC SQL Server Legacy Wire Protocol driver (the SQL Server Legacy Wire Protocol driver) each support the following Microsoft SQL Server database servers:

- Microsoft SQL Server 2012
- Microsoft SQL Server 2008 R1, R2
- Microsoft SQL Server 2005
- Microsoft SQL Server 2000 Desktop Engine (MSDE 2000)
- Microsoft SQL Server 2000 Enterprise Edition (64-bit)
- Microsoft SQL Server 7.0

The SQL Server Legacy Wire Protocol driver is supported in the Windows, UNIX, and Linux environments. See ["Environment-Specific Information" on page 45](#) for detailed information about the environments supported by this driver.

Refer to the readme file shipped with your DataDirect product for the file name of the SQL Server Legacy Wire Protocol driver.

---

## Driver Requirements

The driver has no client requirements.

### Windows

For support of Microsoft SQL Server 7.0, 2000, and 2005, the driver requires the SQL Server 7.0 versions of Net-Library DLL files, which are installed when you install the SQL Server Legacy Wire Protocol driver. The driver communicates with network software through the SQL Server Net-Library interface.

### UNIX and Linux

To use the SQL Server Legacy Wire Protocol driver on UNIX and Linux, you must have TCP/IP configured on both the UNIX and Linux clients and the Windows server on which the Microsoft SQL Server database resides. The UNIX and Linux SQL Server TCP/IP network client library is built into the SQL Server Legacy Wire Protocol driver on UNIX and Linux.

The Microsoft SQL Server Client configuration has been merged with the ODBC driver configuration and is set in the system information file.

---

## Configuring and Connecting to Data Sources

After you install the driver, you configure data sources to connect to the database. See [Chapter 1 “Quick Start Connect” on page 35](#) for an explanation of different types of data sources. The data source contains connection options that allow you to tune the driver for specific performance. If you want to use a data source but need to change some of its values, you can either modify the data source or override its values at connection time through a connection string.

If you choose to use a connection string, you must use specific connection string attributes. See [“Using a Connection String” on page 520](#), [Table 14-1 on page 523](#), and [Table 14-2 on page 524](#) for an alphabetical list of driver connection string attributes and their initial default values.

### Data Source Configuration in the UNIX/Linux `odbc.ini` File

On UNIX and Linux, you must set up the proper ODBC environment before configuring data sources. See [“Environment Configuration” on page 37](#) for basic setup information and [“Environment Variables” on page 97](#) for more detail about this procedure.

Data sources for UNIX and Linux are stored in the system information file (by default, `odbc.ini`). If you have a Motif GUI environment on Linux, you can configure and modify data sources through the DataDirect ODBC Data Source Administrator for Linux (the Linux ODBC Administrator) using a driver Setup dialog box. (See [“Configuration Through the Administrator” on page 100](#) for a detailed explanation of the Administrator.)

If you do not have a GUI environment, you can configure and modify data sources directly by editing the `odbc.ini` file and storing default connection values there. See [“Configuration Through the System Information \(`odbc.ini`\) File” on page 103](#) for detailed information about the specific steps necessary to configure a data source.

Each of [Table 14-1 on page 523](#) and [Table 14-2 on page 524](#) lists driver connection string attributes that must be used in the `odbc.ini` file to set the value of the attributes. Note that only the long name of the attribute can be used in the file. The default listed in the table is the initial default value when the driver is installed.

### Data Source Configuration through a GUI

On Windows, data sources are stored in the Windows Registry. You can configure and modify data sources through the ODBC Administrator using a driver Setup dialog box, as described in this section.

On UNIX and Linux, data sources are stored in the `odbc.ini` file. On Linux, you can configure and modify data sources through the Linux ODBC Administrator using a driver Setup dialog box, as described in this section.

**NOTE:** This book shows dialog box images that are specific to Windows. If you are using the drivers in the Linux environment, the dialog box that you see may differ slightly from the Windows version. Windows-only and UNIX-only connection options are specifically noted by icons in the Setup dialog box descriptions.

When the driver is first installed, the values of its connection options are set by default. These values appear on the driver Setup dialog box tabs when you create a new data source. You can change these default values by modifying the data source. In the following procedure, the description of each tab is followed by a table that lists the connection options for that tab and their initial default values. This table links you to a complete description of the options and their connection string attribute equivalents. The connection string attributes are used to override the default values of the data source if you want to change these values at connection time.

### To configure a Microsoft SQL Server data source:

#### 1 Start the ODBC Administrator:

- On Windows, start the ODBC Administrator by selecting its icon from the DataDirect Connect program group.
- On Linux, change to the *install\_dir/tools* directory and, at a command prompt, enter:

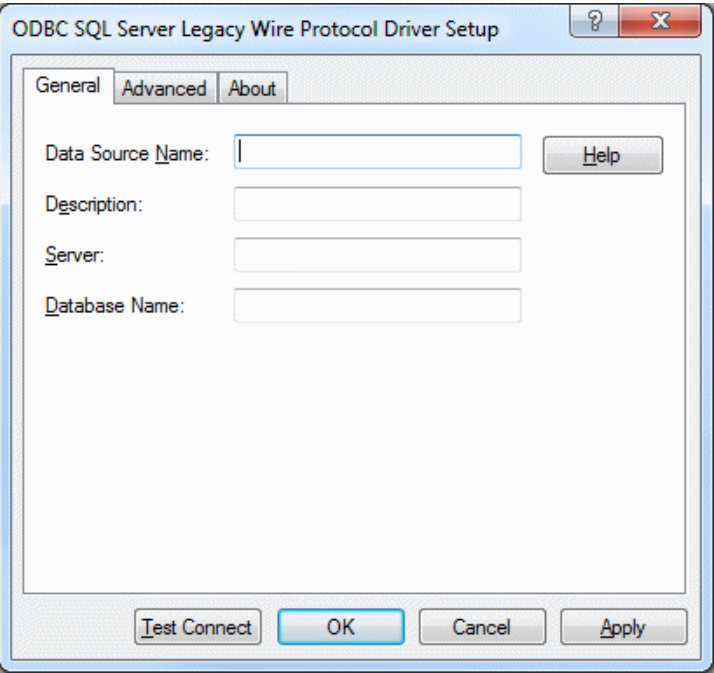
```
odbcadmin
```

where *install\_dir* is the path to the product installation directory.

#### 2 Select a tab:

- **User DSN:** If you are configuring an existing user data source, select the data source name and click **Configure** to display the driver Setup dialog box.  
  
If you are configuring a new user data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.
- **System DSN:** If you are configuring an existing system data source, select the data source name and click **Configure** to display the driver Setup dialog box.  
  
If you are configuring a new system data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.
- **File DSN:** If you are configuring an existing file data source, select the data source file and click **Configure** to display the driver Setup dialog box.  
  
If you are configuring a new file data source, click **Add** to display a list of installed drivers; then, select a driver. Click **Advanced** if you want to specify attributes; otherwise, click **Next** to proceed. Specify a name for the data source and click **Next**. Verify the data source information; then, click **Finish** to display the driver Setup dialog box.

The General tab of the Setup dialog box appears by default.

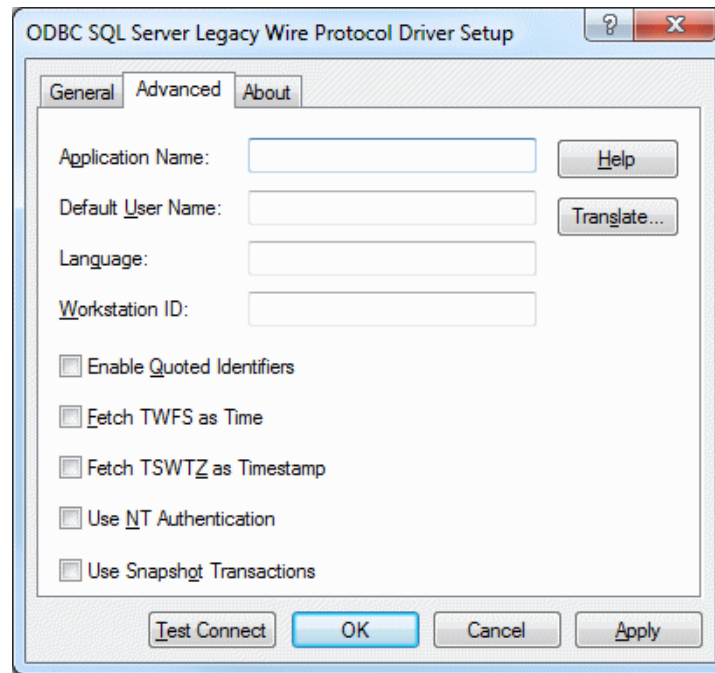


NOTE: The General tab displays only fields that are required for creating a data source. The fields on all other tabs are optional, unless noted otherwise.

- 3 On this tab, provide values for the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: General	Default
<a href="#">Data Source Name (see page 527)</a>	None
<a href="#">Description (see page 528)</a>	None
<a href="#">Server (see page 533)</a>	None
<a href="#">Database Name (see page 527)</a>	None

- 4 Optionally, click the **Advanced** tab to specify additional data source settings.



On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Advanced	Default
<a href="#">Application Name</a> (see page 525)	None
<a href="#">Default User Name</a> (see page 528)	None
<a href="#">Language</a> (see page 530)	None
<a href="#">Workstation ID</a> (see page 535)	None
<a href="#">Enable Quoted Identifiers</a> (see page 528)	Disabled
<a href="#">Fetch TWFS as Time</a> (see page 529)	Disabled
<a href="#">Fetch TSWTZ as Timestamp</a> (see page 528)	Disabled
<a href="#">Use NT Authentication</a> (see page 534)	Disabled
WINDOWS ONLY	
<a href="#">Use Snapshot Transactions</a> (see page 534)	Disabled
<a href="#">IANAAppCodePage</a> (see page 529)	4 (ISO 8559-1 Latin-1)
UNIX ONLY	

**Translate:** Click **Translate** to display the Select Translator dialog box, which lists the translators specified in the ODBC Translators section of the Registry. Progress 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.

- 5 Optionally, click the **Failover** tab to specify failover data source settings. This tab is available only on UNIX and Linux.

See ["Using Failover" on page 65](#) for a general description of failover and its related connection options.

On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Failover	Default
<a href="#">Load Balancing (see page 530)</a>	Disabled
<a href="#">Alternate Servers (see page 524)</a>	None
<a href="#">Connection Retry Count (see page 526)</a>	0
<a href="#">Connection Retry Delay (see page 527)</a>	3

- 6 At any point during the configuration process, you can click **Test Connect** to attempt to connect to the data source using the connection options specified in the driver Setup dialog box. A Login dialog box appears; see ["Using a Connection String" on page 520](#) for details. Note that the information you enter in the Login 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 incorrect environment or connection value, it displays 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.

## Using a Connection String

If you want to use a connection string for connecting to a database, or if your application requires it, you must specify either a DSN (data source name), a File DSN, or a DSN-less connection in the string. The difference is whether you use the `DSN=`, `FILEDSN=`, or the `DRIVER=` keyword in the connection string, as described in the ODBC specification. A DSN or FILEDSN connection string tells the driver where to find the default connection information. Optionally, you may specify *attribute=value* pairs in the connection string to override the default values stored in the data source.

The DSN connection string has the form:

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

The FILEDSN connection string has the form:

```
FILEDSN=filename.dsn[;attribute=value[;attribute=value]...]
```

The DSN-less connection string specifies a driver instead of a data source. All connection information must be entered in the connection string because the information is not stored in a data source.



The DSN-less connection string has the form:

```
DRIVER={[driver_name]}[;attribute=value[;attribute=value]...]
```

[Table 14-1](#) and [Table 14-2](#) list the long and short names for each attribute, as well as the initial default value when the driver is first installed. You can specify either long or short names in the connection string.

An example of a DSN connection string with overriding attribute values for Microsoft SQL Server is:

```
DSN=ACCOUNTING;DATABASE=ACCT
```

A FILEDSN connection string is similar except for the initial keyword:

```
FILEDSN=SQLServer.dsn;DATABASE=ACCT
```

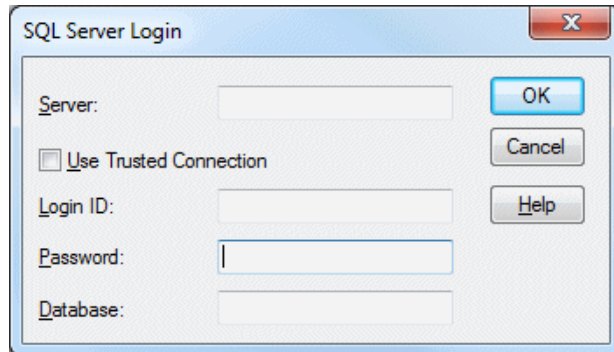
A DSN-less connection string must provide all necessary connection information:

```
DRIVER=DataDirect 7.1 SQL Server Legacy Wire Protocol;  
DATABASE=ACCT;SERVER=SQL2;UID=JOHN;PWD=XYZZY
```

The connection string attribute names are case-sensitive.

## Using a Login Dialog Box

Some ODBC applications display a Login dialog box when you are connecting to a data source. In these cases, the data source name has already been specified.



In the Login dialog box, provide the following information:

- 1 Type an IP address in Server in the following format: *IP\_address, port\_number*. For example, you can enter 199.226.224.34,5000. If your network supports named servers, you can specify an address as: *server\_name, port\_number*. For example, you can enter SSserver,5000.

The IP address can be specified in IPv4 on Windows, and in either IPv4 or IPv6 format, or a combination of the two, on UNIX. See ["Using IP Addresses" on page 54](#) for details about these formats.

To specify a named instance of Microsoft SQL Server, use the format: *server\_name\instance\_name*. If only a server name is specified with no instance name, the driver uses the default named instance on the server.

Type the name of a server on your network. It must be an entry on the Alias tab of the SQL Server Network Client Utility or the network name of a server running Microsoft SQL Server.

You can enter `(local)` when the driver is on the same computer as the Microsoft SQL Server database. You can connect to a local copy of Microsoft SQL Server, even when it is a non-networked version. Microsoft SQL Server 2000 and higher support multiple instances of Microsoft SQL Server running on the same computer.

- 2 Select the **Use Trusted Connection** check box to specify that the SQL Server Legacy Wire Protocol driver request a secure (or trusted) connection to Microsoft SQL Server. SQL Server uses integrated login security to establish connections using this data source, regardless of the current login security mode at the server. Any login ID or password supplied is ignored. The Microsoft SQL Server system administrator must have associated your Windows network ID with a Microsoft SQL Server login ID.

Clear this box to specify that Microsoft SQL Server use standard login security to establish connections using this data source. You must specify a login ID and password for all connection requests.

- 3 Type the Microsoft SQL Server login ID to use for the connection if Use Trusted Connection is not selected. If Use Trusted Connection is selected, the Login ID field is disabled.
- 4 Type the password to use for the connection if Use Trusted Connection is not selected. If Use Trusted Connection is selected, the Password field is disabled.
- 5 Type the name of the database to which you want to connect. If you do not specify a value, the default database defined by Microsoft SQL Server is used.
- 6 Click **OK** to log on to the Microsoft SQL Server database installed on the server you specified and to update the values in the Registry.

---

## Connection Option Descriptions

The following connection option descriptions are listed alphabetically by the GUI name that appears on the driver Setup dialog. The connection string attribute name is listed immediately underneath the GUI name. For example:

### Application Using Threads

Attribute    ApplicationUsingThreads (AUT)

In most cases, the GUI name and the attribute name are the same; however, some exceptions exist. If you need to look up an option by its connection string attribute name, please refer to the alphabetical table of connection string attribute names.

Also, a few connection string attributes, for example, Password, do not have equivalent options that appear on the GUI. They are in the list of descriptions alphabetically by their attribute names.

NOTE: SQL Server driver connection string attributes do not use short name equivalents.

Table 14-1 lists the connection string attributes supported by the SQL Server Legacy Wire Protocol driver on Windows.

## Windows

**Table 14-1. SQL Server Legacy Wire Protocol Attribute Names on Windows**

Attribute	Default
AnsiNPW	yes (Enabled)
APP	None
AttachDBFileName	None
AutoTranslate	yes (Enabled)
DATABASE	None
DataSourceName (DSN)	None
Description (n/a)	None
FetchTSWTZasTimestamp (FTSWTZAT)	0 (Disabled)
FetchTWFSasTime (FTWFSAT)	0 (Disabled)
LANGUAGE	None
Network	None
PWD (use Password for odbc.ini file)	None
QueryLog_On	no (Disabled)
QueryLogFile	None
QueryLogTime	None
QuotedID	no (Disabled)
Regional	yes (Enabled)
SAVEFILE	None
SERVER	None
SnapshotSerializable	0 (Disabled)
StatsLog_On	no (Disabled)
StatsLogFile	None
Trusted_Connection	no (Disabled)
UID (use LogonID for odbc.ini file)	None
WSID	None

## UNIX and Linux

Table 14-2 lists the connection string attributes supported by the SQL Server Legacy Wire Protocol driver on UNIX/Linux.

NOTE: SQL Server driver connection string attributes do not use short name equivalents.

Table 14-2. SQL Server Legacy Wire Protocol Attribute Names on UNIX/Linux

Attribute	Default
Address	None
AlternateServers	None
AnsiNPW	yes (Enabled)
APP	None
ConnectionRetryCount	0
ConnectionRetryDelay	3
DataSourceName (DSN)	None
DATABASE	None
Description (n/a)	None
FetchTSWTZasTimestamp (FTSWTZAT)	0 (Disabled)
FetchTWFSasTime (FTWFSAT)	0 (Disabled)
IANAAppCodePage	4 (ISO 8559-1 Latin-1)
LANGUAGE	None
LoadBalancing (LB)	0 (Disabled)
PWD (use Password for odbc.ini file)	None
QuotedID	no (Disabled)
SnapshotSerializable	0 (Disabled)
UID (use LogonID for odbc.ini file)	None
WSID	None

Alternate Servers

Attribute	AlternateServers
Description	A list of alternate database servers to which the driver tries to connect if the primary database server is unavailable. Specifying a value for this option enables connection failover for the driver. The value you specify must be in the form of a string that defines the physical location of each alternate server. All of the other required connection information for each alternate server is the same as what is defined for the primary server connection.
Valid Values	<p>(Address=addressvalue[, . . .])</p> <p>You must specify the network address of each alternate server.</p> <p>NOTE: An alternate server address in IPv6 format must be enclosed in double quotation marks.</p>
Example	<p>The following two Alternate Servers values define two alternate database servers for connection failover:</p> <pre>AlternateServers=(Address=MySQLServer\Instance1, Address="255.125.1.11, 5002")</pre>

In this example, the network address of the last alternate contains commas. In this case, enclose the network address with double quotation marks as shown.

Default None  
 GUI Tab [Failover tab](#) on page 519

## AnsiNPW

Attribute AnsiNPW  
 Description Determines whether ANSI-defined behaviors are exposed.  
 Valid Values yes | no

When set to yes (Enabled), the driver uses ANSI-defined behaviors for handling NULL comparisons, character data padding, warnings, and NULL concatenation. If the driver appears to be truncating trailing blank spaces, this attribute should be set to no.

When set to no (Disabled), ANSI-defined behaviors are not exposed.

Default yes (Enabled)  
 GUI Tab n/a

## Application Name

Attribute APP  
 Description The name Microsoft SQL Server uses to identify your application.  
 Valid Values *string*

where *string* is your application name.

Default None  
 GUI Tab [Advanced tab](#) on page 519

## AttachDBFileName

Attribute AttachDBFileName  
 Description The name of the primary file of an attachable database.  
 Valid Values *string*

where *string* is name of the primary file of an attachable database.

Include the full path and escape any slash ( \ ) characters if using a C character string variable:

```
AttachDBFileName=C:\\MyFolder\\MyDB.mdf
```

This database is attached and becomes the default database for the connection. To use AttachDBFileName, you must also specify the database name in either the SQLDriverConnect DATABASE parameter or the SQL\_COPT\_CURRENT\_CATALOG connection attribute. If the database was previously attached, Microsoft SQL Server will not reattach it; it will use the attached database as the default for the connection.

Default	None
GUI Tab	n/a

## AutoTranslate

Attribute	AutoTranslate
Description	Determines how ANSI character strings are translated.
Valid Values	yes   no

If set to yes (Enabled), ANSI character strings sent between the client and server are translated by converting through Unicode to minimize problems in matching extended characters between the code pages on the client and the server.

These conversions are performed on the client by the SQL Server Legacy Wire Protocol driver. This requires that the same ANSI code page (ACP) used on the server be available on the client.

These settings have no effect on the conversions that occur for the following transfers:

- Unicode SQL\_C\_WCHAR client data sent to char, varchar, or text on the server.
- Char, varchar, or text server data sent to a Unicode SQL\_C\_WCHAR variable on the client.
- ANSI SQL\_C\_CHAR client data sent to Unicode nchar, nvarchar, or ntext on the server.
- Unicode char, varchar, or text server data sent to an ANSI SQL\_C\_CHAR variable on the client.

If set to no (Disabled), character translation is not performed.

The SQL Server Legacy Wire Protocol driver does not translate client ANSI character SQL\_C\_CHAR data sent to char, varchar, or text variables, parameters, or columns on the server. No translation is performed on char, varchar, or text data sent from the server to SQL\_C\_CHAR variables on the client. If the client and Microsoft SQL Server are using different ACPs, then extended characters can be misinterpreted.

Default	yes (Enabled)
GUI Tab	n/a

## Connection Retry Count

Attribute	ConnectionRetryCount
Description	The number of times the driver retries connection attempts to the primary database server, and if specified, alternate servers until a successful connection is established.
	The Connection Retry Delay option specifies the wait interval, in seconds, to occur between retry attempts.
Valid Values	0   <i>x</i>

where *x* is a positive integer from 1 to 65535.

If set to 0, the driver does not try to connect after the initial unsuccessful attempt.

If set to  $x$ , the driver retries connection attempts the specified number of times. If a connection is not established during the retry attempts, the driver returns an error that is generated by the last server to which it tried to connect.

Default 0

GUI Tab [Failover tab](#) on page 519

## Connection Retry Delay

Attribute ConnectionRetryDelay

Description The number of seconds the driver waits between connection retry attempts when Connection Retry Count is set to a positive integer.

Valid Values 0 |  $x$

where  $x$  is a positive integer from 1 to 65535.

If set to 0, there is no delay between retries.

If set to  $x$ , the driver waits between connection retry attempts the specified number of seconds.

Default 3

GUI Tab [Failover tab](#) on page 519

## Data Source Name

Attribute DataSourceName (DSN)

Description The name of a data source in your Windows Registry or odbc.ini file.

Valid Values *string*

where *string* is the name of a data source.

Default None

GUI Tab [General tab](#) on page 518

## Database Name

Attribute DATABASE

Description The name of the database to which you want to connect.

Valid Values *database\_name*

where *database\_name* is the name of a valid database.

Default None

GUI Tab [General tab](#) on page 518

## Default User Name

Attribute	UID (use LogonID for odbc.ini file)
Description	The default user ID used to connect to your database. Your ODBC application may override this value or you may override it in the logon dialog box or connection string.
Valid Values	<i>userid</i>  where <i>userid</i> is a valid user ID with permissions to access the database.
Default	None
GUI Tab	<a href="#">Advanced tab</a> on page 519

## Description

Attribute	Description (n/a)
Description	An optional long description of a data source. This description is not used as a runtime connection attribute, but does appear in the ODBC.INI section of the Registry and in the odbc.ini file.
Valid Values	<i>string</i>  where <i>string</i> is a description of a data source.
Default	None
GUI Tab	<a href="#">General tab</a> on page 518

## Enable Quoted Identifiers

Attribute	QuotedID
Description	Determines whether the driver allows the use of quoted identifiers.
Valid Values	yes   no  If set to yes (Enabled), Microsoft SQL Server enforces ANSI rules regarding quotation marks. Double quotation marks can only be used for identifiers, such as column and table names. Character strings must be enclosed in single quotation marks, for example:  <pre>SELECT "au_id" FROM "authors" WHERE "au_lname" = 'O'Brien'</pre> If set to no (Disabled), applications that use quoted identifiers encounter errors when they generate SQL statements with quoted identifiers.
Default	no (Disabled)
GUI Tab	<a href="#">Advanced tab</a> on page 519

## Fetch TSWTZ as Timestamp

Attribute	FetchTSWTZasTimestamp (FTSWTZAT)
Description	Determines whether the driver returns column values with the timestamp with time zone data type as the ODBC data type SQL_TYPE_TIMESTAMP or SQL_VARCHAR.



Valid Values 0 | 1

If set to 1 (Enabled), the driver returns column values with the timestamp with time zone data type as the ODBC type SQL\_TYPE\_TIMESTAMP. The time zone information in the fetched value is truncated. Use this value if your application needs to process values the same way as TIMESTAMP columns.

If set to 0 (Disabled), the driver returns column values with the timestamp with time zone data type as the ODBC data type SQL\_VARCHAR. Use this value if your application requires the time zone information in the fetched value.

Default 0 (Disabled)

GUI Tab [Advanced tab](#) on page 519

## Fetch TWFS as Time

Attribute FetchTWFSasTime (FTWFSAT)

Description Determines whether the driver returns column values with the time data type as the ODBC data type SQL\_TYPE\_TIME or SQL\_TYPE\_TIMESTAMP.

Supported only for Microsoft SQL Server 2008.

Valid Values 0 | 1

If set to 1 (Enabled), the driver returns column values with the time data type as the ODBC data type SQL\_TYPE\_TIME. The fractional seconds portion of the value is truncated.

If set to 0 (Disabled), the driver returns column values with the time data type as the ODBC data type SQL\_TYPE\_TIMESTAMP. The fractional seconds portion of the value is preserved. Time columns are not searchable when they are described and fetched as timestamp

NOTE: When returning time with fractional seconds data as SQL\_TYPE\_TIMESTAMP, the Year, Month and Day parts of the timestamp must be set to zero.

Default 0 (Disabled)

GUI Tab [Advanced tab](#) on page 519

## IANAAppCodePage

Attribute IANAAppCodePage

Description An Internet Assigned Numbers Authority (IANA) value. You must specify a value for this option if your application is not Unicode-enabled or if your database character set is not Unicode. Refer to [Chapter 4 "Internationalization, Localization, and Unicode"](#) in the *DataDirect Connect Series for ODBC Reference* for details.

The driver uses the specified IANA code page to convert "W" (wide) functions to ANSI.

The driver and Driver Manager both check for the value of IANAAppCodePage in the following order:

- In the connection string
- In the Data Source section of the system information file (odbc.ini)
- In the ODBC section of the system information file (odbc.ini)

If the driver does not find an IANAAppCodePage value, the driver uses the default value of 4 (ISO 8859-1 Latin-1).

Valid Values *IANA\_code\_page*

where *IANA\_code\_page* is one of the valid values listed in [Chapter 1 “Values for the Attribute IANAAppCodePage”](#) in the *DataDirect Connect Series for ODBC Reference*. The value must match the database character encoding and the system locale.

Default 4 (ISO 8559-1 Latin-1)

GUI Tab [Advanced tab](#) on page 519

## Language

Attribute LANGUAGE

Description The national language to use for Microsoft SQL Server system messages.

Valid Values *lang*

where *lang* is the language to use for Microsoft SQL Server system messages. This overrides the default language specified for the login on the server. If no language is specified, the connection uses the default language specified for the login on the server.

Default None

GUI Tab [Advanced tab](#) on page 519

## Load Balancing

Attribute LoadBalancing (LB)

Description Determines whether the driver uses client load balancing in its attempts to connect to the database servers (primary and alternate). You can specify one or multiple alternate servers by setting the Alternate Servers option.

Valid Values 0 | 1

If set to 1 (Enabled), the driver uses client load balancing and attempts to connect to the database servers (primary and alternate servers) in random order.

If set to 0 (Disabled), the driver does not use client load balancing and connects to each server based on their sequential order (primary server first, then, alternate servers in the order they are specified).

Default 0 (Disabled)

GUI Tab [Failover tab](#) on page 519

## Network

Attribute	Network
Description	The name of a network library dynamic-link library.
Valid Values	<i>string</i>  where <i>string</i> is the name of a network library dynamic-link library. The name need not include the path and must not include the .DLL file name extension, for example, Network=dbnmpntw.
Default	None
GUI Tab	n/a

## PWD

Attribute	PWD (use Password for odbc.ini file)
Description	The password that the application uses to connect to your database. The Password option cannot be specified through the driver Setup dialog box and should not be stored in a data source. It is specified through the Logon dialog box or a connection string.
Valid Values	<i>pwd</i>  where <i>pwd</i> is a valid password.  PWD need not be specified if the login has a NULL password.
Default	None
GUI Tab	n/a

## QueryLogFile

Attribute	QueryLogFile
Description	The full path and file name of a file to be used for logging data about long-running queries. The QueryLog_On option must be set to yes.
Valid Values	<i>string</i>  where <i>string</i> is the full path and file name of the file to be used for logging data.
Default	None
GUI Tab	n/a

## QueryLog\_On

Attribute	QueryLog_On
Description	Determines whether data about long-running queries data is logged.
Valid Values	yes   no  When set to yes (Enabled), logging data about long-running queries data is enabled on the connection.

When set to no (Disabled), long-running query data is not logged.

Default no (Disabled)  
GUI Tab n/a

## QueryLogTime

Attribute QueryLogTime  
Description A digit character string specifying the threshold for logging data about long-running queries.  
Valid Values *string*

where *string* is a digit character string specifying the threshold in milliseconds, for logging data.

Any query that does not receive a response in the time specified is written to the long-running query log file.

Default None  
GUI Tab n/a

## Regional

Attribute Regional  
Description Determines how currency, date, and time data are converted.  
Valid Values yes | no

When set to yes (Enabled), the SQL Server Legacy Wire Protocol driver uses client settings when converting currency, date, and time data to character data. The conversion is one way only; the driver does not recognize non-ODBC standard formats for date strings or currency values.

When set to no (Disabled), the driver uses ODBC standard strings to represent currency, date, and time data that is converted to string data.

Default yes (Enabled)  
GUI Tab n/a

## SAVEFILE

Attribute SAVEFILE  
Description The name of an ODBC data source file into which the attributes of the current connection are saved.  
Valid Values *string*

where *string* is the name of an ODBC data source file into which the attributes of the current connection are saved if the connection is successful.

Default None  
GUI Tab n/a

## Server

Attribute	SERVER
Attribute	Address
Description	The location of the server.
Valid Values	<i>IP_address</i>   <i>named_server</i>   <i>named_instance</i>   <i>server_name</i>

where

*IP\_address* is the IP address of the server to which you want to connect. Specify this address as: *IP\_address, port\_number*. For example, you can enter 199.226.224.34, 5000.

The IP address can be specified in either IPv4 or IPv6 format, or a combination of the two. See ["Using IP Addresses" on page 54](#) for details about these formats.

*named\_server* is the named server address of the server to which you want to connect. Specify this address as: *named\_server, port\_number*. For example, you can enter SSserver, 5000.

*named\_instance* is a named instance of Microsoft SQL Server. Specify this address as: *server\_name\instance\_name*. If only a server name is specified with no instance name, the driver uses the default named instance on the server.

*server\_name* is the name of a server on your network. It must be an entry on the Alias tab of the SQL Server Network Client Utility or the network name of a server running Microsoft SQL Server. You can enter (local) when the driver is on the same computer as the Microsoft SQL Server database. You can connect to a local copy of Microsoft SQL Server, even when it is a non-networked version. Microsoft SQL Server 2000 and higher support multiple instances of Microsoft SQL Server running on the same computer.

Default	None
GUI Tab	<a href="#">General tab</a> on page 518

## StatsLogFile

Attribute	StatsLogFile
Description	The full path and file name of a file to be used for recording SQL Server Legacy Wire Protocol driver performance data. The StatsLog_On option must be set to yes.
Valid Values	<i>string</i>

where *string* is the full path and file name of the file to be used for recording data.

Default	None
GUI Tab	n/a

## StatsLog\_On

Attribute	StatsLog_On
Description	Determines whether SQL Server Legacy Wire Protocol driver performance data is made available.

Valid Values	yes   no
	When set to yes (Enabled), SQL Server Legacy Wire Protocol driver performance data is captured.
	When set to no (Disabled), SQL Server Legacy Wire Protocol driver performance data is not available on the connection.
Default	no (Disabled)
GUI Tab	n/a

## Use NT Authentication

Attribute	Trusted_Connection
Description	Specifies that the SQL Server Legacy Wire Protocol driver request a secure (or trusted) connection to Microsoft SQL Server.
Valid Values	0   1
	When set to 1 (Enabled), Microsoft SQL Server uses integrated login security to establish connections using this data source, regardless of the current login security mode at the server. Any login ID or password supplied is ignored. The Microsoft SQL Server system administrator must have associated your Windows network ID with a Microsoft SQL Server login ID.
	When set to 0 (Disabled), Microsoft SQL Server uses standard login security to establish connections using this data source. In this case, you must specify a login ID and password for all connection requests.
Default	0 (Disabled)
GUI Tab	<a href="#">Advanced tab</a> on page 519

## Use Snapshot Transactions

Attribute	SnapshotSerializable
Description	Allows your application to use the snapshot isolation level if your Microsoft SQL Server database is configured for Snapshot isolation. Supported only for Microsoft SQL Server 2005 and higher.
	See <a href="#">"Isolation and Lock Levels Supported" on page 538</a> for details about using the snapshot isolation level.
	This connection option can affect performance. See <a href="#">"Performance Considerations" on page 535</a> for details.
Valid Values	0   1
	When set to 1 (Enabled) and your application has the transaction isolation level set to serializable, the application uses the snapshot isolation level.
	When set to 0 (Disabled) and your application has the transaction isolation level set to serializable, the application uses the serializable isolation level.

This option is useful for existing applications that set the isolation level to serializable. Using Snapshot Transactions in this case allows you to change to the snapshot isolation level with no or minimum code changes. If developing a new application, you can code it to set the connection attribute `SQL_COPT_SS_TXN_ISOLATION` to the value `SQL_TXN_SS_SNAPSHOT`.

Default 0 (Disabled)  
 GUI Tab [Advanced tab](#) on page 519

## Workstation ID

Attribute WSID  
 Description The workstation ID that is used by the client.  
 Valid Values *string*  
 where *string* is the workstation ID.  
 Default None  
 GUI Tab [Advanced tab](#) on page 519

---

## Performance Considerations

**Use Snapshot Transactions (SnapshotSerializable):** You must have your Microsoft SQL Server 2005 and higher database configured for snapshot isolation for this connection option to work. Snapshot Isolation provides transaction-level read consistency and an optimistic approach to data modifications by not acquiring locks on data until data is to be modified. This Microsoft SQL Server 2005 and higher feature can be useful if you want to consistently return the same result set even if another transaction has changed the data and 1) your application executes many read operations or 2) your application has long running transactions that could potentially block users from reading data. This feature has the potential to eliminate data contention between read operations and update operations. When this connection option is enabled, performance is improved due to increased concurrency.

See ["Using The Snapshot Isolation Level" on page 538](#) for details.

## Data Types

Table 14-3 shows how the Microsoft SQL Server data types are mapped to the standard ODBC data types. "Unicode Support" on page 537 lists Microsoft SQL Server to Unicode data type mappings.

**Table 14-3. Microsoft SQL Server Data Types**

SQL Server	ODBC
binary	SQL_BINARY
bigint <sup>1</sup>	SQL_BIGINT
bit	SQL_BIT
char	SQL_CHAR
date <sup>2</sup>	SQL_TYPE_DATE
datetime	SQL_TYPE_TIMESTAMP
datetime2 <sup>2</sup>	SQL_TYPE_TIMESTAMP
datetimeoffset <sup>2 3</sup>	SQL_WVARCHAR
decimal	SQL_DECIMAL
decimal() identity	SQL_DECIMAL
float	SQL_FLOAT
image	SQL_LONGVARBINARY
int	SQL_INTEGER
int identity	SQL_INTEGER
money	SQL_DECIMAL
numeric	SQL_NUMERIC
numeric() identity	SQL_NUMERIC
real	SQL_REAL
smalldatetime	SQL_TYPE_TIMESTAMP
smallint	SQL_SMALLINT
smallint identity	SQL_SMALLINT
smallmoney	SQL_DECIMAL
text	SQL_LONGVARCHAR
time <sup>2 4</sup>	SQL_TYPE_TIMESTAMP
timestamp	SQL_VARBINARY
tinyint	SQL_TINYINT
tinyint identity	SQL_TINYINT
uniqueidentifier	SQL_GUID
varbinary	SQL_VARBINARY
varbinary(max) <sup>5</sup>	SQL_LONGVARBINARY



**Table 14-3. Microsoft SQL Server Data Types** (cont.)

SQL Server	ODBC
varchar	SQL_VARCHAR
varchar(max) <sup>5</sup>	SQL_LONGVARCHAR

1. Bigint supported on Windows driver only.
2. Supported only on Microsoft SQL Server 2008 and higher.
3. Datetimeoffset mapping changes based on the setting of the Fetch TSWTZ as Timestamp option.
4. Time mapping changes based on the setting of the Fetch TWFS as Time option.
5. Supported only on Microsoft SQL Server 2005 and higher.

See ["Retrieving Data Type Information" on page 59](#) for information about retrieving data types.

## Unicode Support

The SQL Server Legacy Wire Protocol driver maps the Microsoft SQL Server data types to Unicode data types as shown in the following table:

SQL Server Data Type	Mapped to. . .
nchar	SQL_WCHAR
ntext	SQL_WLONGVARCHAR
nvarchar	SQL_WVARCHAR
nvarchar(max) <sup>1</sup>	SQL_WLONGVARCHAR
sysname	SQL_WVARCHAR
xml <sup>1</sup>	SQL_WLONGVARCHAR

1. Supported only for Microsoft SQL Server 2005 and higher.

The driver supports the Unicode ODBC W (Wide) function calls, such as SQLConnectW. This allows the Driver Manager to transmit these calls directly to the driver. Otherwise, the Driver Manager would incur the additional overhead of converting the W calls to ANSI function calls, and vice versa.

See ["UTF-16 Applications on UNIX and Linux" on page 125](#) for related details. Also, refer to [Chapter 4 "Internationalization, Localization, and Unicode"](#) in the *DataDirect Connect Series for ODBC Reference* for a more detailed explanation of Unicode.

---

## Advanced Features

The driver supports failover and its related connection options. Failover connection options are located on the [Failover tab](#) of the driver Setup dialog box. See ["Using Failover" on page 65](#) for a general description of failover and its implementation.

Only the UNIX/Linux driver supports this feature.

---

## Isolation and Lock Levels Supported

Microsoft SQL Server supports isolation levels 0 (Read Uncommitted), 1 (Read Committed), 2 (Repeatable Read), and 3 (Serializable). Microsoft SQL Server supports row-level and table-level locking. Refer to [Chapter 7 "Locking and Isolation Levels"](#) in the *DataDirect Connect Series for ODBC Reference* for details.

Microsoft SQL Server 2005 and higher supports the following additional isolation levels:

- Snapshot
- Read Committed with Snapshots
- Read Committed with Locks (equivalent to Read Committed in previous Microsoft SQL Server versions)

### Using The Snapshot Isolation Level

The Snapshot isolation level is available only with Microsoft SQL Server 2005 and higher. Setting the SnapshotSerializable connection string attribute changes the behavior of the Serializable isolation level to use the Snapshot Isolation level. This allows an application to use the Snapshot Isolation level with minimal or no code changes.

If you are writing a new application, you may want to code it to set the connection attribute SQL\_COPT\_SS\_TXN\_ISOLATION to the value SQL\_TXN\_SS\_SNAPSHOT. The application then uses the snapshot isolation level without requiring the Use Snapshot Transactions connection option.

See ["Use Snapshot Transactions" on page 534](#) for additional information.

---

## SQL Support

The driver supports the core SQL grammar.

---

## ODBC Conformance Level

The driver supports ODBC conformance level 1.

In addition, the following function is supported: SQLDescribeParam.

Refer to [Chapter 2 “ODBC API and Scalar Functions”](#) in the *DataDirect Connect Series for ODBC Reference* for a list of the API functions supported by the SQL Server Legacy Wire Protocol driver.

---

## Number of Connections and Statements Supported

The SQL Server Legacy Wire Protocol driver supports multiple connections and a single statement per connection.

---

## Using Arrays of Parameters

Microsoft SQL Server databases natively support parameter arrays, and the SQL Server Legacy Wire Protocol driver, in turn, supports them. When designing an application for performance, using native parameter arrays for bulk inserts or updates, for example, can improve performance. Refer to [Chapter 5 “Designing ODBC Applications for Performance Optimization”](#) in the *DataDirect Connect Series for ODBC Reference* for more information about using arrays of parameters to improve performance.



## Part 3: The 32-Bit Drivers

This part describes the drivers that are available only in 32-bit versions. See ["Part 2: The 32-Bit/64-Bit Drivers" on page 127](#) for the drivers that are available in both 32- and 64-bit versions.

This part contains the following chapters:

- [Chapter 15 "The Btrieve \(Pervasive.SQL\) Driver" on page 543](#)
- [Chapter 16 "The dBASE Driver" on page 561](#)
- [Chapter 17 "The Informix Driver" on page 587](#)
- [Chapter 18 "The Text Driver" on page 607](#)
- [Chapter 19 "The XML Driver" on page 629](#)



# 15 The Btrieve (Pervasive.SQL) Driver

The DataDirect Connect *for* ODBC Btrieve driver (the Btrieve driver) supports the following versions of Btrieve files:

- Pervasive.SQL 8.5
- Pervasive.SQL 2000
- Pervasive.SQL 7.0
- Btrieve version 6.15

The driver executes SQL statements directly on Btrieve files.

The Btrieve driver is 32-bit only and is supported in the Windows environments. See ["Environment-Specific Information" on page 45](#) for detailed information about the environments supported by this driver.

Refer to the readme file shipped with your DataDirect Connect product for the file name of the Btrieve driver.

---

## Driver Requirements

To access a Btrieve database, you must be using the appropriate client software for the version of the Btrieve database to which you are connecting:

Database Versions	Client Names
Pervasive.SQL 8.5	Pervasive.SQL 8.5 client software
Pervasive.SQL 2000	Pervasive.SQL 2000 client software
Pervasive.SQL 7.0	Pervasive.SQL 7.0 client software
Btrieve 6.15 for Windows 9x	Btrieve Developer's Kit or Btrieve WorkStation Client Engine
Btrieve 6.15 for Windows NT	Btrieve Developer's Kit, Btrieve WorkStation Client Engine, or Btrieve Client/Server Database Engine

NOTE: The Btrieve driver may experience problems if the Btrieve Microkernel Engine's communication buffer size is smaller than that of the Btrieve driver's Array Size option. You can increase the communication buffer size with the Pervasive Software Setup Utility, or you can decrease the value of Array Size option through the ODBC Btrieve Driver setup dialog box or through the ArraySize connection string attribute.

Before you attempt to access Btrieve files, you must incorporate existing Btrieve files into a Scalable SQL database. See ["Managing Databases" on page 544](#) for information about Scalable SQL databases.

---

## Managing Databases

If you already use Scalable SQL, the Btrieve driver can access your Scalable SQL databases directly. If not, your Btrieve files must be incorporated into a Scalable SQL database.

A Scalable SQL database is composed of data files that contain your records and data dictionary files that describe the database. The data files are Btrieve files. The data dictionary files are special Btrieve files that contain descriptions of the data files, views, fields, and indexes in your database.

All Btrieve files in a Scalable SQL database must reside in the same directory. In addition to the Btrieve data files, the three data dictionary files (FILE.DDF, FIELD.DDF, and INDEX.DDF) also must be in the directory.

Incorporating a Btrieve file into a Scalable SQL database does not change the Btrieve file in any way. You can continue to access the file directly with any existing Btrieve application.

---

## Transactions

The Btrieve driver supports *transactions*. A transaction is a series of database changes that is treated as a single unit. In applications that do not use transactions, the Btrieve driver immediately executes Insert, Update, and Delete statements on the database files and the changes are automatically committed when the SQL statement is executed. You cannot undo these changes. In applications that use transactions, the Btrieve driver holds inserts, updates, and deletes until you issue a Commit or Rollback. A Commit saves the changes to the database file; a Rollback undoes the changes.

Transactions affect the removal of record locking. All locks are removed when SQLTransact is called with the Commit or Rollback option to end the active transaction.

To use the Btrieve driver's transaction processing capabilities, consult the Pervasive documentation.

---

## Configuring and Connecting to Data Sources

After you install the driver, you configure data sources to connect to the database. See [Chapter 1 "Quick Start Connect" on page 35](#) for an explanation of different types of data sources. The data source contains connection options that allow you to tune the driver for specific performance. If you want to use a data source but need to change some of its values, you can either modify the data source or override its values at connection time through a connection string.

If you choose to use a connection string, you must use specific connection string attributes. See ["Using a Connection String" on page 548](#) and [Table 15-1 on page 549](#) for an alphabetical list of driver connection string attributes and their initial default values.



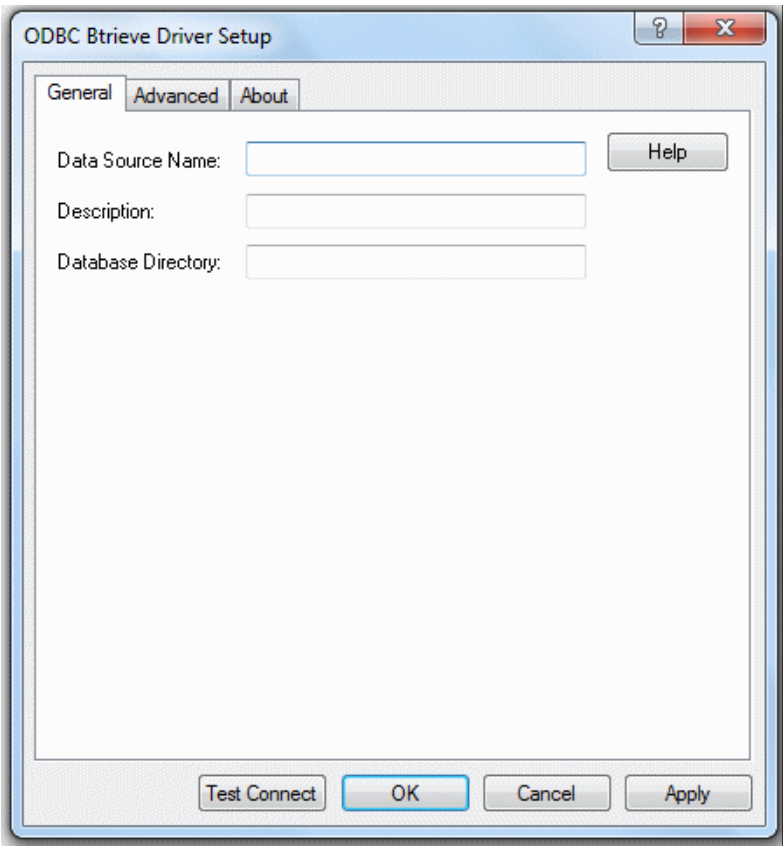
On Windows, data sources are stored in the Windows Registry. You can configure and modify data sources through the ODBC Administrator using a driver Setup dialog box.

When the driver is first installed, the values of its connection options are set by default. These values appear on the driver Setup dialog box tabs when you create a new data source. You can change these default values by modifying the data source. In the following procedure, the description of each tab is followed by a table that lists the connection options for that tab and their initial default values. This table links you to a complete description of the options and their connection string attribute equivalents. The connection string attributes are used to override the default values of the data source if you want to change these values at connection time.

**To configure a Btrieve data source:**

- 1 Start the ODBC Administrator by selecting its icon from the DataDirect Connect program group; then, select a tab:
  - **User DSN:** If you are configuring an existing user data source, select the data source name on the User DSN tab and click **Configure** to display the driver Setup dialog box.  
  
If you are configuring a new user data source, click **Add** on the User DSN tab to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.
  - **System DSN:** To configure a new system data source, click **Add** on the System DSN tab to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.
  - **File DSN:** If you are configuring an existing file data source, select the data source name on the File DSN tab and click **Configure** to display the driver Setup dialog box.  
  
If you are configuring a new file data source, click **Add** on the File DSN tab to display a list of installed drivers. Select the driver and click **Next**. Specify a name for the data source and click **Next**. Verify the data source information; then, click **Finish** to display the driver Setup dialog box.

The General tab of the Setup dialog box appears by default.

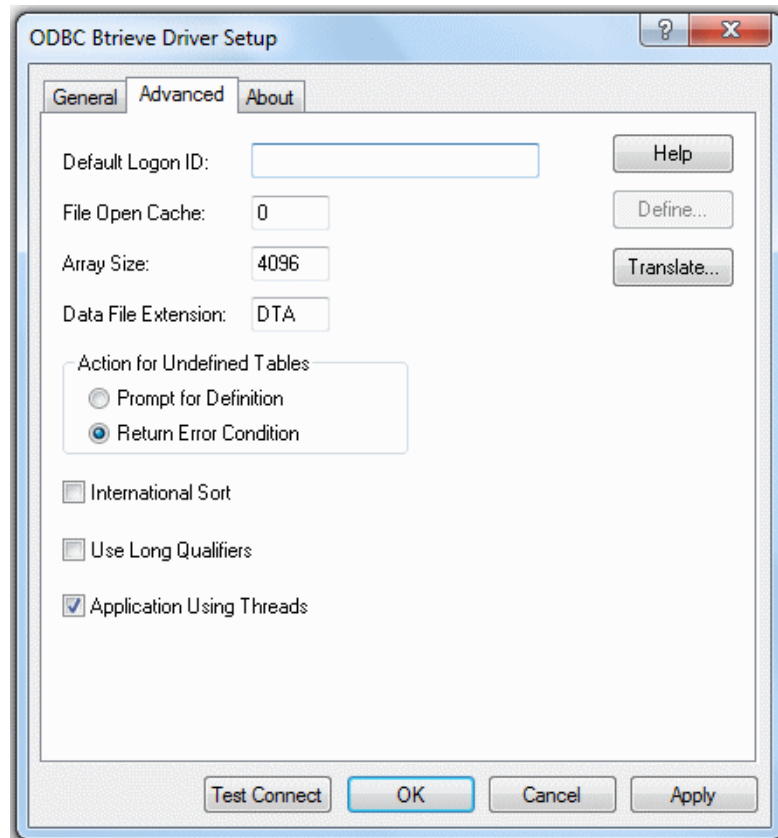


NOTE: The General tab displays only fields that are required for creating a data source. The fields on all other tabs are optional, unless noted otherwise.

- 2 On this tab, provide values for the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: General	Default
<a href="#">Data Source Name (see page 551)</a>	None
<a href="#">Description (see page 552)</a>	None
<a href="#">Database Directory (see page 551)</a>	None

- 3 Optionally, click the **Advanced** tab to specify data source settings.



On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Advanced	Default
<a href="#">Default Logon ID (see page 551)</a>	None
<a href="#">File Open Cache (see page 552)</a>	0
<a href="#">Array Size (see page 550)</a>	4096
<a href="#">Data File Extension (see page 550)</a>	DTA
<a href="#">Action for Undefined Tables (see page 550)</a>	Return Error Condition
<a href="#">International Sort (see page 552)</a>	Disabled
<a href="#">Use Long Qualifiers (see page 553)</a>	Disabled
<a href="#">Application Using Threads (see page 550)</a>	Enabled

**Define:** Click **Define** to define table structure as described in ["Defining Table Structure" on page 553](#).

**Translate:** Click **Translate** to display the Select Translator dialog box, which lists the translators specified in the ODBC Translators section of the Registry. 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.

- 4 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.
  - 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 incorrect environment or connection value, it displays an appropriate error message.

Verify that all required client software is properly installed. If it is not, you will see the message:

`Specified driver could not be loaded due to system error [xxx].`

Click **OK**.
- 5 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.

## Using a Connection String

If you want to use a connection string for connecting to a database, or if your application requires it, you must specify either a DSN (data source name), a File DSN, or a DSN-less connection in the string. The difference is whether you use the `DSN=`, `FILEDSN=`, or the `DRIVER=` keyword in the connection string, as described in the ODBC specification. A DSN or FILEDSN connection string tells the driver where to find the default connection information. Optionally, you may specify *attribute=value* pairs in the connection string to override the default values stored in the data source.

The DSN connection string has the form:

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

The FILEDSN connection string has the form:

```
FILEDSN=filename.dsn[;attribute=value[;attribute=value]...]
```

The DSN-less connection string specifies a driver instead of a data source. All connection information must be entered in the connection string because the information is not stored in a data source.

The DSN-less connection string has the form:

```
DRIVER=[{ }driver_name[ } ][;attribute=value[;attribute=value]...]
```

[Table 15-1](#) lists the long and short names for each attribute, as well as the initial default value when the driver is first installed. You can specify either long or short names in the connection string.

An example of a DSN connection string with overriding attribute values for Btrieve is:

```
DSN=BTRIEVE FILES;DB=J:\Btrvdata
```

A FILEDSN connection string is similar except for the initial keyword:

```
FILEDSN=Btrieve.dsn;DB=J:\Btrvdata
```

A DSN-less connection string must provide all necessary connection information:

```
DRIVER=DataDirect 7.1 Btrieve;DB=J:\Btrvdata;UID=JOHN;PWD=XYZZY
```

## Connection Option Descriptions

The following connection option descriptions are listed alphabetically by the GUI name that appears on the driver Setup dialog box. The connection string attribute name, along with its short name, is listed immediately underneath the GUI name. For example:

### Application Using Threads

Attribute    ApplicationUsingThreads (AUT)

In most cases, the GUI name and the attribute name are the same; however, some exceptions exist. If you need to look up an option by its connection string attribute name, please refer to the alphabetical table of connection string attribute names.

Also, a few connection string attributes, for example, Password, do not have equivalent options that appear on the GUI. They are in the list of descriptions alphabetically by their attribute names.

[Table 15-1](#) lists the connection string attributes supported by the Btrieve driver.

**Table 15-1. Btrieve Attribute Names**

Attribute (Short Name)	Default
<a href="#">ApplicationUsingThreads (AUT)</a>	1 (Enabled)
<a href="#">ArraySize (AS)</a>	4096
<a href="#">Database (DB)</a>	None
<a href="#">DataFileExtension (DFE)</a>	DTA
<a href="#">DataSourceName (DSN)</a>	None
<a href="#">Description (n/a)</a>	None
<a href="#">FileOpenCache (FOC)</a>	0 (No File Open Caching)
<a href="#">IntlSort (IS)</a>	0 (Disabled)
<a href="#">LogonID (UID)</a>	None
<a href="#">Password (PWD)</a>	None
<a href="#">UndefinedTable (UT)</a>	Error
<a href="#">UseLongQualifiers (ULQ)</a>	0 (Disabled)

## Action for Undefined Tables

Attribute	UndefinedTable (UT)
Description	Determines whether the driver prompts the user when it encounters a table for which it has no structure information.
Valid Values	PROMPT   ERROR  Specify PROMPT to prompt the user.  Specify ERROR to return an error.
Default	ERROR (driver returns an error)
GUI Tab	<a href="#">Advanced tab</a> on page 547

## Application Using Threads

Attribute	ApplicationUsingThreads (AUT)
Description	Determines whether the driver works with applications using multiple ODBC threads.
Valid Values	0   1  If set to 1 (Enabled), the driver works with single-threaded and multi-threaded applications.  If set to 0 (Disabled), the driver does not work with multi-threaded applications. If using the driver with single-threaded applications, this value avoids additional processing required for ODBC thread-safety standards.
Default	1 (Enabled)
GUI Tab	<a href="#">Advanced tab</a> on page 547

## Array Size

Attribute	ArraySize (AS)
Description	The number of bytes in the array. This connection option enables the driver to retrieve an array of records from the Btrieve database and, in most cases, results in improved performance for the application.
Valid Values	A positive integer from 1 to 65535
Default	4096
GUI Tab	<a href="#">Advanced tab</a> on page 547

## Data File Extension

Attribute	DataFileExtension (DFE)
Description	A one- to three-character file name extension to use for data files.
Valid Values	<i>ext</i>  where <i>ext</i> is the name of the one- to three-character file name extension.

This value is used for all Create Table statements. Sending a Create Table using an extension other than the value specified for this option causes an error.

In other SQL statements, such as Select or Insert, users can specify an extension other than the one specified for this connection option. The Data File Extension value is used when no extension is specified.

Default DTA  
 GUI Tab [Advanced tab](#) on page 547

## Data Source Name

Attribute DataSourceName (DSN)  
 Description The name of a data source in your Windows Registry or odbc.ini file.  
 Valid Values *string*  
 where *string* is the name of a data source.  
 Default None  
 GUI Tab [General tab](#) on page 546

## Database Directory

Attribute Database (DB)  
 Description The directory that contains the data files.  
 Valid Values *database\_directory*  
 where *database\_directory* is the full path name of the directory in which the data files are stored. If no directory is specified, the current working directory is used.  
 This includes both Btrieve files and the data dictionary files (.DDF). Data dictionary files describe the structure of Btrieve data.  
 Default None  
 GUI Tab [General tab](#) on page 546

## Default Logon ID

Attribute LogonID (UID)  
 Description The default user ID that is used to connect to your database. Your ODBC application may override this value or you may override it in the logon dialog box or connection string.  
 Valid Values *userid*  
 where *userid* is a valid user ID with permissions to access the database.  
 Default None  
 GUI Tab [Advanced tab](#) on page 547

## Description

Attribute	Description (n/a)
Description	An optional long description of a data source. This description is not used as a runtime connection attribute, but does appear in the ODBC.INI section of the Registry and in the <code>odbc.ini</code> file.
Valid Values	<i>string</i>  where <i>string</i> is a description of a data source.
Default	None
GUI Tab	<a href="#">General tab</a> on page 546

## File Open Cache

Attribute	FileOpenCache (FOC)
Description	The maximum number of used file handles to cache.
Valid Values	0   <i>x</i>  where <i>x</i> is a positive integer.  If set to 0, no file open caching is performed.  If set to <i>x</i> , when a user opens and closes <i>x</i> tables, the tables are not actually closed. The driver keeps them open so that if another query uses one of these tables, the driver does not have to perform another open, which is expensive. The advantage of file open caching is improved performance. The disadvantage is that a user who tries to open the file exclusively may get a file locking conflict even though no one appears to have the file open.
Default	0 (No File Open Caching)
GUI Tab	<a href="#">Advanced tab</a> on page 547

## International Sort

Attribute	IntlSort (IS)
Description	Uses international sort order as defined by your operating system when you issue a Select statement with an Order By clause.
Valid Values	0   1  If set to 1 (Enabled), this order is always alphabetic, regardless of case; the letters are sorted as "A, b, C." Refer to your operating system documentation concerning the sorting of accented characters.  If set to 0 (Disabled), ASCII sort order is used. This order sorts items alphabetically with uppercase letters preceding lowercase letters. For example, "A, b, C" is sorted as "A, C, b."
Default	0 (Disabled)
GUI Tab	<a href="#">Advanced tab</a> on page 547



## Password

Attribute	Password (PWD)
Description	The password that you must enter if your Scalable SQL data dictionary files have security restrictions set. The Password option cannot be specified through the Administrator GUI.
Valid Values	<i>pwd</i> where <i>pwd</i> is a valid password.
Default	None
GUI Tab	n/a

## Use Long Qualifiers

Attribute	UseLongQualifiers (ULQ)
Description	Determines whether the driver uses long path names.
Valid Values	0   1  If set to 1 (Enabled), path names can be a maximum of 255 characters.  If set to 0 (Disabled), path names can be a maximum of 128 characters.
Default	0 (Disabled)
GUI Tab	<a href="#">Advanced tab</a> on page 547

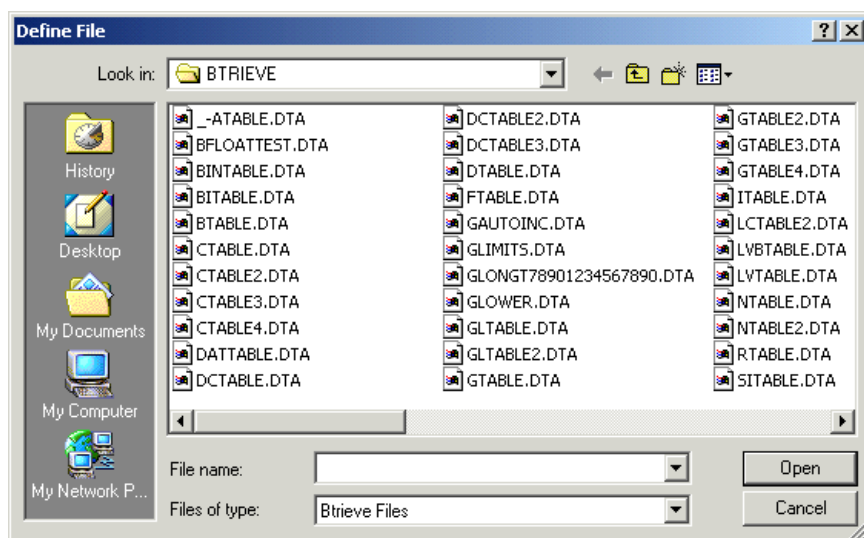
---

## Defining Table Structure

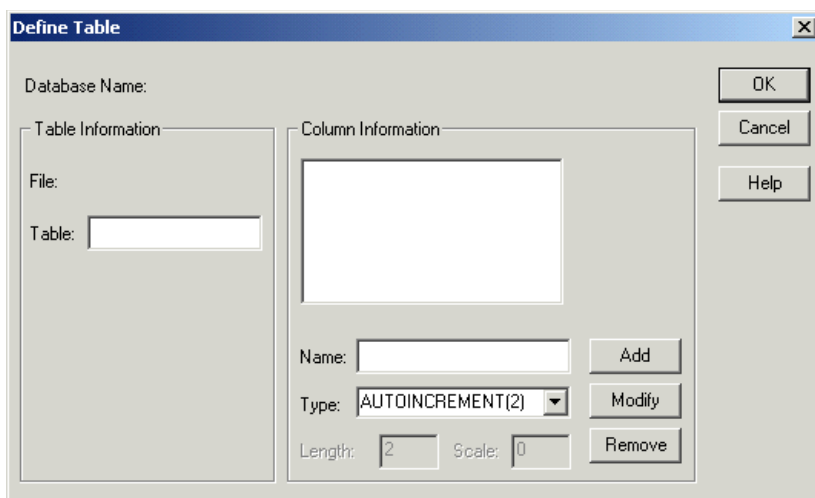
Because Btrieve does not store any column information in the data file, you may need to define its structure. Tables created by the Btrieve driver or by Scalable SQL will not require this. Utilities are also available from Pervasive Software that will perform this operation.

**To define the structure of a file:**

- 1 Display the ODBC Btrieve Driver Setup dialog box through the ODBC Administrator. Click the **Advanced** tab; then, click **Define** to display the Define File dialog box.



- 2 In the Define File dialog box, select the file you want to define and click **Open** to display the Define Table dialog box.



**Database Name:** This field displays the directory in which the file you selected in the Define File dialog box is located.

**File:** This field displays the name of the file that you selected in the Define File dialog box.

**Table:** Type the name of the table to be returned by SQLTables. The name can be a maximum of 20 characters and cannot be the same as another defined table in the database. This field is required.

- 3 Specify values in the following fields to define each column. Click **Add** to add the column name to the list box.
 

**Name:** Type the name of the column.

**Type:** Select the data type of the column.

**Length:** Type the length of the column, if applicable.

**Scale:** Type the scale of the column, if applicable.
- 4 To modify an existing column definition, select the column name in the list box. Modify the values for that column name; then, click **Modify**.
- 5 To delete an existing column definition, select a column name in the list box and click **Remove**.
- 6 Click **OK** to define the table.

---

## Data Types

[Table 15-2](#) shows how the Btrieve data types map to the standard ODBC data types. The Btrieve data types are used when you incorporate Btrieve files into a Scalable SQL database.

---

**Table 15-2. Btrieve Data Types**

---

Btrieve	ODBC
Autoincrement(2)	SQL_SMALLINT
Autoincrement(4)	SQL_INTEGER
Bfloat(4)	SQL_REAL
Bfloat(8)	SQL_DOUBLE
Bit	SQL_BIT
Blob	SQL_LONGVARGINARY
Char	SQL_CHAR
Currency	SQL_DECIMAL
Date	SQL_TYPE_DATE
Decimal	SQL_DECIMAL
Float(4)	SQL_REAL
Float(8)	SQL_DOUBLE
Integer(1)	SQL_TINYINT
Integer(2)	SQL_SMALLINT
Integer(4)	SQL_INTEGER
Integer(8)	SQL_BIGINT
Logical(1)	SQL_BIT
Logical(2)	SQL_BIT

**Table 15-2. Btrieve Data Types** (cont.)

<b>Btrieve</b>	<b>ODBC</b>
Lstring	SQL_VARCHAR
Money	SQL_DECIMAL
Note	SQL_LONGVARCHAR
Numeric	SQL_NUMERIC
Numericsts	SQL_NUMERIC
Time	SQL_TYPE_TIME
Timestamp	SQL_TYPE_TIMESTAMP
Unsigned(1)	SQL_TINYINT
Unsigned(8)	SQL_BIGINT
Zstring	SQL_VARCHAR

See ["Retrieving Data Type Information" on page 59](#) for information about retrieving data types.

## Indexes

NOTE: If you define an index using the Btrieve driver, the index will not have the restrictions discussed here.

For query optimization, the Btrieve driver does not use the following:

- Indexes containing all-segment-null keys or any-segment-null keys.
- Any index key that is marked case-insensitive.
- Any index keys where the data type of the index key does not match the data type of the field. The one exception is if the index key is declared as an unsigned integer and the field in the file is declared as signed integer, or vice versa, then the driver assumes the field contains only unsigned quantities and uses the index. Note that this can lead to incorrect results if the field in fact does contain signed quantities.

The Btrieve driver only uses an alternate-collating-sequence (ASC) index key for equality lookups. Additionally, if an ASC key is part of a segmented index, the other index segments are not used for query optimization unless the Where clause contains an equality condition for the ASC key.

---

## Column Names

Column names in SQL statements (such as Select and Insert) can be up to 20 characters long. If column names are in all lowercase, a combination of upper and lowercase, contain blank spaces, or are reserved words, they must be surrounded by the grave character ( ` ) (ASCII 96). For example:

```
SELECT `name` FROM emp
```

---

## Select Statement

You use the SQL Select statement to specify the columns and records to be read. Btrieve Select statements support all the Select statement clauses described in [Chapter 12 “SQL Statements for Flat-File Drivers”](#) in the *DataDirect Connect Series for ODBC Reference*. This section describes the information that is specific to Btrieve.

### Rowid Pseudo-Column

Each Btrieve record contains a special column named Rowid. This field contains a unique number that indicates the record's sequence in the database. You can use Rowid in Where and Select clauses.

Rowid is particularly useful when you are updating records. You can retrieve the Rowid of the records in the database along with the other field values. For example:

```
SELECT last_name, first_name, salary, rowid FROM emp
```

Then, you can use the Rowid of the record that you want to update to ensure that you are updating the correct record and no other. For example:

```
UPDATE emp set salary = 40000 FROM emp WHERE rowid=21
```

The fastest way of updating a single row is to use a Where clause with the Rowid. You cannot update the Rowid column.

Select statements that use the Rowid pseudo-column in the Where clause achieve maximum performance only for exact equality matches. If you use range scans instead of exact equality matches, a full table scan is performed. For example:

```
SELECT * FROM emp WHERE rowid=21    //fast search
SELECT * FROM emp WHERE rowid <=25  //full table scan
```

---

## Alter Table Statement

The Btrieve driver supports the Alter Table statement to add one or more columns to a table or to delete (drop) a single column.

The Alter Table statement has the form:

```
ALTER TABLE table_name {ADD column_name data_type
| ADD (column_name data_type [, column_name data_type]...)
| DROP [COLUMN] column_name}
```

*table\_name* is the name of the table to which you are adding or dropping columns.

*column\_name* assigns a name to the column you are adding or specifies the column you are dropping.

*data\_type* specifies the native data type of each column you add.

For example, to add two columns to the emp table:

```
ALTER TABLE emp (ADD startdate date, dept char 10)
```

You cannot add columns and drop columns in a single statement, and you can drop only one column at a time. For example, to drop a column:

```
ALTER TABLE emp DROP startdate
```

The Alter Table statement fails when you attempt to drop a column upon which other objects, such as indexes or views, are dependent.

---

## Create and Drop Index Statements

The Btrieve driver supports SQL statements to create and delete indexes.

### Create Index

The Create Index statement for Btrieve files has the form:

```
CREATE [UNIQUE] INDEX index_name ON table_name ([field_name [ASC | DESC]
[, field_name
[ASC | DESC]]...)
```

Unique means that Btrieve does not let you insert two records with the same index values.

*index\_name* is the name of the index.

*table\_name* is the name of the table on which the index is to be created.

ASC tells Btrieve to create the index in ascending order. DESC tells Btrieve to create the index in descending order. By default, indexes are created in ascending order. For example:

```
CREATE INDEX lname ON emp (last_name)
```

## Drop Index

The form of the Drop Index statement is:

```
DROP INDEX table_name.index_name
```

*table\_name* is the name of the table from which the index is to be dropped.

*index\_name* is the name of the index.

For example:

```
DROP INDEX emp.lname
```

---

## Isolation and Lock Levels Supported

Btrieve supports isolation level 1 (read committed) only. Btrieve supports record-level locking.

Refer to [Chapter 7 “Locking and Isolation Levels”](#) in the *DataDirect Connect Series for ODBC Reference* for details.

---

## SQL Support

The driver supports the minimum SQL grammar with several core extensions.

---

## ODBC Conformance Level

The driver is Level 1 compliant, that is, it supports all ODBC Core and Level 1 functions.

In addition, the following function is supported: SQLSetPos.

The driver also supports backward and random fetching in SQLExtendedFetch and SQLFetchScroll.

Refer to [Chapter 2 “ODBC API and Scalar Functions”](#) in the *DataDirect Connect Series for ODBC Reference* for a list of the API functions supported by the Btrieve driver.

---

## Number of Connections and Statements Supported

Btrieve files support a single connection and multiple statements per connection.





# 16 The dBASE Driver

The DataDirect Connect *for* ODBC dBASE driver (the dBASE driver) supports the following file types:

- dBASE IV, V
- Clipper
- FoxPro 6.0 with 3.0 functionality only
- FoxPro 3.0
- FoxPro 2.5, 2.6
- FoxPro 3.0 database container (DBC)

The dBASE driver runs the SQL statements directly on dBASE- and FoxPro-compatible files. You do not need to own dBASE or FoxPro products to access these files. The dBASE driver cannot access files that are larger than 2 GB.

The dBASE driver is 32-bit only and is supported in the Windows, UNIX, and Linux environments. See ["Environment-Specific Information" on page 45](#) for detailed information about the environments supported by this driver.

Refer to the readme file shipped with your DataDirect Connect product for the file name of the dBASE driver.

---

## Driver Requirements

There are no client requirements for the dBASE driver.

---

## Configuring and Connecting to Data Sources

After you install the driver, you configure data sources to connect to the database. See [Chapter 1 "Quick Start Connect" on page 35](#) for an explanation of different types of data sources. The data source contains connection options that allow you to tune the driver for specific performance. If you want to use a data source but need to change some of its values, you can either modify the data source or override its values at connection time through a connection string.

If you choose to use a connection string, you must use specific connection string attributes. See ["Using a Connection String" on page 568](#) and [Table 16-1 on page 569](#) for an alphabetical list of driver connection string attributes and their initial default values.

## Data Source Configuration in the UNIX/Linux odbc.ini File

On UNIX and Linux, you must set up the proper ODBC environment before configuring data sources. See ["Environment Configuration" on page 37](#) for basic setup information and ["Environment Variables" on page 97](#) for more detail about this procedure.

Data sources for UNIX and Linux are stored in the system information file (by default, odbc.ini). If you have a Motif GUI environment on Linux, you can configure and modify data sources through the DataDirect ODBC Data Source Administrator for Linux (the Linux ODBC Administrator) using a driver Setup dialog box. (See ["Configuration Through the Administrator" on page 100](#) for a detailed explanation of the Administrator.)

If you do not have a GUI environment, you can configure and modify data sources directly by editing the odbc.ini file and storing default connection values there. See ["Configuration Through the System Information \(odbc.ini\) File" on page 103](#) for detailed information about the specific steps necessary to configure a data source.

[Table 16-1 on page 569](#) lists driver connection string attributes that must be used in the odbc.ini file to set the value of the attributes. Note that only the long name of the attribute can be used in the file. The default listed in the table is the initial default value when the driver is installed.

## Data Source Configuration through a GUI

On Windows, data sources are stored in the Windows Registry. You can configure and modify data sources through the ODBC Administrator using a driver Setup dialog box, as described in this section.

On UNIX and Linux, data sources are stored in the odbc.ini file. On Linux, you can configure and modify data sources through the Linux ODBC Administrator using a driver Setup dialog box, as described in this section.

**NOTE:** This book shows dialog box images that are specific to Windows. If you are using the drivers in the Linux environment, the dialog box that you see may differ slightly from the Windows version. Windows-only and UNIX-only connection options are specifically noted by icons in the Setup dialog box descriptions.

When the driver is first installed, the values of its connection options are set by default. These values appear on the driver Setup dialog box tabs when you create a new data source. You can change these default values by modifying the data source. In the following procedure, the description of each tab is followed by a table that lists the connection options for that tab and their initial default values. This table links you to a complete description of the options and their connection string attribute equivalents. The connection string attributes are used to override the default values of the data source if you want to change these values at connection time.

## dBASE

To configure a dBASE data source:

### 1 Start the ODBC Administrator:

- On Windows, start the ODBC Administrator by selecting its icon from the DataDirect Connect program group.
- On Linux, change to the *install\_dir/tools* directory and, at a command prompt, enter:

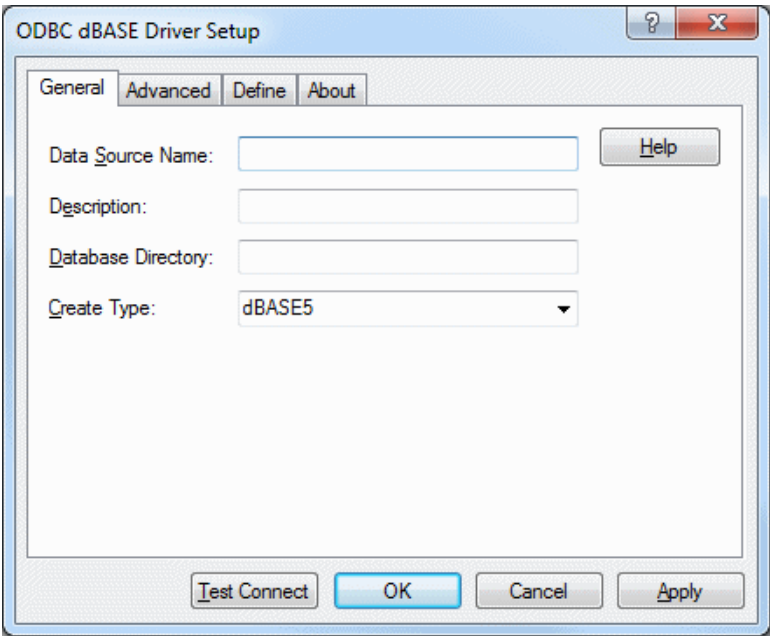
```
odbcadmin
```

where *install\_dir* is the path to the product installation directory.

### 2 Select a tab:

- **User DSN:** If you are configuring an existing user data source, select the data source name and click **Configure** to display the driver Setup dialog box.  
  
If you are configuring a new user data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.
- **System DSN:** If you are configuring an existing system data source, select the data source name and click **Configure** to display the driver Setup dialog box.  
  
If you are configuring a new system data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.
- **File DSN:** If you are configuring an existing file data source, select the data source file and click **Configure** to display the driver Setup dialog box.  
  
If you are configuring a new file data source, click **Add** to display a list of installed drivers; then, select a driver. Click **Advanced** if you want to specify attributes; otherwise, click **Next** to proceed. Specify a name for the data source and click **Next**. Verify the data source information; then, click **Finish** to display the driver Setup dialog box.

The General tab of the Setup dialog box appears by default.

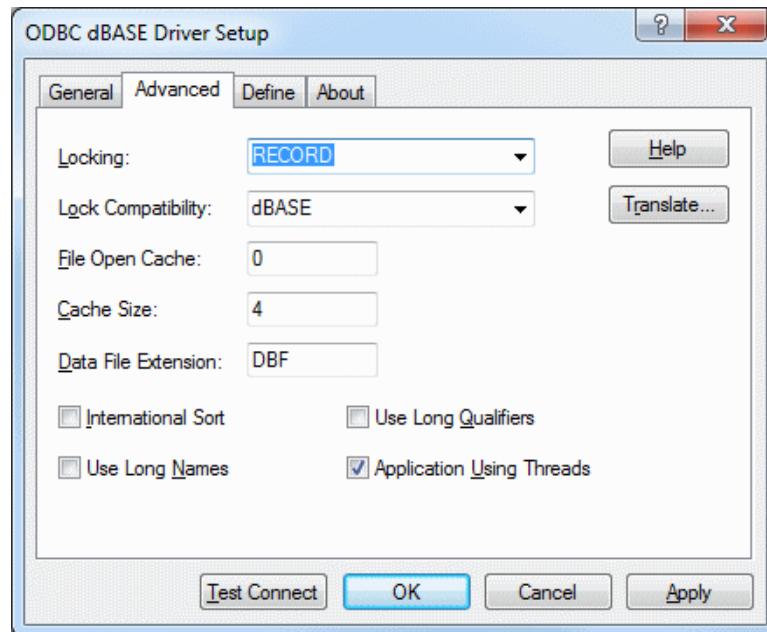


NOTE: The General tab displays the only fields that are required for creating a data source. The fields on all other tabs are optional, unless noted otherwise.

- 3 On this tab, provide values for the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: General	Default
<a href="#">Data Source Name (see page 572)</a>	None
<a href="#">Description (see page 573)</a>	None
<a href="#">Database Directory (see page 572)</a>	None
<a href="#">Create Type [dBASE] (see page 571)</a>	dBASE5

- 4 Optionally, click the **Advanced** tab to specify data source settings.



On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Advanced	Default
<a href="#">Locking (see page 575)</a>	RECORD
<a href="#">Lock Compatibility (see page 574)</a>	dBASE
<a href="#">File Open Cache (see page 573)</a>	0
<a href="#">Cache Size (see page 570)</a>	4
<a href="#">Data File Extension (see page 571)</a>	DBF
<a href="#">International Sort (see page 574)</a>	Disabled
<a href="#">Use Long Names (see page 576)</a>	Disabled
<a href="#">Use Long Qualifiers (see page 576)</a>	Disabled
<a href="#">ApplicationUsingThreads (see page 570)</a>	Enabled
<a href="#">IANAApCodePage (see page 574)</a>	4 (ISO 8559-1 Latin-1)
UNIX ONLY	

**Translate:** Click **Translate** to display the Select Translator dialog box, which lists the translators specified in the ODBC Translators section of the Registry. Progress 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.

- 5 If you use index files that have different names than their corresponding data files and you have not defined this association, click the **Define** tab. See ["Defining Index Attributes on Windows" on page 576](#) for step-by-step instructions.

- 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.
  - 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 incorrect environment or connection value, it displays 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.

## FoxPro 3.0 DBC

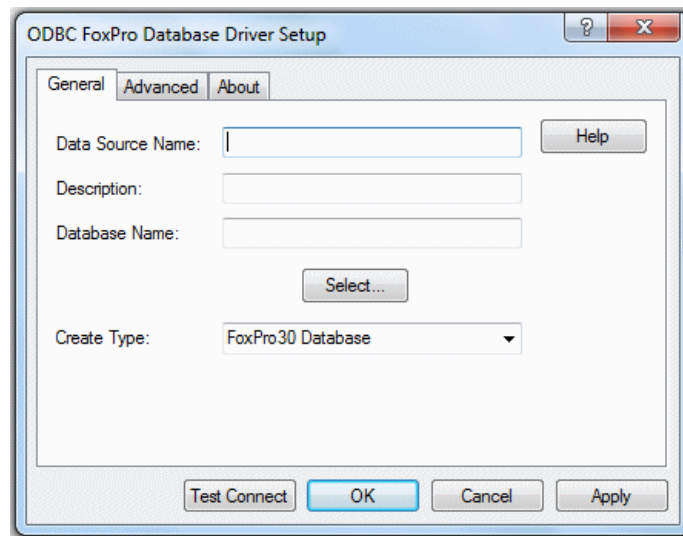
To configure a FoxPro 3.0 database container data source:

- 1 Start the ODBC Administrator:
  - On Windows, start the ODBC Administrator by selecting its icon from the DataDirect Connect program group.
  - On Linux, change to the `install_dir/tools` directory and, at a command prompt, enter:
 

```
odbcadmin
```

 where `install_dir` is the path to the product installation directory.
- 2 Select a tab:
  - **User DSN:** If you are configuring an existing user data source, select the data source name and click **Configure** to display the driver Setup dialog box.  
 If you are configuring a new user data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.
  - **System DSN:** If you are configuring an existing system data source, select the data source name and click **Configure** to display the driver Setup dialog box.  
 If you are configuring a new system data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.
  - **File DSN:** If you are configuring an existing file data source, select the data source file and click **Configure** to display the driver Setup dialog box.  
 If you are configuring a new file data source, click **Add** to display a list of installed drivers; then, select a driver. Click **Advanced** if you want to specify attributes; otherwise, click **Next** to proceed. Specify a name for the data source and click **Next**. Verify the data source information; then, click **Finish** to display the driver Setup dialog box.

The General tab of the Setup dialog box appears by default.



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 this tab, provide values for the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

#### Connection Options: General

#### Default

Data Source Name (see page 572)

None

Description (see page 573)

None

Database Name (see page 572)

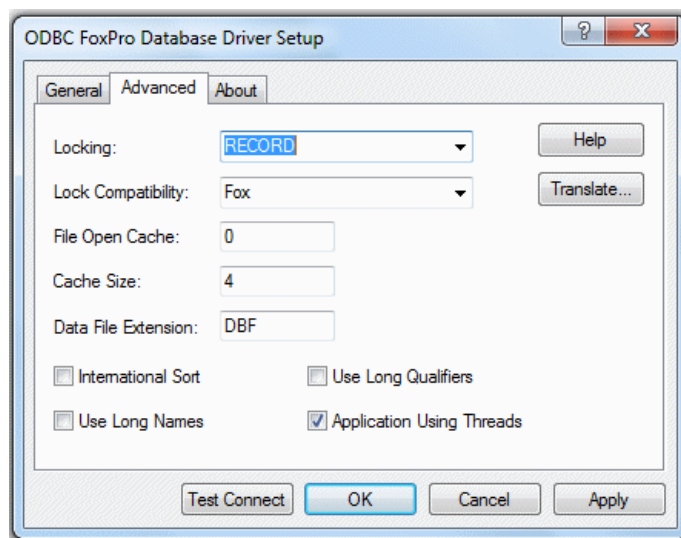
None

Create Type [FoxPro] (see page 571)

FoxPro30 Database

Click **Select** to choose the directory and .DBC file that you want to use.

- 4 Optionally, click the **Advanced** tab to specify data source settings.



On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Advanced	Default
<a href="#">Locking (see page 575)</a>	RECORD
<a href="#">Lock Compatibility (see page 574)</a>	Fox
<a href="#">File Open Cache (see page 573)</a>	0
<a href="#">Cache Size (see page 570)</a>	4
<a href="#">Data File Extension (see page 571)</a>	DBF
<a href="#">International Sort (see page 574)</a>	Disabled
<a href="#">Use Long Names (see page 576)</a>	Disabled
<a href="#">Use Long Qualifiers (see page 576)</a>	Disabled
<a href="#">ApplicationUsingThreads (see page 570)</a>	Enabled
<a href="#">IANAAppCodePage (see page 574)</a>	4 (ISO 8559-1 Latin-1)
UNIX ONLY	

**Translate:** Click **Translate** to display the Select Translator dialog box, which lists the translators specified in the ODBC Translators section of the Registry. Progress 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.

- 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.
  - 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 incorrect environment or connection value, it displays an appropriate error message. Click **OK**.
- 6 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.

## Using a Connection String

If you want to use a connection string for connecting to a database, or if your application requires it, you must specify either a DSN (data source name), a File DSN, or a DSN-less connection in the string. The difference is whether you use the `DSN=`, `FILEDSN=`, or the `DRIVER=` keyword in the connection string, as described in the ODBC specification. A DSN or FILEDSN connection string tells the driver where to find the default connection information. Optionally, you may specify *attribute=value* pairs in the connection string to override the default values stored in the data source.

The DSN connection string has the form:

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



The FILEDSN connection string has the form:

```
FILEDSN=filename.dsn[;attribute=value[;attribute=value]...]
```

The DSN-less connection string specifies a driver instead of a data source. All connection information must be entered in the connection string because the information is not stored in a data source.

The DSN-less connection string has the form:

```
DRIVER=[{driver_name}] [;attribute=value[;attribute=value]...]
```

[Table 16-1](#) lists the long and short names for each attribute, as well as the initial default value when the driver is first installed. You can specify either long or short names in the connection string.

An example of a DSN connection string with overriding attribute values for dBASE is:

```
DSN=DBASE FILES;LCK=NONE;IS=0
```

A FILEDSN connection string is similar except for the initial keyword:

```
FILEDSN=DBASE.dsn;LCK=NONE;IS=0
```

A DSN-less connection string must provide all necessary connection information:

```
DRIVER=DataDirect 7.1 dBASEFile (*.dbf);DB=C:\DBASE;CT=dBASE5
```

---

## Connection Option Descriptions

The following connection option descriptions are listed alphabetically by the GUI name that appears on the driver Setup dialog box. The connection string attribute name, along with its short name, is listed immediately underneath the GUI name. For example:

### Application Using Threads

Attribute    ApplicationUsingThreads (AUT)

In most cases, the GUI name and the attribute name are the same; however, some exceptions exist. If you need to look up an option by its connection string attribute name, please refer to the alphabetical table of connection string attribute names.

Also, a few connection string attributes, for example, Password, do not have equivalent options that appear on the GUI. They are in the list of descriptions alphabetically by their attribute names.

[Table 16-1](#) lists the connection string attributes supported by the dBASE driver.

---

**Table 16-1. dBASE Attribute Names**

---

Attribute (Short Name)	Default
<a href="#">ApplicationUsingThreads (AUT)</a>	1 (Enabled)
<a href="#">CacheSize (CSZ)</a>	4

**Table 16-1. dBASE Attribute Names** (cont.)

Attribute (Short Name)	Default
CreateType (CT) [dBASE]	dBASE5
CreateType (CT)) [FoxPro]	FoxPro30 Database
Database (DB) [dBASE]	None
Database (DB) [FoxPro]	None
DataFileExtension (DFE)	DBF
DataSourceName (DSN)	None
Description (n/a)	None
ExtensionCase (EC)	UPPER
FileOpenCache (FOC)	0 (no file open caching)
IANAAppCodePage (IACP)	4 (ISO 8559-1 Latin-1)
UNIX ONLY	
IntlSort (IS)	0 (Disabled)
LockCompatibility (LCOMP)	dBASE
Locking (LCK)	RECORD
UseLongNames (ULN)	0 (Disabled)
UseLongQualifiers (ULQ)	0 (Disabled)

## ApplicationUsingThreads

Attribute	ApplicationUsingThreads (AUT)
Description	Determines whether the driver works with applications using multiple ODBC threads.
Valid Values	0   1
	If set to 1 (Enabled), the driver works with single-threaded and multi-threaded applications.
	If set to 0 (Disabled), the driver does not work with multi-threaded applications. If using the driver with single-threaded applications, this value avoids additional processing required for ODBC thread-safety standards.
Default	1 (Enabled)
GUI tab	<a href="#">Advanced tab</a> on page 565 [dBASE] <a href="#">Advanced tab</a> on page 567 [FoxPro]

## Cache Size

Attribute	CacheSize (CSZ)
Description	The number of 64 KB blocks the driver uses to cache database records. The larger the number of blocks, the better the performance.

Valid Values 0 |  $x$

where  $x$  is a positive integer that specifies the number of 64 KB blocks for caching.

If set to 0, no records are cached.

If set to  $x$ , the specified number of 64 KB blocks are set aside for caching. The maximum number of blocks you can set depends on the system memory available. If the cache size is greater than 0, when browsing backwards, you are not able to see updates made by other users until you run the Select statement again.

Default 4

GUI tab [Advanced tab](#) on page 565 [dBASE]  
[Advanced tab](#) on page 567 [FoxPro]

## Create Type [dBASE]

Attribute CreateType (CT) [dBASE]

Description The type of table or index to be created on a Create Table or Create Index statement.

Valid Values dBASE4 | dBASE5 | Clipper | FoxPro25 | FoxPro30

Default dBASE5

GUI tab [General tab](#) on page 564

## Create Type [FoxPro]

Attribute CreateType (CT)) [FoxPro]

Description The type of table or index to be created on a Create Table or Create Index statement.

Valid Value FoxPro30 Database

Default FoxPro30 Database

GUI tab [General tab](#) on page 567

## Data File Extension

Attribute DataFileExtension (DFE)

Description A one- to three-character file name extension to use for data files.

Valid Values *ext*

where *ext* is the name of the one- to three-character file name extension.

This value is used for all Create Table statements. Sending a Create Table using an extension other than the value specified for this option causes an error.

In other SQL statements, such as Select or Insert, users can specify an extension other than the one specified for this connection option. The Data File Extension value is used when no extension is specified.

The file extension cannot be one the driver already uses, such as MDX or CDX.

Default	DBF
GUI tab	<a href="#">Advanced tab</a> on page 565 [dBASE] <a href="#">Advanced tab</a> on page 567 [FoxPro]

## Data Source Name

Attribute	DataSourceName (DSN)
Description	The name of a data source in your Windows Registry or odbc.ini file.
Valid Values	<i>string</i>

where *string* is the name of a data source.

Default	None
GUI Tab	<a href="#">General tab</a> on page 564 [dBASE] <a href="#">General tab</a> on page 567 [FoxPro]

## Database Directory

Attribute	Database (DB) [dBASE]
Description	The directory that contains the data files.
Valid Values	<i>database_directory</i>

where *database\_directory* is the full path name of the directory in which the data files are stored. If no directory is specified, the current working directory is used.

Default	None
GUI Tab	<a href="#">General tab</a> on page 564

## Database Name

Attribute	Database (DB) [FoxPro]
Description	The directory that contains the database container (.DBC) files.
Valid Values	<i>database_directory</i>

where *database\_directory* is the full path name of the directory and .DBC file that you want to use.

Default	None
GUI Tab	<a href="#">General tab</a> on page 567

## Description

Attribute	Description (n/a)
Description	An optional long description of a data source. This description is not used as a runtime connection attribute, but does appear in the ODBC.INI section of the Registry and in the <code>odbc.ini</code> file.
Valid Values	<i>string</i>  where <i>string</i> is a description of a data source.
Default	None
GUI Tab	<a href="#">General tab</a> on page 564 [dBASE] <a href="#">General tab</a> on page 567 [FoxPro]

## Extension Case

Attribute	ExtensionCase (EC)
Description	This option determines whether uppercase or lowercase file extensions are accepted.
Valid Values	LOWER   UPPER  When set to UPPER, uppercase extensions are accepted.  When set to LOWER, lowercase extensions are accepted.
Default	UPPER
GUI tab	<a href="#">Advanced tab</a> on page 565 [dBASE] <a href="#">Advanced tab</a> on page 567 [FoxPro]

## File Open Cache

Attribute	FileOpenCache (FOC)
Description	The maximum number of used file handles to cache.
Valid Values	0   <i>x</i>  where <i>x</i> is a positive integer.  If set to 0, no file open caching is performed.  If set to <i>x</i> , when a user opens and closes <i>x</i> tables, the tables are not actually closed. The driver keeps them open so that if another query uses one of these tables, the driver does not have to perform another open, which is expensive. The advantage of file open caching is improved performance. The disadvantage is that a user who tries to open the file exclusively may get a file locking conflict even though no one appears to have the file open.
Default	0 (No File Open Caching)
GUI tab	<a href="#">Advanced tab</a> on page 565 [dBASE] <a href="#">Advanced tab</a> on page 567 [FoxPro]

## IANAAppCodePage

Attribute	IANAAppCodePage (IACP)
Description	An Internet Assigned Numbers Authority (IANA) value. You must specify a value for this option if your application is not Unicode-enabled and/or if your database character set is not Unicode (refer to <a href="#">Chapter 4 “Internationalization, Localization, and Unicode”</a> in the <i>DataDirect Connect Series for ODBC Reference</i> for details). The value you specify must match the database character encoding and the system locale.  The Driver Manager checks for the value of IANAAppCodePage in the following order: <ul style="list-style-type: none"> <li>■ In the connection string</li> <li>■ In the Data Source section of the system information file (odbc.ini)</li> <li>■ In the ODBC section of the system information file (odbc.ini)</li> </ul>
Valid Values	<i>IANA_code_page</i>  where <i>IANA_code_page</i> is one of the valid values listed in <a href="#">Chapter 1 “Values for the Attribute IANAAppCodePage”</a> in the <i>DataDirect Connect Series for ODBC Reference</i> . The value must match the database character encoding and the system locale.
Default	4 (ISO 8559-1 Latin-1)
GUI tab	<a href="#">Advanced tab</a> on page 565 [dBASE] <a href="#">Advanced tab</a> on page 567 [FoxPro]

## International Sort

Attribute	IntlSort (IS)
Description	Uses international sort order as defined by your operating system when you issue a Select statement with an Order By clause.
Valid Values	0   1  If set to 1 (Enabled), this order is always alphabetic, regardless of case; the letters are sorted as "A, b, C." Refer to your operating system documentation concerning the sorting of accented characters.  If set to 0 (Disabled), ASCII sort order is used. This order sorts items alphabetically with uppercase letters preceding lowercase letters. For example, "A, b, C" is sorted as "A, C, b."
Default	0 (Disabled)
GUI tab	<a href="#">Advanced tab</a> on page 565 [dBASE] <a href="#">Advanced tab</a> on page 567 [FoxPro]

## Lock Compatibility

Attribute	LockCompatibility (LCOMP)
Description	The locking scheme the driver uses when locking records.
Valid Values	Clipper   dBASE   Fox   Q+E   Q+EVirtual <ul style="list-style-type: none"> <li>■ Clipper specifies Clipper-compatible locking.</li> <li>■ dBASE specifies Borland-compatible locking.</li> </ul>

- Fox specifies FoxPro-compatible locking.
- Q+E specifies that locks be placed on the actual bytes occupied by the record. Only applications that use the dBASE driver can read and write to the database. Other applications are locked out of the table completely (they cannot even read other records). This locking is compatible with earlier versions of Q+E products.
- Q+EVirtual specifies that locks be placed on bytes beyond the physical end-of-file. Q+EVirtual is the same as Q+E except that other applications can open the table and read the data.

The advantage of using a Q+E locking scheme over dBASE locking is that, on Inserts and Updates, Q+E locks only individual index tags, while dBASE locks the entire index. The following values determine locking support as described:

If you are accessing a table with an application that uses the dBASE driver, your locking scheme does not have to match the Create Type. If you access a table with two applications, however, and only one uses the dBASE driver, set your locking scheme to match the other application. For example, you do not have to set this value to Fox to work with a FoxPro table. But if you are using a FoxPro application simultaneously with an application using the dBASE driver on the same set of tables, set this value to Fox to ensure that your data does not become corrupted.

Default dBASE

GUI tab [Advanced tab](#) on page 565 [dBASE]  
[Advanced tab](#) on page 567 [FoxPro]

## Locking

Attribute Locking (LCK)

Description The level of locking for the database file.

Valid Values NONE | RECORD | FILE

- NONE offers the best performance, but is intended only for single-user environments. See "[Locking](#)" on page 59 for details.
- RECORD locks only the records affected by the statement.
- FILE locks all of the records in the table.

Default RECORD

GUI tab [Advanced tab](#) on page 565 [dBASE]  
[Advanced tab](#) on page 567 [FoxPro]

## Use Long Names

Attribute	UseLongNames (ULN)
Description	Specifies whether to use long file names as table names.
Valid Values	0   1
	<p>If set to 1 (Enabled), the driver uses long file names as table names. The maximum table name length is specific to the environment in which you are running.</p> <p>If set to 0 (Disabled), the driver does not long file names as table names.</p>
Default	0 (Disabled)
GUI tab	<a href="#">Advanced tab</a> on page 565 [dBASE] <a href="#">Advanced tab</a> on page 567 [FoxPro]

## Use Long Qualifiers

Attribute	UseLongQualifiers (ULQ)
Description	Determines whether the driver uses long path names.
Valid Values	0   1
	<p>If set to 1 (Enabled), path names can be a maximum of 255 characters.</p> <p>If set to 0 (Disabled), path names can be a maximum of 128 characters.</p>
Default	0 (Disabled)
GUI tab	<a href="#">Advanced tab</a> on page 565 [dBASE] <a href="#">Advanced tab</a> on page 567 [FoxPro]

---

## Defining Index Attributes on Windows

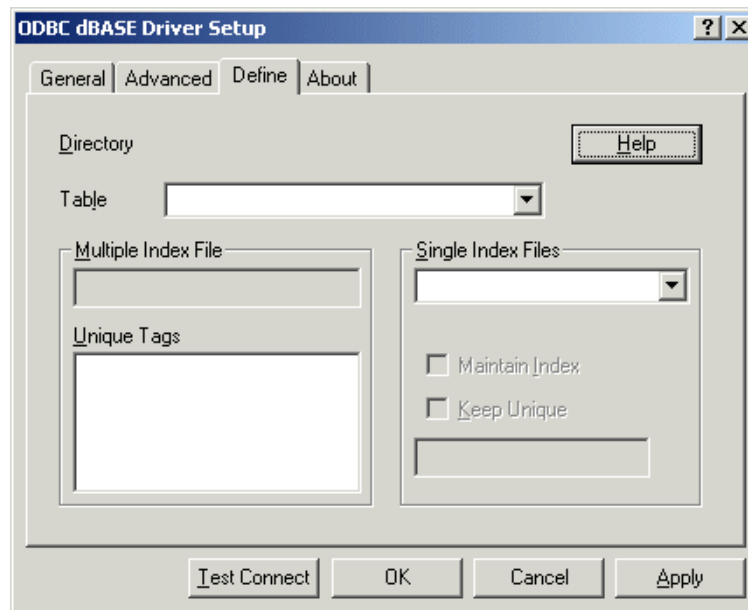
The Define tab of the ODBC dBASE Driver Setup dialog box allows you to define the attributes of index files. With the exception of Clipper, the family of databases that includes dBASE and FoxPro uses a multiple index file associated with a particular table (database file). This index file has a .MDX or .CDX extension and is automatically maintained by the driver. Tags within this index can be marked as unique.

Clipper uses single index files that are not automatically associated with a particular table. You can choose to have the driver maintain an index and choose whether or not the index is unique.



**To define index file attributes:**

- 1 Display the ODBC dBASE Driver Setup dialog box.
- 2 Click the **Define** tab.



On this tab, provide the following information; then, click **Apply**.

**Table:** Type or select the name of the table that contains the database information.

**Multiple Index File:** This field displays the name of any multiple index file (with a .CDX extension or .MDX extension) associated with the table you selected. This index file cannot be marked as unique, but tags within it can be.

**Unique Tags:** This field displays tags associated with the multiple index file. To mark tags as unique, click each one; each one remains selected until you click it again.

**Single Index Files:** The Single Index Files group is active only if you have selected a Clipper table.

Select the file from the drop-down list to define the attributes of a single index file.

**Maintain Index:** Select this check box to associate the specified single index file with the selected table.

**Keep Unique:** Select this check box to specify that the single index file is unique.

- 3 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.

## Defining Index Attributes on UNIX and Linux

Index files for dBASE contain index tags for each index that exists for a database file. These index tags can be marked as unique, that is, the driver will ensure that no duplicate values exist for the columns that define the index tag. The unique attribute is not natively supported by the dBASE or FoxPro products. The enforcement and recognition of the unique attribute is an extension of the dBASE driver. The driver must be notified that index tags are unique. No configuration is needed for unique indexes that were created using the DataDirect Connect for ODBC dBASE driver. When using files that were not created with the dBASE driver, you must define unique index tags as outlined in the following procedure.

In the directory where the database and index files are located, use any text editor, such as vi, to define or edit the QEDBF.INI as follows:

- 1 Create a `[filename]` section where `filename` is the name of the database file. This entry is case-sensitive and the file extension must be included, for example, `[accts.dbf]`.
- 2 In the `[filename]` section, specify the number of unique indexes on the file (`NUMUNIQUE=`) and the index specifications (`UNIQUE#=index_filename,index_tag`). The `index_tag` can be determined by calling the ODBC function `SQLStatistics` and examining the `INDEX_NAME` result column.

For example, to define two unique indexes on the `accts.dbf` database file, the QEDBF.INI would be defined as:

```
[accts.dbf]
NUMUNIQUE=2
UNIQUE0=accts.mdx,ACCT_NAME
UNIQUE1=accts.mdx,ACCT_ID
```

## Data Types

Table 16-2 shows how dBASE data types map to the standard ODBC data types. These dBASE data types can be used in a Create Table statement. Refer to [Chapter 12 “SQL Statements for Flat-File Drivers”](#) in the *DataDirect Connect Series for ODBC Reference* for the syntax of the Create Table statement.

Table 16-3 shows how the additional FoxPro 3.0 data types map to the ODBC data types.

NOTE: A few products can create dBASE files with numbers that do not conform to the precision and scale of the Number column. For example, these products can store 100000 in a column declared as `NUMBER(5,2)`. When this occurs, the dBASE driver displays error 1244, `Unsupported decimal format`. To remedy this situation, multiply the nonconforming column by 1, which converts it to the Float data type. For example:

```
SELECT BADCOL * 1 FROM BADFILE
```

`BADCOL * 1` is evaluated as an expression and is returned as a float value.

**Table 16-2. dBASE Data Types**

<b>dBASE</b>	<b>ODBC</b>
Binary <sup>1</sup>	SQL_LONGVARBINARY
Char <sup>2</sup>	SQL_CHAR
Date	SQL_TYPE_DATE
Float <sup>3</sup>	SQL_DECIMAL
General <sup>4</sup>	SQL_LONGVARBINARY
Logical	SQL_BIT
Memo	SQL_LONGVARCHAR
Numeric	SQL_DECIMAL

1. dBASE V only.

2. 254 characters maximum (1024 for Clipper).

3. dBASE IV and V only.

4. FoxPro and dBASE V only.

**Table 16-3. Additional FoxPro 3.0 Data Types**

<b>FoxPro 3.0</b>	<b>ODBC</b>
Character (binary)	SQL_CHAR
Currency	SQL_DOUBLE
Datetime	SQL_TYPE_TIMESTAMP
Double	SQL_DOUBLE
Integer	SQL_INTEGER
Memo (binary)	SQL_LONGVARBINARY

See ["Retrieving Data Type Information" on page 59](#) for information about retrieving data types.

## Column Names

Column names in SQL statements (such as Select and Insert, for example) can be up to ten characters long. A column name can contain alphanumeric characters and the hyphen character (-). The first character must be a letter (a through z).

## Select Statement

You use a SQL Select statement to specify the columns and records to be read. All of the Select statement clauses described in [Chapter 12 “SQL Statements for Flat-File Drivers”](#) in the *DataDirect Connect Series for ODBC Reference* are supported by dBASE Select statements. This section describes the information that is specific to dBASE, which is Rowid.

### Rowid Pseudo-Column

Each dBASE record contains a special column named Rowid. This field contains a unique number that indicates the record's sequence in the database. For example, a table that contains 50 records has Rowid values from 1 to 50 (if no records are marked deleted). You can use Rowid in Where and Select clauses.

Rowid is particularly useful when you are updating records. You can retrieve the Rowid of the records in the database along with the other field values. For example:

```
SELECT last_name, first_name, salary, rowid FROM emp
```

Then, you can use the Rowid of the record that you want to update to ensure that you are updating the correct record and no other. For example:

```
UPDATE emp set salary = 40000 FROM emp WHERE rowid=21
```

The fastest way of updating a single row is to use a Where clause with the Rowid. You cannot update the Rowid column.

Select statements that use the Rowid pseudo-column in the Where clause achieve maximum performance only for exact equality matches. If you use range scans instead of exact equality matches, a full table scan is performed. For example:

```
SELECT * FROM emp WHERE rowid=21    //fast search
SELECT * FROM emp WHERE rowid <=25  //full table scan
```

## Alter Table Statement

The dBASE driver supports the Alter Table statement to add one or more columns to a table or to delete (drop) a single column.

The Alter Table statement has the form:

```
ALTER TABLE table_name {ADD column_name data_type |
ADD(column_name data_type [, column_name data_type]... ) |
DROP[COLUMN] column_name}
```

*table\_name* is the name of the table to which you are adding or dropping columns.

*column\_name* assigns a name to the column you are adding or specifies the column you are dropping.

*data\_type* specifies the native data type of each column you add.

For example, to add two columns to the emp table:

```
ALTER TABLE emp (ADD startdate date, dept char (10))
```

You cannot add columns and drop columns in a single statement, and you can drop only one column at a time. For example, to drop a column:

```
ALTER TABLE emp DROP startdate
```

The Alter Table statement fails if you attempt to drop a column upon which other objects, such as indexes or views, are dependent.

---

## Create and Drop Index Statements

The dBASE driver supports SQL statements to create and delete indexes.

### Create Index

The type of index you create is determined by the value of the CreateType attribute, which you set in the driver Setup dialog box (for UNIX and Linux, edit the system information file) or as a connection string attribute. The index can be:

- dBASE IV or V (.MDX)
- Clipper (.NTX)
- FoxPro (.CDX)

The syntax for creating an index is:

```
CREATE [UNIQUE] INDEX index_name ON base_table_name
(field_name [ASC | DESC] [, field_name [ASC | DESC]]...)
```

Unique means that the driver creates an ANSI-style unique index over the column and ensures uniqueness of the keys. Use of unique indexes improves performance. ANSI-style unique indexes are different from dBASE-style unique indexes. With ANSI-style unique indexes, you receive an error message when you try to insert a duplicate value into an indexed field. With dBASE-style unique indexes, you do not see an error message when you insert a duplicate value into an indexed field. This is because only one key is inserted in the index file.

*index\_name* is the name of the index file. For FoxPro and dBASE IV or V, this is a tag, which is required to identify the indexes in an index file. Each index for a table must have a unique name.

*base\_table\_name* is the name of the database file whose index is to be created. The .DBF extension is not required; the driver automatically adds it if it is not present. By default, dBASE IV or V index files are named *base\_table\_name*.MDX and FoxPro indexes are named *base\_table\_name*.CDX.

*field\_name* is a name of a column in the dBASE table. You can substitute a valid dBASE-style index expression for the list of field names.

ASC tells dBASE to create the index in ascending order. DESC tells dBASE to create the index in descending order. By default, indexes are created in ascending order. You cannot specify both ASC and DESC orders within a single Create Index statement. For example, the following statement is invalid:

```
CREATE INDEX emp_i ON emp (last_name ASC, emp_id DESC)
```

Table 16-4 shows the attributes of the different index files supported by the dBASE driver. For each type supported, it provides the following details:

- Whether dBASE-style unique indexes are supported
- Whether descending order is supported
- The maximum size supported for key columns
- The maximum size supported for the column specification in the Create Index statement
- Whether production/structural indexes are supported

**Table 16-4. dBASE-Compatible Index Summary**

Create Type .Extension	dBASE UNIQUE	DESC	Max Size of Key Column	Max Size of Column Specification	Production/ Structural Indexes	Supports FOR Expressions
dBASE IV, V .MDX	Yes	Yes	100	220	Yes	Yes
Clipper .NTX	Yes	Yes	250	255	No	Yes
FoxPro .IDX <sup>1</sup>	Yes	Yes	240	255	No	Yes
FoxPro .CDX	Yes	Yes	240	255	Yes	Yes

1. Compact IDX indexes have the same internal structure as a tag in a CDX file. These indexes can be created if the IDX extension is included with the index name in the Create Index statement.

## Drop Index

The syntax for dropping an index is as follows:

```
DROP INDEX table_name.index_name
```

*table\_name* is the name of the dBASE file without the extension.

For FoxPro and dBASE IV or V, *index\_name* is the tag. Otherwise, *index\_name* is the name of the index file without the extension.

To drop the index EMPHIRE.MDX, issue the following statement:

```
DROP INDEX emp.emphire
```

## Pack Statement

When records are deleted from a dBASE file, they are not removed from the file. Instead, they are marked as having been deleted. Also, when memo fields are updated, space may be wasted in the files. To remove the deleted records and free the unused space from updated memo fields, you must use the Pack statement. It has the following form:

```
PACK filename
```

*filename* is the name of the dBASE file to be packed. The .DBF extension is not required; the driver automatically adds the extension if it is not present. For example:

```
PACK emp
```

You cannot pack a file that is opened by another user, and you cannot use the Pack statement in manual commit mode.

For the specified file, the Pack statement performs the following actions:

- Removes all deleted records from the file
- Compresses unused space in the memo file (.DBT or .FPT)
- Removes the entries for all deleted records from .CDX and .MDX files having the same name as the file

## SQL Statements for FoxPro 3.0 Database Containers

The FoxPro DBC driver supports four additional SQL statements:

- Create Database
- Add Table
- Remove Table
- Use

To create a new FoxPro 3.0 database container, use:

```
CREATE DATABASE database_name
```

To add an existing table to the database container, use:

```
ADD TABLE table_name
```

To remove a table from the database container (not delete the table, but unlink it from the database container), use:

```
REMOVE TABLE table_name
```

To set the current database container to an existing database container, use:

```
USE database_name
```

To add or delete columns from a table in a database container, use the Alter Table statement (see ["Alter Table Statement" on page 580](#)).

---

## Locking

With the dBASE driver, you can build and run applications that share dBASE database files on a network. Whenever more than one user is running an application that accesses a shared database file, the applications should lock the records that are being changed. Locking a record prevents other users from locking, updating, or deleting the record.

### Levels of Database Locking

The dBASE driver supports three levels of database locking: NONE, RECORD, and FILE. You can set these levels in:

- The connection string (LCK=)
- The Setup dialog box

No locking offers the best performance, but is intended only for single-user environments.

With record or file locking, the system locks the database files during Insert, Update, Delete, or Select...For Update statements. The locks are released when the user commits the transaction. The locks prevent other users from modifying the locked objects, but they do not lock out readers.

With record locking, only records affected by the statement are locked. Record locking provides better concurrency with other users who also want to modify the database file.

With file locking, all the records in the database file are locked. File locking has lower overhead and may work better if records are modified infrequently, if records are modified primarily by one user, or if a large number of records are modified.

### Limit on Number of Locks

There is a limit on the number of locks that can be placed on a file. If you are accessing a dBASE file from a server, the limit depends on the server (refer to your server documentation).

If you are accessing a dBASE file locally, the limit depends on the buffer space allocated when SHARE.EXE was loaded (refer to your DOS documentation). If you are exceeding the number of locks available, you may want to switch to file locking.

### How Transactions Affect Record Locks

When an Update or Delete statement is run, the driver locks the records affected by that statement. The locks are released after the driver commits the changes. Under manual commit mode, the locks are held until the application commits the transaction. Under autocommit mode, the locks are held until the statement is run.

When a Select...For Update statement is run, the driver locks a record only when the record is fetched. If the record is updated, the driver holds the lock until the changes are committed. Otherwise, the lock is released when the next record is fetched.



---

## Isolation and Lock Levels Supported

dBASE supports isolation level 1 (read committed). It supports both file-level and record-level locking.

Refer to [Chapter 7 “Locking and Isolation Levels”](#) in the *DataDirect Connect Series for ODBC Reference* for details.

---

## SQL Support

The driver supports the minimum SQL grammar.

---

## ODBC Conformance Level

The driver is Level 1 compliant, that is, it supports all ODBC Core and Level 1 functions. In addition, the SQLSetPos function is supported.

The driver also supports backward and random fetching in SQLExtendedFetch and SQLFetchScroll.

Refer to [Chapter 2 “ODBC API and Scalar Functions”](#) in the *DataDirect Connect Series for ODBC Reference* for a list of the API functions supported by the dBASE driver.

---

## Number of Connections and Statements Supported

dBASE supports multiple connections and multiple statements per connection.



# 17 The Informix Driver

The DataDirect Connect *for* ODBC Informix driver (the Informix driver) supports multiple connections to the following Informix database servers when using the appropriate client software.

- Informix Dynamic Server 11.0, 11.5, 11.7
- Informix Dynamic Server 10.0
- Informix Dynamic Server 9.2, 9.3, 9.4

The Informix driver is 32-bit only and is supported in the Windows and UNIX environments, but not in the Linux environments. See ["Environment-Specific Information" on page 45](#) for detailed information about the environments supported by this driver.

Refer to the readme file shipped with your DataDirect product for the file name of the Informix driver.

NOTE: The Informix driver requires Informix client software. Progress DataDirect also provides an Informix driver that does not require any client software to access Informix databases. See ["The Informix Wire Protocol Driver" on page 191](#) for details.

---

## Driver Requirements

This section provides the system requirements for using the Informix driver on all supported platforms.

### Windows

To access supported remote Informix databases through the Informix driver, you need one of the following:

- Informix Connect for Windows platforms, version 2.x
- Informix Client Software Development Kit for Windows platforms, version 2.x

Use the Setnet32 utility supplied by Informix to define servers and the location of the INFORMIX directory. Use llogin to test your connection to the Informix server. The path to the ISQLT09A.DLL must be in your PATH environment variable.

### UNIX (AIX, HP-UX PA-RISC, and Solaris)

The environment variable INFORMIXDIR must be set to the directory where you have installed the Informix client.

For example, the following syntax is valid for C-shell users:

```
setenv INFORMIXDIR /databases/informix
```

For Bourne- or Korn-shell users, the following syntax is valid:

```
INFORMIXDIR=/databases/informix;export INFORMIXDIR
```

In addition, the INFORMIXSERVER variable must be set to the name of the Informix server (as defined in your \$INFORMIXDIR/etc/sqlhosts file). For further details, refer to the Informix documentation.

To access supported remote Informix databases through the Informix driver, you need one of the following:

- On AIX: Informix Client Software Development Kit version 2.2 or higher; or Informix Connect version 2.2 or higher
- On HP-UX and Solaris: Informix Connect version 2.x
- On HP-UX and Solaris: Informix Client Software Development Kit version 2.x

---

## Configuring and Connecting to Data Sources

After you install the driver, you configure data sources to connect to the database. See [Chapter 1 “Quick Start Connect” on page 35](#) for an explanation of different types of data sources. The data source contains connection options that allow you to tune the driver for specific performance. If you want to use a data source but need to change some of its values, you can either modify the data source or override its values at connection time through a connection string.

If you choose to use a connection string, you must use specific connection string attributes. See [“Using a Connection String” on page 593](#) and [Table 17-1 on page 595](#) for an alphabetical list of driver connection string attributes and their initial default values.

### Data Source Configuration in the UNIX/Linux odbc.ini File

On UNIX and Linux, you must set up the proper ODBC environment before configuring data sources. See [“Environment Configuration” on page 37](#) for basic setup information and [“Environment Variables” on page 97](#) for more detail about this procedure.

Data sources for UNIX and Linux are stored in the system information file (by default, odbc.ini). If you have a Motif GUI environment on Linux, you can configure and modify data sources through the DataDirect ODBC Data Source Administrator for Linux (the Linux ODBC Administrator) using a driver Setup dialog box. (See [“Configuration Through the Administrator” on page 100](#) for a detailed explanation of the Administrator.)

If you do not have a GUI environment, you can configure and modify data sources directly by editing the odbc.ini file and storing default connection values there. See [“Configuration Through the System Information \(odbc.ini\) File” on page 103](#) for detailed information about the specific steps necessary to configure a data source.

[Table 17-1 on page 595](#) lists driver connection string attributes that must be used in the odbc.ini file to set the value of the attributes. Note that only the long name of the attribute

can be used in the file. The default listed in the table is the initial default value when the driver is installed.

## Data Source Configuration through a GUI

On Windows, data sources are stored in the Windows Registry. You can configure and modify data sources through the ODBC Administrator using a driver Setup dialog box, as described in this section.

On UNIX and Linux, data sources are stored in the `odbc.ini` file. On Linux, you can configure and modify data sources through the Linux ODBC Administrator using a driver Setup dialog box, as described in this section.

NOTE: This book shows dialog box images that are specific to Windows. If you are using the drivers in the Linux environment, the dialog box that you see may differ slightly from the Windows version. Windows-only and UNIX-only connection options are specifically noted by icons in the Setup dialog box descriptions.

When the driver is first installed, the values of its connection options are set by default. These values appear on the driver Setup dialog box tabs when you create a new data source. You can change these default values by modifying the data source. In the following procedure, the description of each tab is followed by a table that lists the connection options for that tab and their initial default values. This table links you to a complete description of the options and their connection string attribute equivalents. The connection string attributes are used to override the default values of the data source if you want to change these values at connection time.

### To configure an Informix data source:

#### 1 Start the ODBC Administrator:

- On Windows, start the ODBC Administrator by selecting its icon from the DataDirect Connect program group.
- On Linux, change to the `install_dir/tools` directory and, at a command prompt, enter:

```
odbcadmin
```

where `install_dir` is the path to the product installation directory.

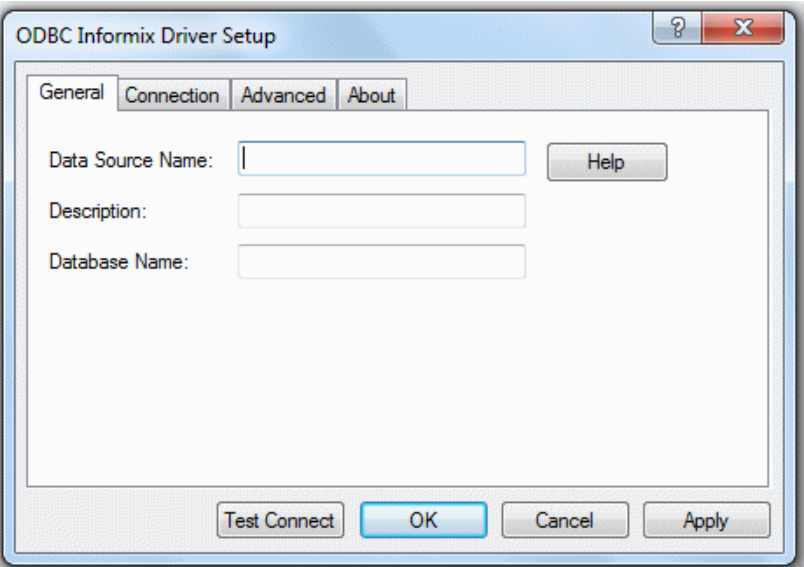
#### 2 Select a tab:

- **User DSN:** If you are configuring an existing user data source, select the data source name and click **Configure** to display the driver Setup dialog box.  
  
If you are configuring a new user data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.
- **System DSN:** If you are configuring an existing system data source, select the data source name and click **Configure** to display the driver Setup dialog box.  
  
If you are configuring a new system data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.

- **File DSN:** If you are configuring an existing file data source, select the data source file and click **Configure** to display the driver Setup dialog box.

If you are configuring a new file data source, click **Add** to display a list of installed drivers; then, select a driver. Click **Advanced** if you want to specify attributes; otherwise, click **Next** to proceed. Specify a name for the data source and click **Next**. Verify the data source information; then, click **Finish** to display the driver Setup dialog box.

The General tab of the ODBC Informix Driver Setup dialog box appears by default.

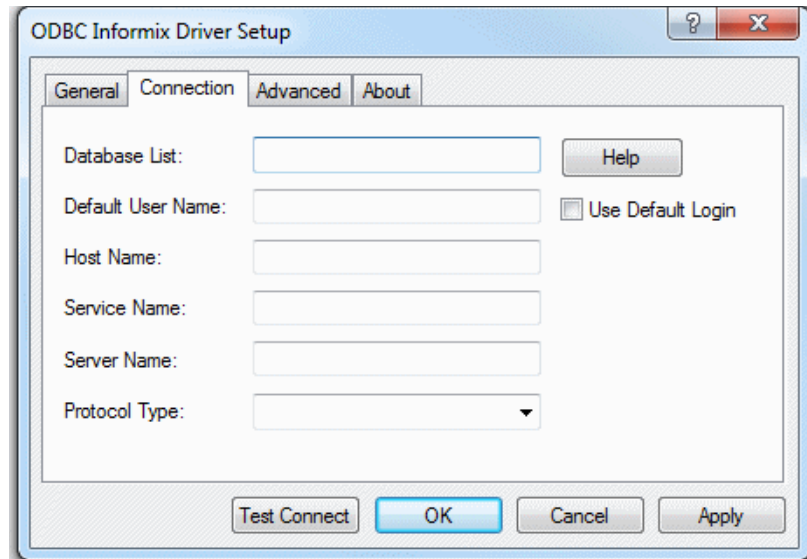


NOTE: The General tab displays only fields that are required for creating a data source. The fields on all other tabs are optional, unless noted otherwise.

- 3 On this tab, provide values for the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: General	Default
<a href="#">Data Source Name (see page 598)</a>	None
<a href="#">Description (see page 598)</a>	None
<a href="#">Database Name (see page 598)</a>	None

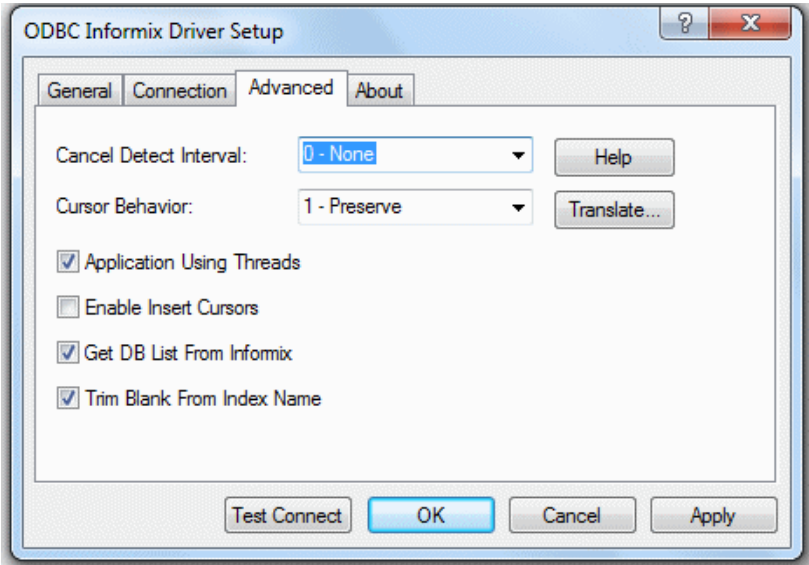
- 4 Optionally, click the **Connection** tab to specify connection information. If you want to configure the data source so that the logon dialog box does not appear during connection, you must specify the connection information on this tab.



On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Connection	Default
<a href="#">Database List (see page 597)</a>	None
<a href="#">Default User Name (see page 598)</a>	None
<a href="#">Use Default Login (see page 601)</a>	Disabled
WINDOWS ONLY	
<a href="#">Host Name (see page 599)</a>	None
<a href="#">Service Name (see page 601)</a>	None
WINDOWS ONLY	
<a href="#">Server Name (see page 601)</a>	None
<a href="#">Protocol Type (see page 600)</a>	None
WINDOWS ONLY	

5 Optionally, click the **Advanced** tab to specify data source settings.



On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Advanced	Default
<a href="#">Cancel Detect Interval (see page 596)</a>	0 - None
<a href="#">Cursor Behavior (see page 597)</a>	0 - Close
<a href="#">Application Using Threads (see page 596)</a>	Enabled
<a href="#">Enable Insert Cursors (see page 599)</a>	Disabled
<a href="#">Get DB List From Informix (GDBLFI) (see page 599)</a>	Enabled
<a href="#">Trim Blank From Index Name (see page 601)</a>	Enabled
<a href="#">IANAAppCodePage (see page 599)</a>	4 (ISO 8559-1 Latin 1)
UNIX ONLY	

**Translate:** Click **Translate** to display the Select Translator dialog box, which lists the translators specified in the ODBC Translators section of the Registry. Progress 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.



- 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 appears (see ["Using a Logon Dialog Box" on page 594](#) for details). 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 incorrect environment or connection value, it displays an appropriate error message.

Verify that all required client software is properly installed. If it is not, you will see the message:

Specified driver could not be loaded due to system error [xxx].

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.

## Using a Connection String

If you want to use a connection string for connecting to a database, or if your application requires it, you must specify either a DSN (data source name), a File DSN, or a DSN-less connection in the string. The difference is whether you use the DSN=, FILEDSN=, or the DRIVER= keyword in the connection string, as described in the ODBC specification. A DSN or FILEDSN connection string tells the driver where to find the default connection information. Optionally, you may specify *attribute=value* pairs in the connection string to override the default values stored in the data source.

The DSN connection string has the form:

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

The FILEDSN connection string has the form:

```
FILEDSN=filename.dsn[;attribute=value[;attribute=value]...]
```

The DSN-less connection string specifies a driver instead of a data source. All connection information must be entered in the connection string because the information is not stored in a data source.

The DSN-less connection string has the form:

```
DRIVER=[{]driver_name[;attribute=value[;attribute=value]...]
```

[Table 17-1 on page 595](#) lists the long and short names for each attribute, as well as the initial default value when the driver is first installed. You can specify either long or short names in the connection string.

An example of a DSN connection string with overriding attribute values for Informix is:

```
DSN=INFORMIX TABLES;DB=PAYROLL
```

A FILEDSN connection string is similar except for the initial keyword:

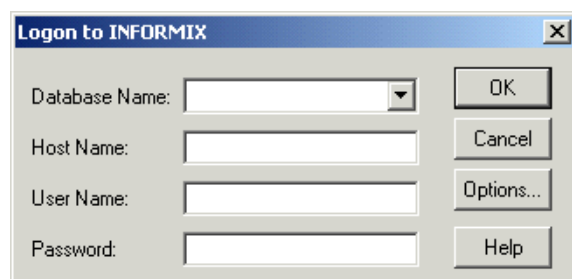
```
FILEDSN=Informix.dsn;DB=DBPAYROLL
```

A DSN-less connection string must provide all necessary connection information:

```
DRIVER=DataDirect 7.1 Informix;DB=PAYROLL;UID=JOHN;PWD=XYZZY
```

## 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.



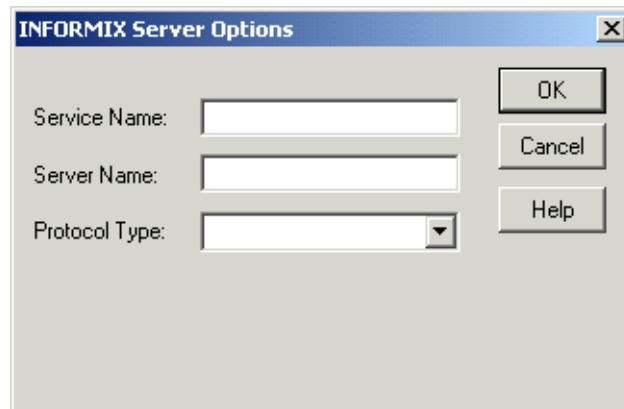
In this dialog box, provide the following information:

- 1 Type the name of the database you want to access, or, on Windows, select the name from the Database Name drop-down list.

On Windows, the names on the list are determined by the status of the **Get DB List From Informix** check box on the Advanced tab of the ODBC Informix driver Setup dialog box. If the check box is selected, the names displayed are returned from the Informix server. If cleared, the names displayed are returned from the user-entered list, which you specify in the Database List field on the Connection tab of the driver Setup dialog box.

- 2 Type the name of the host machine on which the Informix server resides.
- 3 If required, type your user name as specified on the Informix server.
- 4 If required, type your password.

- 5 On Windows, click **Options** to display the Informix Server Options dialog box, where you can change the Service Name, Server Name, and Protocol Type that you specified in the ODBC Informix Driver Setup dialog box. Click **OK** to save your changes.



- 6 Click **OK** to complete the logon and to update these values in the Registry.

## Connection Option Descriptions

The following connection option descriptions are listed alphabetically by the GUI name that appears on the driver Setup dialog box. The connection string attribute name, along with its short name, is listed immediately underneath the GUI name. For example:

### Application Using Threads

Attribute ApplicationUsingThreads (AUT)

In most cases, the GUI name and the attribute name are the same; however, some exceptions exist. If you need to look up an option by its connection string attribute name, please refer to the alphabetical table of connection string attribute names.

Also, a few connection string attributes, for example, Password, do not have equivalent options that appear on the GUI. They are in the list of descriptions alphabetically by their attribute names.

[Table 17-1](#) lists the connection string attributes supported by the Informix driver.

**Table 17-1. Informix Attribute Names**

Attribute (Short Name)	Default
<a href="#">ApplicationUsingThreads (AUT)</a>	1 (Enabled)
<a href="#">CancelDetectInterval (CDI)</a>	0 (None)
<a href="#">Database (DB)</a>	None
<a href="#">Database (DL)</a>	None
<a href="#">DataSourceName (DSN)</a>	None
<a href="#">Description (n/a)</a>	None

**Table 17-1. Informix Attribute Names** (cont.)

Attribute (Short Name)	Default
<a href="#">HostName (HOST)</a>	None
<a href="#">IANAAppCodePage (IACP)</a> UNIX ONLY	4 (ISO 8559-1 Latin 1)
<a href="#">LogonID (UID)</a>	None
<a href="#">Password (PWD)</a>	None
<a href="#">ServerName (SRVR)</a>	None
<a href="#">TrimBlankFromIndexName (TBFIN)</a>	1 (Enabled)
<a href="#">UseDefaultLogin (UDL)</a>	0 (Disabled)

## Application Using Threads

Attribute	ApplicationUsingThreads (AUT)
Description	<p>Determines whether the driver works with applications using multiple ODBC threads.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 602</a> for details.</p>
Valid Values	<p>0   1</p> <p>If set to 1 (Enabled), the driver works with single-threaded and multi-threaded applications.</p> <p>If set to 0 (Disabled), the driver does not work with multi-threaded applications. If using the driver with single-threaded applications, this value avoids additional processing required for ODBC thread-safety standards.</p>
Default	1 (Enabled)
GUI Tab	<a href="#">Advanced tab</a> on page 592

## Cancel Detect Interval

Attribute	CancelDetectInterval (CDI)
Description	<p>Determines whether long-running queries in threaded applications can be cancelled if the application issues a SQLCancel.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 602</a> for details.</p>

Valid Values	0   <i>x</i>
	where <i>x</i> is the number of seconds the driver waits before checking for SQLCancel calls.
	If set to 0 (None), the driver does not allow long-running queries in threaded applications to be canceled, even if the application issues a SQLCancel.
	If set to <i>x</i> (seconds), for every pending query, the driver checks for SQLCancel calls at the specified interval. If the driver determines that a SQLCancel has been issued, the driver cancels the query.
Example	If you specify 5, for every pending query, the driver checks every five seconds to see whether the application has issued a SQLCancel call. If it detects a SQLCancel call, the driver cancels the query.
Default	0 (None)
GUI Tab	<a href="#">Advanced tab</a> on page 592

## Cursor Behavior

Attribute	CursorBehavior (CB)
Description	Determines whether cursors will be preserved or closed at the end of transactions.
Valid Values	0   1
	If set to 1 (Enabled), cursors are held at their current position when transactions end. This value may slow the performance of your database operations.
	If set to set to 0 (Disabled), cursors are closed at the end of transactions.
Default	0 (Disabled)
GUI Tab	<a href="#">Advanced tab</a> on page 592

## Database List

Attribute	Database (DL)
Description	A list of database names that will be displayed in the Logon dialog box if <b>Get DB List From Informix</b> on the Advanced tab is <i>not</i> selected.
Valid Values	<i>database_name</i> [, <i>database_name</i> ] [...]
	where <i>database_name</i> is a database name you want to appear in the Logon dialog box. Separate multiple values with commas.
Example	db1, db2, db3
Default	None
GUI Tab	<a href="#">Connection tab</a> on page 591

## Database Name

Attribute	Database (DB)
Description	The name of the database to which you want to connect.
Valid Values	<i>database_name</i>  where <i>database_name</i> is the name of a valid database.
Default	None
GUI Tab	<a href="#">General tab</a> on page 590

## Data Source Name

Attribute	DataSourceName (DSN)
Description	The name of a data source in your Windows Registry or odbc.ini file.
Valid Values	<i>string</i>  where <i>string</i> is the name of a data source.
Default	None
GUI Tab	<a href="#">General tab</a> on page 590

## Default User Name

Attribute	LogonID (UID)
Description	The default user ID that is used to connect to your database. Your ODBC application may override this value or you may override it in the logon dialog box or connection string.
Valid Values	<i>userid</i>  where <i>userid</i> is a valid user ID with permissions to access the database.
Default	None
GUI Tab	<a href="#">Connection tab</a> on page 591

## Description

Attribute	Description (n/a)
Description	An optional long description of a data source. This description is not used as a runtime connection attribute, but does appear in the ODBC.INI section of the Registry and in the odbc.ini file.
Valid Values	<i>string</i>  where <i>string</i> is a description of a data source.
Default	None
GUI Tab	<a href="#">General tab</a> on page 590

## Enable Insert Cursors

Attribute	EnableInsertCursors (EIC)
Description	Determines whether the driver can use Insert cursors during inserts governed by parameters.
Valid Values	If set to 1 (Enabled), the driver uses Insert cursors.  If set to 0 (Disabled), the driver does not use Insert cursors.
Default	1 (Enabled)
GUI Tab	<a href="#">Advanced tab</a> on page 592

## Get DB List From Informix (GDBLFI)

Attribute	GetDBListFromInformix (GDBLFI)
Description	Determines whether the driver requests the database list to be returned from the Informix server or from the database list that the user entered at driver setup.
Valid Values	0   1  If set to 1 (Enabled), the driver requests the database list from the Informix server.  If set to 0 (Disabled), the driver uses the list that was entered by the user at driver setup.
Default	1 (Enabled)
GUI Tab	<a href="#">Advanced tab</a> on page 592

## Host Name

Attribute	HostName (HOST)
Description	The name of the server to which you want to connect.
Valid Values	<i>server_name</i>  where <i>server_name</i> is the name of the server to which you want to connect.
Default	None
GUI Tab	<a href="#">General tab</a> on page 590

## IANAAppCodePage

Attribute	IANAAppCodePage (IACP)
Description	An Internet Assigned Numbers Authority (IANA) value. You must specify a value for this option if your application is not Unicode-enabled and/or if your database character set is not Unicode (refer to <a href="#">Chapter 4 “Internationalization, Localization, and Unicode”</a> in the

*DataDirect Connect Series for ODBC Reference* for details). The value you specify must match the database character encoding and the system locale.

The Driver Manager checks for the value of IANAAppCodePage in the following order:

- In the connection string
- In the Data Source section of the system information file (odbc.ini)
- In the ODBC section of the system information file (odbc.ini)

Valid Values *IANA\_code\_page*

where *IANA\_code\_page* is one of the valid values listed in [Chapter 1 “Values for the Attribute IANAAppCodePage”](#) in the *DataDirect Connect Series for ODBC Reference*. The value must match the database character encoding and the system locale.

Default 4 (ISO 8559-1 Latin-1)

GUI Tab [Advanced tab](#) on page 592

## Password

Attribute Password (PWD)

Description The password that the application uses to connect to your database. The Password option cannot be specified through the driver Setup dialog box and should not be stored in a data source. It is specified through the Logon dialog box or a connection string.

Valid Values *pwd*

where *pwd* is a valid password.

Default None

GUI Tab n/a

## Protocol Type

Attribute Protocol (PRO)

Description Determines the protocol used by the driver to communicate with the server.

Valid Values olsocspix | olsoctcp | onsocspix | onsoctcp | seipcpip | sesocspix | sesoctcp

Specify the appropriate Informix protocol.

Default None

GUI Tab [Connection tab](#) on page 591



## Server Name

Attribute	ServerName (SRVR)
Description	The name of the Informix server.
Valid Values	<i>server_name</i>  where <i>server_name</i> is a name that uniquely identifies the Informix server.
Default	None
GUI Tab	<a href="#">Connection tab</a> on page 591

## Service Name

Attribute	Service (SERV)
Description	The name of the Informix service. The service name is assigned by the system administrator.
Valid Values	<i>service_name</i>  where <i>service_name</i> is the a name that uniquely identifies the Informix service. This name must be specified as it appears in the services file on the server machine.
Default	None
GUI Tab	<a href="#">Connection tab</a> on page 591

## Trim Blank From Index Name

Attribute	TrimBlankFromIndexName (TBFIN)
Description	Determines whether the driver trims leading spaces from system-generated index names. Some applications cannot process a leading space in index names.
Valid Values	If set to 1 (Enabled), the driver trims leading spaces from system-generated index names.  If set to 0 (Disabled), the driver does not trim leading spaces from system-generated index names.
Default	1 (Enabled)
GUI Tab	<a href="#">Advanced tab</a> on page 592

## Use Default Login

Attribute	UseDefaultLogin (UDL)
Description	Determines where the driver reads login credentials.
Valid Values	0   1  If set to 0 (Disabled), login credentials are read from the Windows Registry, the connection string, or the Logon to Informix dialog box.  If set to 1 (Enabled), login credentials are read directly from the Informix registry.

Default 0 (Disabled)  
GUI Tab [Connection tab](#) on page 591

---

# Performance Considerations

The following connection options can enhance driver performance. You can also enhance performance through efficient application design. Refer to [Chapter 5 “Designing ODBC Applications for Performance Optimization”](#) in the *DataDirect Connect Series for ODBC Reference* for details.

**Application Using Threads (ApplicationUsingThreads):** The driver coordinates concurrent database operations (operations from different threads) by acquiring locks. Although locking prevents errors in the driver, it also decreases performance. If your application does not make ODBC calls from different threads, the driver has no reason to coordinate operations. In this case, the ApplicationUsingThreads attribute should be disabled (set to 0).

NOTE: If you are using a multi-threaded application, you must enable the Application Using Threads option.

**Cancel Detect Interval (CancelDetectInterval):** If your application uses threads, it may allow canceling of long running queries (may issue synchronous SQLCancel calls). If your application does not issue synchronous SQLCancel calls, the driver can improve performance if Cancel Detect Interval is disabled (set to 0). In this case, the driver does not incur the overhead of periodically checking for SQLCancel. In the case where your application does issue synchronous SQLCancel calls, this option should be set to a value that specifies how often the driver checks to see if a long running query has been canceled.

---

# Data Types

[Table 17-2](#) shows how the Informix data types map to the standard ODBC data types.

---

**Table 17-2. Informix Data Types**

---

Informix	ODBC
BLOB	SQL_LONGVARBINARY
BOOLEAN	SQL_BIT
BYTE <sup>1</sup>	SQL_LONGVARBINARY
CHAR	SQL_CHAR
CLOB	SQL_LONGVARCHAR
DATE	SQL_TYPE_DATE
DATETIME YEAR TO FRACTION(f) <sup>2</sup>	SQL_TYPE_TIMESTAMP
DATETIME YEAR TO SECOND	SQL_TYPE_TIMESTAMP

**Table 17-2. Informix Data Types** (cont.)

<b>Informix</b>	<b>ODBC</b>
DATETIME YEAR TO DAY	SQL_TYPE_DATE
DATETIME HOUR TO SECOND	SQL_TYPE_TIME
DATETIME HOUR TO FRACTION(f) <sup>2</sup>	SQL_TYPE_TIME
DECIMAL	SQL_DECIMAL
FLOAT	SQL_DOUBLE
INT8	SQL_BIGINT
INTEGER	SQL_INTEGER
INTERVAL YEAR(p) TO YEAR	SQL_INTERVAL_YEAR
INTERVAL YEAR(p) TO MONTH	SQL_INTERVAL_YEAR_TO_MONTH
INTERVAL MONTH(p) TO MONTH	SQL_INTERVAL_MONTH
INTERVAL DAY(p) TO DAY	SQL_INTERVAL_DAY
INTERVAL DAY(p) TO HOUR	SQL_INTERVAL_DAY_TO_HOUR
INTERVAL DAY(p) TO MINUTE	SQL_INTERVAL_DAY_TO_MINUTE
INTERVAL DAY(p) TO SECOND	SQL_INTERVAL_DAY_TO_SECOND
INTERVAL DAY(p) TO FRACTION(f) <sup>2</sup>	SQL_INTERVAL_DAY_TO_SECOND
INTERVAL HOUR(p) TO HOUR	SQL_INTERVAL_HOUR
INTERVAL HOUR(p) TO MINUTE	SQL_INTERVAL_HOUR_TO_MINUTE
INTERVAL HOUR(p) TO SECOND	SQL_INTERVAL_HOUR_TO_SECOND
INTERVAL HOUR(p) TO FRACTION(f) <sup>2</sup>	SQL_INTERVAL_HOUR_TO_SECOND
INTERVAL MINUTE(p) TO MINUTE	SQL_INTERVAL_MINUTE
INTERVAL MINUTE(p) TO SECOND	SQL_INTERVAL_MINUTE_TO_SECOND
INTERVAL MINUTE(p) TO FRACTION(f) <sup>2</sup>	SQL_INTERVAL_MINUTE_TO_SECOND
INTERVAL SECOND(p) TO SECOND	SQL_INTERVAL_SECOND
INTERVAL SECOND(p) TO FRACTION(f) <sup>2</sup>	SQL_INTERVAL_SECOND
LVARCHAR(p) <sup>2</sup>	SQL_VARCHAR
MONEY	SQL_DECIMAL
NCHAR	SQL_CHAR
NVARCHAR	SQL_VARCHAR
SERIAL	SQL_INTEGER
SERIAL8	SQL_BIGINT
SMALLFLOAT	SQL_REAL
SMALLINT	SQL_SMALLINT
TEXT <sup>1</sup>	SQL_LONGVARCHAR
VARCHAR <sup>1</sup>	SQL_VARCHAR

1. Not supported for Standard Engine databases.

2. (f) can have a value of 1, 2, 3, 4, or 5. The precision is type-dependent and the scale is 5.

The Informix driver does not support any complex data types (for example, set, multiset, list, and named/unnamed abstract types). When the driver encounters a complex type it will return an Unknown Data Type error (SQL State HY000).

See ["Retrieving Data Type Information" on page 59](#) for information about retrieving data types.

---

## MTS Support

On Windows, the driver can take advantage of Microsoft Transaction Server (MTS) capabilities, specifically, the Distributed Transaction Coordinator (DTC) using the XA Protocol. For a general discussion of MTS and DTC, refer to the help file of the Microsoft Transaction Server SDK.

NOTE: The DataDirect Connect *for* ODBC 32-bit drivers can operate in a 64-bit Windows environment; however, they do not support DTC in this environment. Only the DataDirect Connect64 *for* ODBC 64-bit drivers support DTC in a 64-bit Windows environment.

To enable DTC support, you must be using Informix Connect version 2.20 or higher clients.

### To enable support for the DTC:

- 1 Use the **Setnet32** utility supplied by Informix to define:
  - The INFORMIXDIR environment variable, which identifies the location of the client programs, library files, message files, header files, and other Informix software components
  - The INFORMIXSERVER environment variable, which identifies the default database server
  - An Informix server, which identifies either an existing Informix database server or a new one
  - A host name, which identifies the host computer with the database server you want to use
  - A user name, which identifies a user name for an account on the currently selected host computer
  - A password for the specified user name, if required

When enlisting in a distributed transaction, the Informix clients only use the defaults specified in **Setnet32**.
- 2 Run the **regcopy** utility provided with INFORMIX-Connect to copy the registry entries created by **Setnet32** to an area in the registry that is accessible by the DTC. The DTC is a service, and services do not search for configuration information in the Windows registry where **Setnet32** stores client products environment variables. Therefore, if you do not run regcopy after setting the defaults in **Setnet32**, enlistment in a distributed transaction will fail.

For information on using the **Setnet32** and **regcopy** utilities, see the Informix documentation.

---

## Persisting a Result Set as an XML Data File

The driver allows you to persist a result as an XML data file with embedded schema. See ["Persisting a Result Set as an XML Data File" on page 60](#) for details about implementation.

---

## Isolation and Lock Levels Supported

If connected to an Online Server, Informix supports isolation levels 0 (read uncommitted), 1 (read committed), and 3 (serializable). The default is 1. The Standard Engine supports isolation level 0 (read uncommitted) only.

Informix also supports an alternative isolation level 1, called "cursor stability." Your ODBC application can use this isolation level by calling `SQLSetConnectAttr (1040,1)`.

Additionally, if transaction logging has not been enabled for your database, then transactions are not supported by the driver (the driver is always in auto-commit mode).

Informix supports page-level and row-level locking.

Refer to [Chapter 7 "Locking and Isolation Levels"](#) in the *DataDirect Connect Series for ODBC Reference* for details.

---

## SQL Support

The driver supports the core SQL grammar.

---

## ODBC Conformance Level

The driver is Level 1 compliant, that is, it supports all ODBC Core and Level 1 functions.

In addition, the following functions are supported:

- SQLProcedures
- SQLColumnPrivileges
- SQLTablePrivileges
- SQLPrimaryKeys
- SQLForeignKeys
- SQLProcedureColumns

The driver also supports scrollable cursors with `SQLFetchScroll` or `SQLExtendedFetch`.

Refer to [Chapter 2 “ODBC API and Scalar Functions”](#) in the *DataDirect Connect Series for ODBC Reference* for a list of the API functions supported by the Informix driver.

---

## Number of Connections and Statements Supported

The Informix driver supports multiple connections and multiple statements per connection to the Informix database system.

# 18 The Text Driver

The DataDirect Connect *for* ODBC Text driver (the Text driver) supports:

ASCII text files

These files can be printed directly or edited with text editors or word processors, because none of the data is stored in a binary format.

The Text driver is 32-bit only and is supported in the Windows, UNIX, and Linux environments. See ["Environment-Specific Information" on page 45](#) for detailed information about the environments supported by this driver.

The Text driver executes SQL statements directly on the text files. The driver supports Insert statements and inserts the record at the end of the file. You can execute Update and Delete statements conditionally.

The Text driver can access files up to 15 GB in size.

Refer to the readme file shipped with your DataDirect Connect product for the file name of the text driver.

---

## Driver Requirements

There are no client requirements for the Text driver.

---

## Formats for Text Files

Some common formats for text files are listed in [Table 18-1](#).

---

**Table 18-1. Common Text File Formats**

---

Format	Description
Comma-separated values	Commas separate column values, and each line is a separate record. Column values can vary in length. These files often have the .CSV extension.
Tab-separated values	Tabs separate column values, and each line is a separate record. Column values can vary in length.
Character-separated values	Any printable character except single and double quotes can separate column values, and each line is a separate record. Column values can vary in length.

**Table 18-1. Common Text File Formats** (cont.)

Format	Description
Fixed	No character separates column values. Instead, values start at the same position and have the same length in each line. The values appear in fixed columns if you display the file. Each line is a separate record.
Stream	No character separates column values nor records. The table is one long stream of bytes.

Comma-, tab-, and character-separated files are called character-delimited files because values are separated by a special character.

# Configuring Data Sources

After you install the driver, you configure data sources to connect to the database. See [Chapter 1 “Quick Start Connect” on page 35](#) for an explanation of different types of data sources. The data source contains connection options that allow you to tune the driver for specific performance. If you want to use a data source but need to change some of its values, you can either modify the data source or override its values at connection time through a connection string.

If you choose to use a connection string, you must use specific connection string attributes. See [“Using a Connection String” on page 612](#) and [Table 18-2 on page 613](#) for an alphabetical list of driver connection string attributes and their initial default values.

## Data Source Configuration in the UNIX/Linux odbc.ini File

On UNIX and Linux, you must set up the proper ODBC environment before configuring data sources. See [“Environment Configuration” on page 37](#) for basic setup information and [“Environment Variables” on page 97](#) for more detail about this procedure.

Data sources for UNIX and Linux are stored in the system information file (by default, odbc.ini). If you have a Motif GUI environment on Linux, you can configure and modify data sources through the DataDirect ODBC Data Source Administrator for Linux (the Linux ODBC Administrator) using a driver Setup dialog box. (See [“Configuration Through the Administrator” on page 100](#) for a detailed explanation of the Administrator.)

If you do not have a GUI environment, you can configure and modify data sources directly by editing the odbc.ini file and storing default connection values there. See [“Configuration Through the System Information \(odbc.ini\) File” on page 103](#) for detailed information about the specific steps necessary to configure a data source.

[Table 18-2 on page 613](#) lists driver connection string attributes that must be used in the odbc.ini file to set the value of the attributes. Note that only the long name of the attribute



can be used in the file. The default listed in the table is the initial default value when the driver is installed.

## Data Source Configuration through a GUI

On Windows, data sources are stored in the Windows Registry. You can configure and modify data sources through the ODBC Administrator using a driver Setup dialog box, as described in this section.

On UNIX and Linux, data sources are stored in the `odbc.ini` file. On Linux, you can configure and modify data sources through the Linux ODBC Administrator using a driver Setup dialog box, as described in this section.

NOTE: This book shows dialog box images that are specific to Windows. If you are using the drivers in the Linux environment, the dialog box that you see may differ slightly from the Windows version. Windows-only and UNIX-only connection options are specifically noted by icons in the Setup dialog box descriptions.

When the driver is first installed, the values of its connection options are set by default. These values appear on the driver Setup dialog box tabs when you create a new data source. You can change these default values by modifying the data source. In the following procedure, the description of each tab is followed by a table that lists the connection options for that tab and their initial default values. This table links you to a complete description of the options and their connection string attribute equivalents. The connection string attributes are used to override the default values of the data source if you want to change these values at connection time.

### To configure a Text data source:

#### 1 Start the ODBC Administrator:

- On Windows, start the ODBC Administrator by selecting its icon from the DataDirect Connect program group.
- On Linux, change to the `install_dir/tools` directory and, at a command prompt, enter:

```
odbcadmin
```

where `install_dir` is the path to the product installation directory.

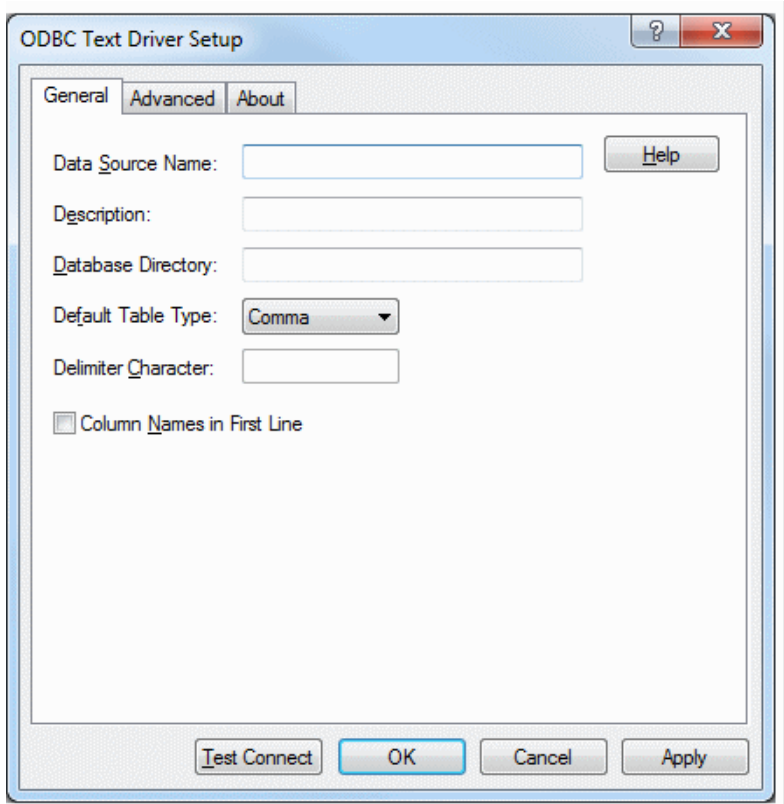
#### 2 Select a tab:

- **User DSN:** If you are configuring an existing user data source, select the data source name and click **Configure** to display the driver Setup dialog box.  
  
If you are configuring a new user data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.
- **System DSN:** If you are configuring an existing system data source, select the data source name and click **Configure** to display the driver Setup dialog box.  
  
If you are configuring a new system data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.

- **File DSN:** If you are configuring an existing file data source, select the data source file and click **Configure** to display the driver Setup dialog box.

If you are configuring a new file data source, click **Add** to display a list of installed drivers; then, select a driver. Click **Advanced** if you want to specify attributes; otherwise, click **Next** to proceed. Specify a name for the data source and click **Next**. Verify the data source information; then, click **Finish** to display the driver Setup dialog box.

The General tab of the Setup dialog box appears by default.

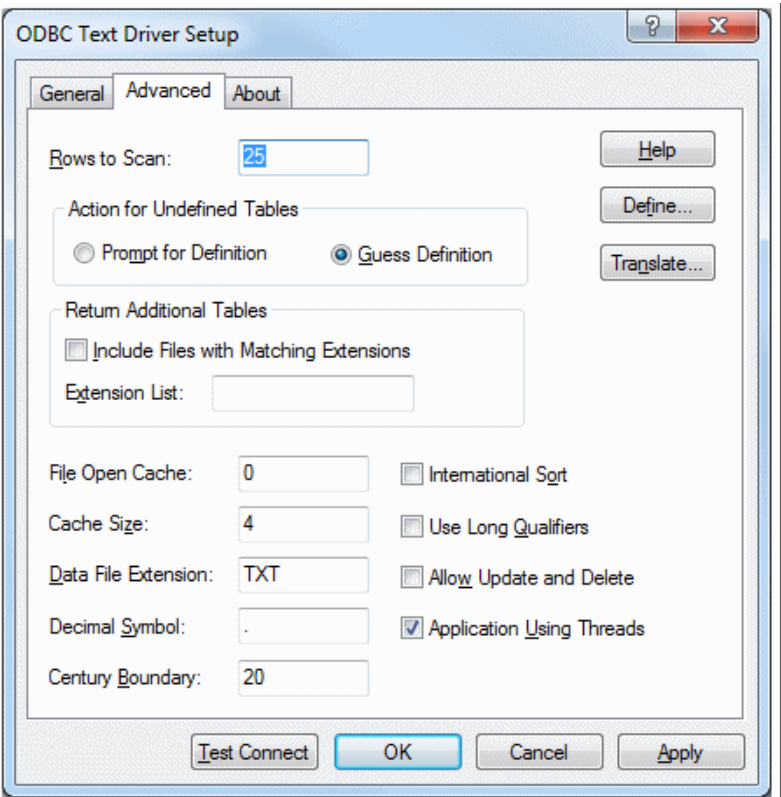


NOTE: The General tab displays the only fields that are required for creating a data source. The fields on all other tabs are optional, unless noted otherwise.

- 3 On this tab, provide values for the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: General	Default
<a href="#">Data Source Name (see page 616)</a>	None
<a href="#">Description (see page 618)</a>	None
<a href="#">Database Directory (see page 616)</a>	None
<a href="#">Default Table Type (see page 617)</a>	Comma
<a href="#">Delimiter Character (see page 617)</a>	None
<a href="#">Column Names in First Line (see page 616)</a>	Disabled

- 4 Optionally, click the **Advanced** tab to specify data source settings.



On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Advanced	Default
<a href="#">Rows to Scan (see page 620)</a>	25
<a href="#">Action for Undefined Tables (see page 614)</a>	GUESS
<a href="#">Include Files with Matching Extensions (see page 619)</a>	Disabled
<a href="#">Extension List (see page 618)</a>	None
<a href="#">File Open Cache (see page 618)</a>	0
<a href="#">International Sort (see page 620)</a>	Disabled
<a href="#">Cache Size (see page 615)</a>	4
<a href="#">Use Long Qualifiers (see page 620)</a>	Disabled
WINDOWS ONLY	
<a href="#">Data File Extension (see page 616)</a>	TXT
<a href="#">Allow Update and Delete (see page 614)</a>	Disabled
<a href="#">Decimal Symbol (see page 617)</a>	. (Period)
<a href="#">Application Using Threads (see page 615)</a>	Enabled
<a href="#">Century Boundary (see page 615)</a>	20
<a href="#">IANAAppCodePage (see page 619)</a>	4 (ISO 8559-1 Latin-1)
UNIX ONLY	

**Define:** Click **Define** to define the structure of your text files as described in ["Defining Table Structure on Windows" on page 621](#).

**Translate:** Click **Translate** to display the Select Translator dialog box, which lists the translators specified in the ODBC Translators section of the Registry. Progress 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.

- 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.
  - 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 incorrect environment or connection value, it displays an appropriate error message. Click **OK**.
- 6 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.

---

## Using a Connection String

If you want to use a connection string for connecting to a database, or if your application requires it, you must specify either a DSN (data source name), a File DSN, or a DSN-less connection in the string. The difference is whether you use the `DSN=`, `FILEDSN=`, or the `DRIVER=` keyword in the connection string, as described in the ODBC specification. A DSN or FILEDSN connection string tells the driver where to find the default connection information. Optionally, you may specify *attribute=value* pairs in the connection string to override the default values stored in the data source.

The DSN connection string has the form:

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

The FILEDSN connection string has the form:

```
FILEDSN=filename.dsn[;attribute=value[;attribute=value]...]
```

The DSN-less connection string specifies a driver instead of a data source. All connection information must be entered in the connection string because the information is not stored in a data source.

The DSN-less connection string has the form:

```
DRIVER=[{ }driver_name[ } ][;attribute=value[;attribute=value]...]
```

[Table 18-2](#) lists the long and short names for each attribute, as well as the initial default value when the driver is first installed. You can specify either long or short names in the connection string.

An example of a DSN connection string with overriding attribute values for Text is:

```
DSN=Text1;DB=C:\TEXTDATA;TT=Comma
```

A FILEDSN connection string is similar except for the initial keyword:

```
FILEDSN=Text1.dsn;DB=C:\TEXTDATA;TT=Comma
```

A DSN-less connection string must provide all necessary connection information:

```
DRIVER=DataDirect 7.1 TextFile (*.*) ;DB=C:\TEXTDATA;TT=Comma
```

## Connection Option Descriptions

The following connection option descriptions are listed alphabetically by the GUI name that appears on the driver Setup dialog box. The connection string attribute name, along with its short name, is listed immediately underneath the GUI name. For example:

### Application Using Threads

Attribute    ApplicationUsingThreads (AUT)

In most cases, the GUI name and the attribute name are the same; however, some exceptions exist. If you need to look up an option by its connection string attribute name, please refer to the alphabetical table of connection string attribute names.

Also, a few connection string attributes, for example, Password, do not have equivalent options that appear on the GUI. They are in the list of descriptions alphabetically by their attribute names.

[Table 18-2](#) lists the connection string attributes supported by the Text driver.

**Table 18-2. Text Attribute Names**

Attribute (Short Name)	Default
<a href="#">AllowUpdateAndDelete (AUD)</a>	0 (Disabled)
<a href="#">ApplicationUsingThreads (AUT)</a>	1 (Enabled)
<a href="#">CacheSize (CSZ)</a>	4
<a href="#">CenturyBoundary (CB)</a>	20
<a href="#">Database (DB)</a>	None
<a href="#">DataFileExtension (DFE)</a>	TXT
<a href="#">DataSourceName (DSN)</a>	None
<a href="#">DecimalSymbol (CS)</a>	. (Period)
<a href="#">Delimiter (DC)</a>	, (Comma)
<a href="#">Description (n/a)</a>	None
<a href="#">ExtraExtensions (EE)</a>	None
<a href="#">FileOpenCache (FOC)</a>	0
<a href="#">FirstLineNames (FLN)</a>	0 (Disabled)

**Table 18-2. Text Attribute Names** (cont.)

Attribute (Short Name)	Default
<a href="#">IANAAppCodePage (IACP)</a> UNIX ONLY	4 (ISO 8559-1 Latin-1)
<a href="#">IntlSort (IS)</a>	0 (Disabled)
<a href="#">ScanRows (SR)</a>	25
<a href="#">TableType (TT)</a>	Comma
<a href="#">UndefinedTable (UT)</a>	GUESS
<a href="#">UseLongQualifiers (ULQ)</a> WINDOWS ONLY	0 (Disabled)

**Action for Undefined Tables**

Attribute	UndefinedTable (UT)
Description	Determines whether the driver prompts the user when it encounters a table for which it has no structure information.
Valid Values	PROMPT   GUESS  Specify PROMPT to prompt the user.  Specify GUESS for the driver to guess the format of the file.
Default	GUESS
GUI Tab	<a href="#">Advanced tab</a> on page 611

**Allow Update and Delete**

Attribute	AllowUpdateAndDelete (AUD)
Description	Allows Update and Delete statements. Because Update and Delete statements cause immediate changes to a text file, only one connection at a time can operate on a file. Each update and delete on a text file can cause significant changes to the file, and performance may be degraded. Consider a more appropriate database form if performance is a significant factor.
Valid Values	0   1  If set to 1 (Enabled), text files are opened exclusively by the current connection.  If set to 0 (Disabled), Update and Delete statements are not allowed.
Default	0 (Disabled)
GUI Tab	<a href="#">Advanced tab</a> on page 611

## Application Using Threads

Attribute	ApplicationUsingThreads (AUT)
Description	Determines whether the driver works with applications using multiple ODBC threads.
Valid Values	0   1  If set to 1 (Enabled), the driver works with single-threaded and multi-threaded applications.  If set to 0 (Disabled), the driver does not work with multi-threaded applications. If using the driver with single-threaded applications, this value avoids additional processing required for ODBC thread-safety standards.
Default	1 (Enabled)
GUI Tab	<a href="#">Advanced tab</a> on page 611

## Cache Size

Attribute	CacheSize (CSZ)
Description	The number of 64 KB blocks the driver uses to cache database records. The larger the number of blocks, the better the performance.
Valid Values	0   $x$  where $x$ is a positive integer that specifies the number of 64 KB blocks for caching.  If set to 0, no records are cached.  If set to $x$ , the specified number of 64 KB blocks are set aside for caching. The maximum number of blocks you can set depends on the system memory available. If the cache size is greater than 0, when browsing backwards, you are not able to see updates made by other users until you run the Select statement again.
Default	4
GUI Tab	<a href="#">Advanced tab</a> on page 611

## Century Boundary

Attribute	CenturyBoundary (CB)
Description	The cutoff year for century inference when converting two-digit dates to four-digit dates.
Valid Values	$xx$  where $xx$ is a two-digit number.  Two-digit dates that are less than the specified year number are converted to 20 $xx$ . Two-digit dates greater than or equal to the number are converted to 19 $xx$ . For example, using the default value of 20, a date of 19 will be interpreted as 2019 and a date of 21 is interpreted as 1921.
Default	20
GUI Tab	<a href="#">Advanced tab</a> on page 611

## Column Names in First Line

Attribute	FirstLineNames (FLN)
Description	Determines whether the driver looks for column names in the first line of the file.  NOTE: The Column Names in First Line setting applies only to tables not previously defined. It also determines the attributes of new tables created with the Create Table statement.
Valid Values	0   1  If set to 1 (Enabled), the driver looks for column names in the first line of the file.  If set to 0 (Disabled), the driver does not look for column names in the first line of the file.
Default	0 (Disabled)
GUI Tab	<a href="#">General tab</a> on page 610

## Data File Extension

Attribute	DataFileExtension (DFE)
Description	A one- to three-character file name extension to use for data files.
Valid Values	<i>ext</i>  where <i>ext</i> is the name of the one- to three-character file name extension.  This value is used for all Create Table statements. Sending a Create Table using an extension other than the value specified for this option causes an error.  In other SQL statements, such as Select or Insert, users can specify an extension other than the one specified for this connection option. The Data File Extension value is used when no extension is specified.
Default	TXT
GUI Tab	<a href="#">Advanced tab</a> on page 611

## Data Source Name

Attribute	DataSourceName (DSN)
Description	The name of a data source in your Windows Registry or odbc.ini file.
Valid Values	<i>string</i>  where <i>string</i> is the name of a data source.
Default	None
GUI Tab	<a href="#">General tab</a> on page 610

## Database Directory

Attribute	Database (DB)
Description	The directory that contains the data files.



Valid Values *database\_directory*

where *database\_directory* is the full path name of the directory in which the data files are stored. If no directory is specified, the current working directory is used.

Default None

GUI Tab [General tab](#) on page 610

## Decimal Symbol

Attribute DecimalSymbol (CS)

Description The decimal separator used when data is stored.

Valid Values , | .

If set to Comma (,), the driver uses a comma as the decimal separator.

If set to Period (.), the driver uses a period as the decimal separator.

The international decimal symbol (.) must be used in DML statements and parameter buffers.

Default . (Period)

GUI Tab [Advanced tab](#) on page 611

## Default Table Type

Attribute TableType (TT)

Description The type of text file (table) that is used when creating a new table and opening an undefined table.

NOTE: The Default Table Type setting applies only to tables not previously defined. It also determines the attributes of new tables created with the Create Table statement.

Valid Values Comma | Tab | Character | Fixed | Stream

The value chosen determines the type of text used for a table: comma-separated, tab-separated, character-separated, fixed length, or stream.

Default Comma

GUI Tab [General tab](#) on page 610

## Delimiter Character

Attribute Delimiter (DC)

Description The character used as a delimiter for character-separated files.

NOTE: The Delimiter Character setting applies only to tables not previously defined. It also determines the attributes of new tables created with the Create Table statement.

Valid Values *x*

where *x* is any printable character except single quotes, double quotes, or semicolons.

Note that it is possible to specify a semicolon if you configure the data source using the Windows ODBC Administrator.

- Default , (Comma)
- GUI Tab [General tab](#) on page 610

Description

- Attribute Description (n/a)
- Description An optional long description of a data source. This description is not used as a runtime connection attribute, but does appear in the ODBC.INI section of the Registry and in the odbcc.ini file.
- Valid Values *string*  
where *string* is a description of a data source.
- Default None
- GUI Tab [General tab](#) on page 610

Extension List

- Attribute ExtraExtensions (EE)
- Description A comma-separated list of file name extensions for the files that you want returned in addition to the extension specified in the Data File Extension field.  
  
NOTE: You must have also enabled the Files with Matching Extensions option.
- Valid Values *ext* | NONE  
  
where *ext* is a file name extension.  
  
To have files with no extensions returned, specify NONE. For example, if some of your files have the extensions TXT and CSV and others have no extension, specify *TXT,CSV,NONE*.  
  
By default, when an application requests a list of tables, only files that have been defined are returned.
- Default None
- GUI Tab [Advanced tab](#) on page 611

File Open Cache

- Attribute FileOpenCache (FOC)
- Description The maximum number of used file handles to cache.
- Valid Values 0 | *x*  
  
where *x* is a positive integer.  
  
If set to 0, no file open caching is performed.

If set to  $x$ , when a user opens and closes  $x$  tables, the tables are not actually closed. The driver keeps them open so that if another query uses one of these tables, the driver does not have to perform another open, which is expensive. The advantage of file open caching is improved performance. The disadvantage is that a user who tries to open the file exclusively may get a file locking conflict even though no one appears to have the file open.

Default 0 (No File Open Caching)

GUI Tab [Advanced tab](#) on page 611

## IANAAppCodePage

Attribute IANAAppCodePage (IACP)

Description An Internet Assigned Numbers Authority (IANA) value. You must specify a value for this option if your application is not Unicode-enabled and/or if your database character set is not Unicode (refer to [Chapter 4 “Internationalization, Localization, and Unicode”](#) in the *DataDirect Connect Series for ODBC Reference* for details). The value you specify must match the database character encoding and the system locale.

The Driver Manager checks for the value of IANAAppCodePage in the following order:

- In the connection string
- In the Data Source section of the system information file (odbc.ini)
- In the ODBC section of the system information file (odbc.ini)

Valid Values *IANA\_code\_page*

where *IANA\_code\_page* is one of the valid values listed in [Chapter 1 “Values for the Attribute IANAAppCodePage”](#) in the *DataDirect Connect Series for ODBC Reference*. The value must match the database character encoding and the system locale.

Default 4 (ISO 8559-1 Latin-1)

GUI Tab [Advanced tab](#) on page 611

## Include Files with Matching Extensions

Attribute n/a

Description Enables the driver to return files with a given file name extension in addition to the extension specified through the Data File Extension option. After enabling this option, specify the file name extensions through the Extension List option.

Valid Values 0 | 1

If set to 1 (Enabled), the driver returns files with the file name extensions specified through the Extension List and Data File Extension options.

If set to 0 (Disabled), the driver returns only files with the file name extension specified through the Data File Extension option.

Default 0 (Disabled)

GUI Tab [Advanced tab](#) on page 611

## International Sort

Attribute	IntlSort (IS)
Description	Uses international sort order as defined by your operating system when you issue a Select statement with an Order By clause.
Valid Values	0   1  If set to 1 (Enabled), this order is always alphabetic, regardless of case; the letters are sorted as "A, b, C." Refer to your operating system documentation concerning the sorting of accented characters.  If set to 0 (Disabled), ASCII sort order is used. This order sorts items alphabetically with uppercase letters preceding lowercase letters. For example, "A, b, C" is sorted as "A, C, b."
Default	0 (Disabled)
GUI Tab	<a href="#">Advanced tab</a> on page 611

## Rows to Scan

Attribute	ScanRows (SR)
Description	The number of rows in a text file that the driver scans to determine the data types in the file.  NOTE: The Rows to Scan setting applies only to tables not previously defined. It also determines the attributes of new tables created with the Create Table statement.
Valid Values	0   <i>x</i>  where <i>x</i> is a positive integer.  If set to 0, all rows in the file are scanned.  If set to <i>x</i> , <i>x</i> rows are scanned to determine the data types in a file.
Default	25
GUI Tab	<a href="#">Advanced tab</a> on page 611

## Use Long Qualifiers

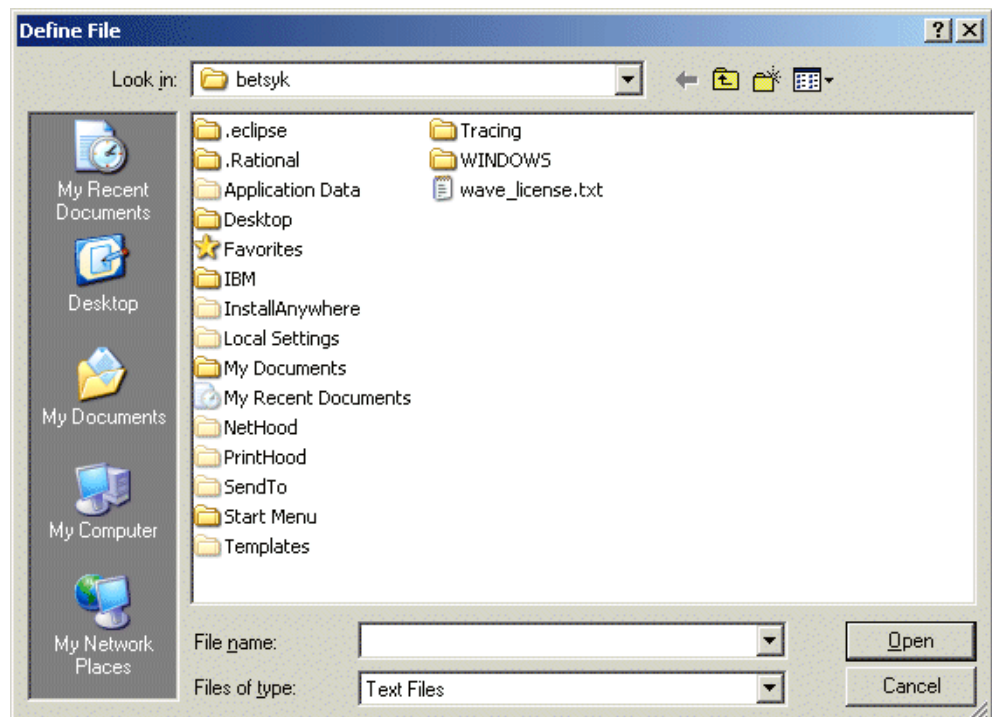
Attribute	UseLongQualifiers (ULQ)
Description	Determines whether the driver uses long path names.
Valid Values	0   1  If set to 1 (Enabled), path names can be a maximum of 255 characters.  If set to 0 (Disabled), path names can be a maximum of 128 characters.
Default	0 (Disabled)
GUI Tab	<a href="#">Advanced tab</a> on page 611

## Defining Table Structure on Windows

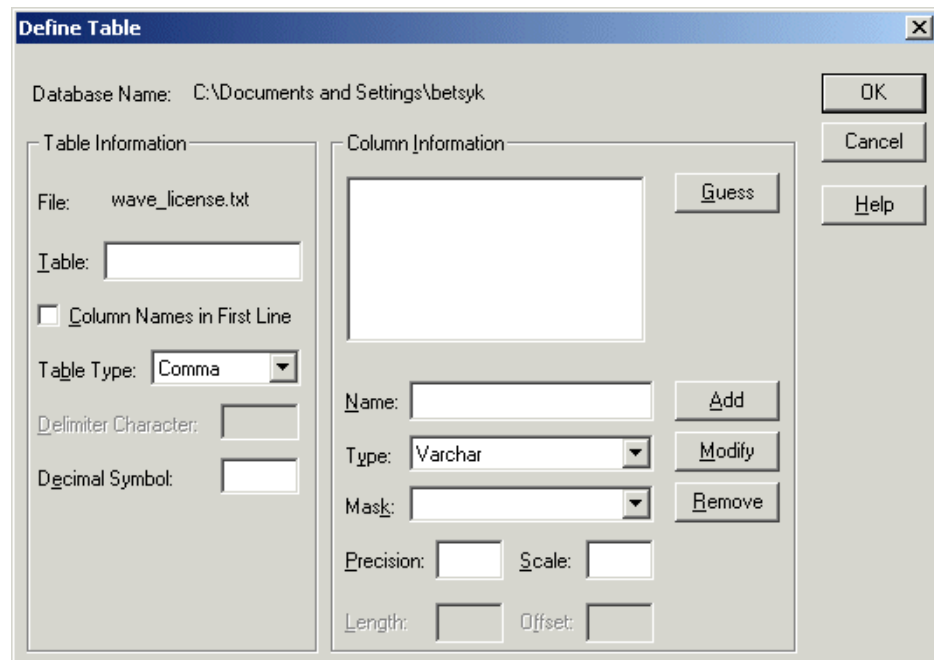
Because text files do not all have the same structure, the driver provides the option of defining the structure of an existing file. Although defining the structure is not mandatory (the driver can attempt to guess the names and types of the columns), this feature is extremely useful.

### To define the structure of a file:

- 1 Display the ODBC Text Driver Setup dialog box through the ODBC Administrator. Click the **Advanced** tab; then, click **Define** to display the Define File dialog box.



- 2 Select the correct file and click **Open** to display the Define Table dialog box.



The **Define Table** dialog box is shown with the following fields and controls:

- Database Name:** C:\Documents and Settings\betsyk
- File:** wave\_license.txt
- Table:** (empty text box)
- ☐ **Column Names in First Line**
- Table Type:** Comma (dropdown menu)
- Delimiter Character:** (empty text box)
- Decimal Symbol:** (empty text box)
- Column Information:** (empty list box)
- Guess** (button)
- Name:** (empty text box) with **Add** (button)
- Type:** Varchar (dropdown menu) with **Modify** (button)
- Mask:** (empty text box) with **Remove** (button)
- Precision:** (empty text box) **Scale:** (empty text box)
- Length:** (empty text box) **Offset:** (empty text box)
- OK**, **Cancel**, and **Help** (buttons)

**Database Name:** This field displays the name of the database directory that you selected in the Define File dialog box.

**File:** This field displays the name of the file that you selected in the Define File dialog box.

**Table:** Type a table name in the Table field. This name specifies the table name associated with the text file you selected earlier. The name can be a maximum of 32 characters and must be unique. This name is returned by SQLTables. By default, it is the file name without its extension (for example, Trc\_read).

**Column Names in First Line:** Select this check box if the first line of the file contains column names; otherwise, do not select this box.

**Table Type:** Select the type of text file, either comma, tab, fixed, character, or stream.

**Delimiter Character:** If the table type is Character, type the delimiter used in character-separated files. The value can be any printable character except single and double quotes.

**Decimal Symbol:** Type the decimal separator used when data is stored. Valid values are a comma or a period. The international decimal symbol (.) must be used in DML statements and parameter buffers.

- 3 If you specified a comma-separated, tab-separated, or character-separated type in the Table Type field, the Guess/Parse button displays Guess. Click **Guess** to have the driver guess at the column names and display them in the list box of the Column Information pane.

If you specified a fixed-length or stream type in the Table Type field, the Guess/Parse button displays Parse. Click **Parse** to have the driver display the Parse Table dialog box and define the table columns.



This dialog box displays the first line of the file. You must mark where *each* field begins and ends by enclosing it in square brackets [ ]. These brackets indicate the position and length of each field value in the record. Click **OK** to close the Parse Table dialog box. The driver will suggest column names in the list box of the Column Information pane.

- 4 If you do not want the driver to guess or parse, enter values in the following fields to define each column. Click **Add** to add the column name to the Column Information box.

**Name:** Type the name of the column.

**Type:** Select the data type of the column. If the field type is Date, the Mask field is enabled and you must select a date mask or type one in. See ["Date Masks" on page 625](#) for more information.

**Mask:** Select a date mask. If you selected Date for the Type field, you must select a date mask for the field or type one in. See ["Date Masks" on page 625](#) for more information.

**Precision:** Type the precision of the column. The precision of numeric data types is defined as the maximum number of digits used by the data type of the column. For character types, this is the length in characters of the data. Note that the precision and scale values determine how numeric data is to be returned.

**Scale:** Type the scale of the column. The scale of numeric data types is defined as the maximum number of digits to the right of the decimal point. Note that the precision and scale values determine how numeric data is to be returned.

**Length:** If you specified a fixed-length table type, type the length, which is the number of bytes the data takes up in storage.

**Offset:** If you specified a fixed-length table type, type the offset, which is the number of bytes from the start of the table to the start of the field.

- 5 To modify an existing column definition, select the column name in the Column Information box. Modify the values for that column name; then, click **Modify**.
- 6 To delete an existing column definition, select a column name in the Column Information box and click **Remove**.
- 7 Click **OK** to define the table.

## Defining Table Structure on UNIX and Linux

Because text files do not all have the same structure, the driver provides the option to define the structure of an existing file. Although defining the structure is not mandatory, because the driver can attempt to guess the names and types of the columns, this feature is extremely useful.

To define the structure of a text file, you create a QETXT.INI file using any plain text editor, such as vi. The file name must be in uppercase. All of the tables you want to define are specified in the QETXT.INI file. When you specify table attributes in QETXT.INI, you override the attributes specified in the system information file (odbc.ini) or in the connection string.

### To define the QETXT.INI file:

- 1 Create a [Defined Tables] section and list all of the tables you are defining. Specify the text file name (in either upper or lowercase, depending on the file) followed by the name you want to give the table, for example:

```
emptext.txt=EMP
```

Table names can be up to 32 characters in length and cannot be the same as another defined table in the database. This name is returned by SQLTables. By default, it is the file name without its extension.

- 2 For each table listed in the [Defined Tables] section, you must specify the text file (FILE=), the table type (TT=), whether the first line of the file contains column names (FLN=), and the delimiter character (DC=).

- Specify the text file name. For example:

```
FILE=emptext.txt
```

- To define the table type, specify how the fields are separated (comma, tab, fixed, or character). For example:

```
TT=COMMA
```

- If the table type is CHARACTER, specify the delimiter character. The value can be any printable character except single and double quotes. For example, if the fields are separated by comma:

```
DC=,
```

- Specify whether the first line of the file contains column names, using 1 for yes and 0 for no. For example:

```
FLN=0
```

- 3 Define the fields in the table, beginning with FIELD1. For each field, specify the field name, field type, precision, scale, length, offset (for fixed tables), and date/time mask. See ["Date Masks" on page 625](#) for information about masks.

Separate the values with commas. For example, to define two fields:

```
FIELD1=EMP_ID,VARCHAR,6,0,6,0,
FIELD2=HIRE_DATE,DATE,10,0,10,0,m/d/yy
```



- 4 Save the file as QETXT.INI. The driver looks for this file in the directory specified by the Database attribute in odbci.ini, or in the current directory.

## Example of QETXT.INI

The following is an example of a QETXT.INI file. This file defines the structure of the emptext.txt file, which is a sample data file shipped with the DataDirect ODBC Text file.

```
[Defined Tables]
emptext.txt=EMP

[EMP]
FILE=emptext.txt
FLN=1
TT=Comma
FIELD1=FIRST_NAME, VARCHAR, 10, 0, 10, 0,
FIELD2=LAST_NAME, VARCHAR, 9, 0, 9, 0,
FIELD3=EMP_ID, VARCHAR, 6, 0, 6, 0,
FIELD4=HIRE_DATE, DATE, 10, 0, 10, 0, m/d/yy
FIELD5=SALARY, NUMERIC, 8, 2, 8, 0,
FIELD6=DEPT, VARCHAR, 4, 0, 4, 0,
FIELD7=EXEMPT, VARCHAR, 6, 0, 6, 0,
FIELD8=INTERESTS, VARCHAR, 136, 0, 136, 0,
```

## Date Masks

Date masks tell the driver how a date is stored in a text file. When a value is inserted into a text file, the date is formatted so that it matches the mask. When reading a text file, the driver converts the formatted date into a date data type.

[Table 18-3](#) lists the symbols to use when specifying the date mask.

**Table 18-3. Date Masks for Text Driver**

Symbol	Description
m	Output the month's number (1–12).
mm	Output a leading zero if the month number is less than 10.
mmm, Mmm, MMM	Output the three-letter abbreviation for the month depending on the case of the Ms (for example, jan, Jan, JAN).
mmmm, Mmmm, MMMM	Output the name of the full month depending on the case of the Ms (for example, january, January, JANUARY).
d	Output the day number (1–31).
dd	Output a leading zero if the day number is less than 10.
ddd, Ddd, DDD	Output the three-letter day abbreviation depending on the case of the Ds (for example, mon, Mon, MON).

**Table 18-3. Date Masks for Text Driver** (cont.)

Symbol	Description
dddd, Dddd, DDDD	Output the name of the full day depending on the case of the Ds (for example, monday, Monday, MONDAY).
yy	Output the last two digits of the year.
yyyy	Output the full four digits of the year.
J	Output the Julian value for the date. The Julian value is the number of days since 4712 BC.
\ - . : , (space)	Special characters used to separate the parts of a date.
\	Output the next character. For example, if the mask is mm/dd/yyyy \A\D, the value appears as 10/01/2003 AD in the text file.
"string", 'string'	Output the string in the text file.

Table 18-4 shows some example date values, masks, and how the date appears in the text file.

**Table 18-4. Date Mask Examples**

Date	Mask	Value
2003-10-01	yyyy-mm-dd	2003-10-01
	m/d/yy	10/1/03
	Ddd, Mmm dd, yyyy	Fri, Oct 01, 2003

# Data Types

Table 18-5 shows how the text file data types are mapped to the standard ODBC data types.

**Table 18-5. Text Data Types**

Text	ODBC
Numeric	SQL_NUMERIC
Date	SQL_TYPE_DATE
Varchar	SQL_VARCHAR

See "Retrieving Data Type Information" on page 59 for information about retrieving data types.

---

## Select Statement

You use a SQL Select statement to specify the columns and records to be read. All of the Select statement clauses described in [Chapter 12 “SQL Statements for Flat-File Drivers”](#) in the *DataDirect Connect Series for ODBC Reference* are supported by the Text driver.

---

## Alter Table Statement

The Text driver supports the Alter Table statement to add one or more columns to a table or to delete (drop) a single column.

The Alter Table statement has the form:

```
ALTER TABLE table_name {ADD column_name data_type |
ADD(column_name data_type [, column_name data_type]... ) |
DROP [COLUMN] column_name}
```

*table\_name* is the name of the table to which you are adding or dropping columns.

*column\_name* assigns a name to the column you are adding or specifies the column you are dropping.

*data\_type* specifies the native data type of each column you add.

For example, to add two columns to the emp table:

```
ALTER TABLE emp (ADD startdate date, dept varchar(10))
```

You cannot add columns and drop columns in a single statement, and you can drop only one column at a time. For example, to drop a column:

```
ALTER TABLE emp DROP startdate
```

The Alter Table statement fails when you attempt to drop a column upon which other objects, such as indexes or views, are dependent.

---

## SQL Support

The driver supports the minimum SQL grammar.

---

## ODBC Conformance Level

The driver is Level 1 compliant, that is, it supports all ODBC Core and Level 1 functions. In addition, the SQLSetPos function is supported.

The driver supports backward and random fetching in SQLExtendedFetch and SQLFetchScroll.

Refer to [Chapter 2 “ODBC API and Scalar Functions”](#) in the *DataDirect Connect Series for ODBC Reference* for a list of the API functions supported by the Text driver.

---

## Number of Connections and Statements Supported

Text files support multiple connections and multiple statements per connection.

# 19 The XML Driver

The DataDirect Connect *for* ODBC XML driver (the XML driver) supports:

Tabular- and hierarchical-formatted XML documents that can be accessed from either a local file system, a web server, or a web service. The three main types of tabular-formatted files that the driver supports are Microsoft Data Islands, ADO 2.5 persisted files, and DataDirect Format.

See ["Supported Tabular Formats for XML Documents" on page 630](#) for more details.

The XML driver is 32-bit only and is supported in the Windows environments. See ["Environment-Specific Information" on page 45](#) for detailed information about the environments supported by this driver.

The XML driver includes a SQL Engine that provides ANSI SQL-92 support. The following table lists the SQL statements that the driver supports for the different types of file formats.

File Format	Select	Create/Drop	Insert	Update	Delete
Tabular, Microsoft Data Islands	X	X	X	X	X
Tabular, ADO 2.5 Persisted	X	X	X	X	X
Tabular, DataDirect	X	X	X	X	X
Tabular, other formats	X		X	X	X
Hierarchical	X				

See ["SQL Support" on page 658](#) for more information.

Refer to the readme file shipped with your DataDirect Connect product for the file name of the XML driver.

---

## Driver Requirements

You must have Internet Explorer 5 or higher installed. You must also have the Microsoft XML parser, msxml4.dll, not a higher version, installed. If you need to download the file, go to the site:

<http://www.microsoft.com>

On the Microsoft site, search on "msxml4.dll". Select the link for downloading the parser.

# Supported Tabular Formats for XML Documents

The three main XML tabular-formats that the XML driver can access are described in [Table 19-1](#). In some instances, you may need to define hints to help the XML driver read the tabular-format of an XML document correctly. See ["Configure Location Dialog Box Descriptions" on page 647](#).

**Table 19-1. Common Tabular Formats for XML Documents**

Format	Description
ADO 2.5 persisted files	<p>These files are identified by a unique schema namespace URL. Although ADO uses the same data types defined by XML-Data, the data types use extensions, such as adding a maximum column width for string columns. ADO 2.5 persisted files are identified by the following unique XML element:</p> <pre>&lt;xml xmlns:s="uuid:BDC6E3F0-6DA3-11d1-A2A3-00AA00C14882"       xmlns:dt="uuid:C2F41010-65B3-11d1-A29F-00AA00C14882"       xmlns:rs="urn:schemas-microsoft-com:rowset"       xmlns:z="#RowsetSchema"&gt;</pre>
DataDirect Format	<p>This XML format conforms to the W3C recommendation for XML schema, Working Draft April 07, 2000. These files are identified by the following unique XML element (schema namespace URL):</p> <pre>&lt;table targetNamespace= "http://www.merant.com/namespaces/datadirect/xmlrecordset" xsi:schemaLocation= "&lt;http://www.merant.com/namespaces/datadirect/xmlrecordset/EMP.xml&gt;" xmlns="http://www.w3.org/1999/XMLSchema" xmlns:xsi= "http://www.w3.org/1999/XMLSchema-instance" xmlns:rs= "http://www.merant.com/namespaces/datadirect/xmlrecordset"&gt;</pre>
Microsoft Data Islands	<p>These islands are identified by the &lt;XML&gt; tag in an HTML document. The Data Island can be embedded in the HTML document. Data Islands can include the following Schema definition and namespace:</p> <pre>&lt;Schema xmlns="urn:schemas-microsoft-com:xml-data" xmlns:dt="urn:schemas-microsoft-com:datatypes"&gt;</pre>

# Hierarchical-Formatted XML Document Support

The XML driver can be configured so that it supports hierarchical-formatted documents. In this case, the driver assumes that the document that it is accessing can contain more than one table. The driver scans the document to locate all tables; the available tables are visible through a SQLTables operation. Then, the driver does a second scan to gather each table's column information and to determine a data type for each column.

The following is an example of a hierarchical document:

```
<?xml version="1.0"?>
  <purchaseOrder orderDate="1999-10-20">
    <shipTo country="US">
      <name>Alice Smith</name>
      <street>123 Maple Street</street>
      <city>Mill Valley</city>
      <state>CA</state>
      <zip>90952</zip>
    </shipTo>
    <billTo country="US">
      <name>Robert Smith</name>
      <street>8 Oak Avenue</street>
      <city>Old Town</city>
      <state>PA</state>
      <zip>95819</zip>
    </billTo>
    <comment>Hurry, my lawn is going wild!</comment>
    <items>
      <item partNum="872-AA">
        <productName>Lawnmower</productName>
        <quantity>1</quantity>
        <USPrice>148.95</USPrice>
        <comment>Confirm this is electric</comment>
      </item>
      <item partNum="926-AA">
        <productName>Baby Monitor</productName>
        <quantity>1</quantity>
        <USPrice>39.98</USPrice>
        <shipDate>1999-05-21</shipDate>
      </item>
    </items>
  </purchaseOrder>
```

First, the XML driver returns two tables: "purchaseOrder" and "items." Two tables are returned because two items are found for a single purchase order. The XML driver found commonality of child elements.

Second, the XML driver determines which columns are in a specific table. An `_ID` column, which is essentially a primary key, is automatically generated for each table. If a table is determined to be a child of another table, then it is given a second generated column. The name of this column is prefixed with the parent table's name and ends with `_ID`, for example, `_purchaseOrder_ID`.

Consider the previous example document. The items table will receive two generated columns, `_ID` and `_purchaseOrder_ID`, which are assigned an integer data type. The purchaseOrder table receives only the `_ID` column, because it does not have a parent table.

The tables returned from the example file include the following columns:

Table	Columns	
items	_ID	quantity
	_purchaseOrder_ID	USPrice
	partNum	comment
	productName	shipDate
purchaseOrder	_ID	billTo_country
	orderDate	billTo_name
	shipTo_country	billTo_street
	shipTo_name	billTo_city
	shipTo_street	billTo_state
	shipTo_city	billTo_zip
	shipTo_state	comment
	shipTo_zip	

## Column Data Types

The XML driver determines the column data types by inspecting the column values. The data type determination limits its data types to a subset of the DataDirect Format data types, as listed in the following table. For a complete list of DataDirect Format data types, see Table 19-6 on page 656.

Data Type	Sample Values
wvchar	"Foo", "best320"
varbinary	"27AB2F9C"
int	"34", "-7000"
unsignedint	"0", "123456789"
long	"-12345678012345"
unsignedlong	"123456789012345"
boolean	"true", "false"
date	1963-12-19
time	10:09:58
timeinstant	1963-12-19T10:09:58
decimal	1245.678



## Defining Locations

When configuring an XML data source, you must define the location of the XML or HTML documents that the driver will access. The locations can be either from a local file system or from a Web server.

The types of locations are:

Folder	Implies that each XML file is a single table. When defining a Folder location, you specify only a directory as the location (not a directory and a file name), for example, C:\xmlsample.
XML Document	Implies that the full path to the XML document, including the XML file name, is the location. Using this type of location, each document can have one or more tables and can be a hierarchical-formatted XML document. When defining an XML Document location, you specify a path and an XML file name as the location, for example:  C:\xmlsample\file.xml  You can also specify a web service through a URL, for example:  http://xxx.company.com/search=XML&mode=books
HTML Document	Implies the use of an HTML document with embedded XML Data Islands. Using this type of location, each document can have one or more tables. When defining an HTML Document location, you specify a path and an HTML file name, for example, C:\htmlsample\file.html, as the location.

## Specifying Table Names in SQL Statements

When defining locations, you specify a name for the location along with a directory, or path and file name. For example, suppose you define two locations for a data source, a Folder location and an XML Document location. The Folder location is on a local filing system and the XML Document location is on a web server with a URL prefix of <http://www.acme.com/xmldata>.

For example:

The Folder location:  
c:\xmldata\xmlsample as LOC1

The XML Document location: <http://www.acme.com/xmldata/doc.xml> as LOC2

For complete information about how to configure locations in an XML data source, see ["Configuring and Connecting to Data Sources" on page 635](#).

If you are connected to this data source and the data source had the "Show Manufactured Schemas" option set as the Schema Mode (see the Schema Mode option under ["Configuring and Connecting to Data Sources" on page 635](#)) and then you performed an unqualified SQLTables operation, you would get the following results.

Schema name	Table name
LOC1#	FILE1
LOC1#	FILE2
LOC2#	TABLE1
LOC2#	TABLE2

Location names are fabricated into the schema name by adding a # symbol to the end of the location name.

NOTE: If you had the "Show Virtual Schemas" option set, the above table would have "XML" listed in the Schema name column.

To fully qualify a table name in a SQL statement, you could use the following:

LOC1#.FILE1

or

XML.FILE1

LOC2#.TABLE2

or

XML.TABLE2

This design gives you a simpler table name qualifier. This is an important advantage given the complexity of URL names, and the requirement to double quote them in SQL statements. For example, the following query uses a fully qualified table name for an XML Document location:

```
SELECT * FROM "http://www.acme.com/xmldata/doc.xml#TABLE2" WHERE productName=
'lawnmower'
```

Compare that to the same query using a location name:

```
SELECT * FROM LOC2#.TABLE2 WHERE productName='lawnmower'
```

Another example demonstrating the Folder location is as follows:

```
SELECT * FROM "c:\xmldata\xmlsample\FILE1.XML" WHERE productName='lawnmower'
```

Compare that to the same query using a location name:

```
SELECT * FROM LOC1#.FILE1 WHERE productName='lawnmower'
```

# Configuring and Connecting to Data Sources

After you install the driver, you configure data sources to connect to the database. See [Chapter 1 “Quick Start Connect” on page 35](#) for an explanation of different types of data sources. The data source contains connection options that allow you to tune the driver for specific performance. If you want to use a data source but need to change some of its values, you can either modify the data source or override its values at connection time through a connection string.

If you choose to use a connection string, you must use specific connection string attributes. See [“Using a Connection String” on page 641](#) and [Table 19-2 on page 642](#) and [Table 19-3 on page 647](#) for an alphabetical list of driver connection string attributes and their initial default values.

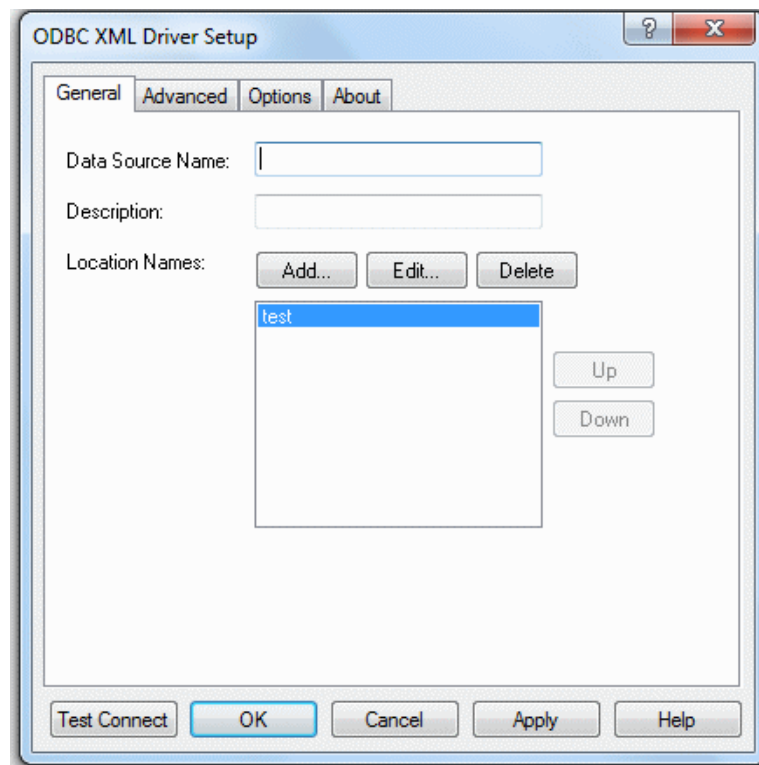
On Windows, data sources are stored in the Windows Registry. You can configure and modify data sources through the ODBC Administrator using a driver Setup dialog box.

When the driver is first installed, the values of its connection options are set by default. These values appear on the driver Setup dialog box tabs when you create a new data source. You can change these default values by modifying the data source. In the following procedure, the description of each tab is followed by a table that lists the connection options for that tab and their initial default values. This table links you to a complete description of the options and their connection string attribute equivalents. The connection string attributes are used to override the default values of the data source if you want to change these values at connection time.

## To configure an XML data source:

- 1 Start the ODBC Administrator by selecting its icon from the DataDirect Connect program group; then, select a tab:
  - **User DSN:** If you are configuring an existing user data source, select the data source name on the User DSN tab and click **Configure** to display the driver Setup dialog box.  
  
If you are configuring a new user data source, click **Add** on the User DSN tab to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.
  - **System DSN:** To configure a new system data source, click **Add** on the System DSN tab to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.
  - **File DSN:** If you are configuring an existing file data source, select the data source name on the File DSN tab and click **Configure** to display the driver Setup dialog box.  
  
If you are configuring a new file data source, click **Add** on the File DSN tab to display a list of installed drivers. Select the driver and click **Next**. Specify a name for the data source and click **Next**. Verify the data source information; then, click **Finish** to display the driver Setup dialog box.

The General tab of the Setup dialog box appears by default.



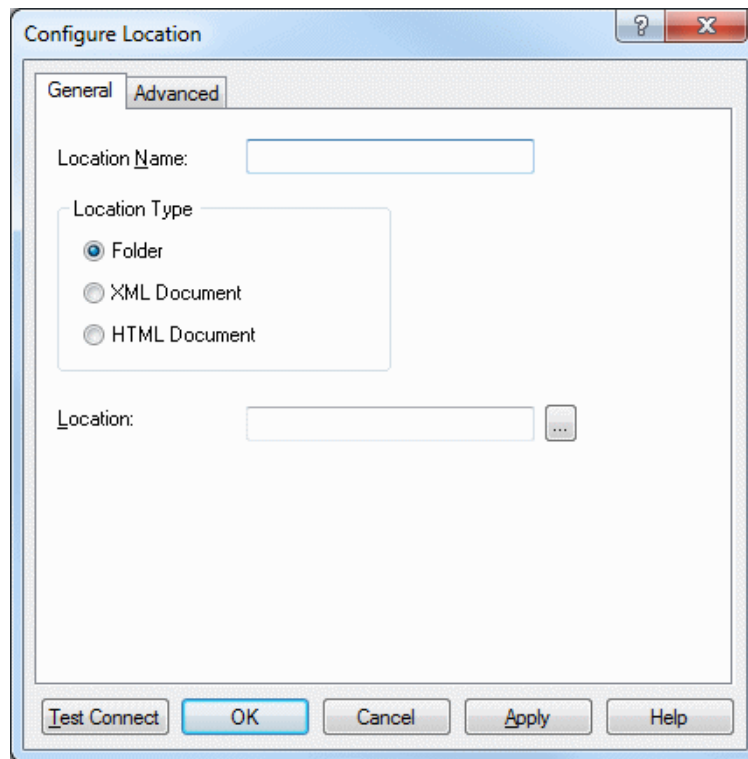
NOTE: The General tab displays only fields that are required for creating a data source. The fields on all other tabs are optional, unless noted otherwise.

- 2 On the General tab, provide the following information; then, click **Apply**.

Connection Options: General	Default
Data Source Name (see page 643)	None
Description (see page 643)	None
Location Names (see page 644)	None

- 3 If you want to edit or delete a location name, or change its position in the list, select it; then, click **Edit**, **Delete**, **Up**, or **Down** as appropriate.

- 4 If you want to define a location, click **Add**. The Configure Location dialog box appears.



- 5 On the General tab of the Configure Location dialog box, provide the following required information; then, click **Apply**.

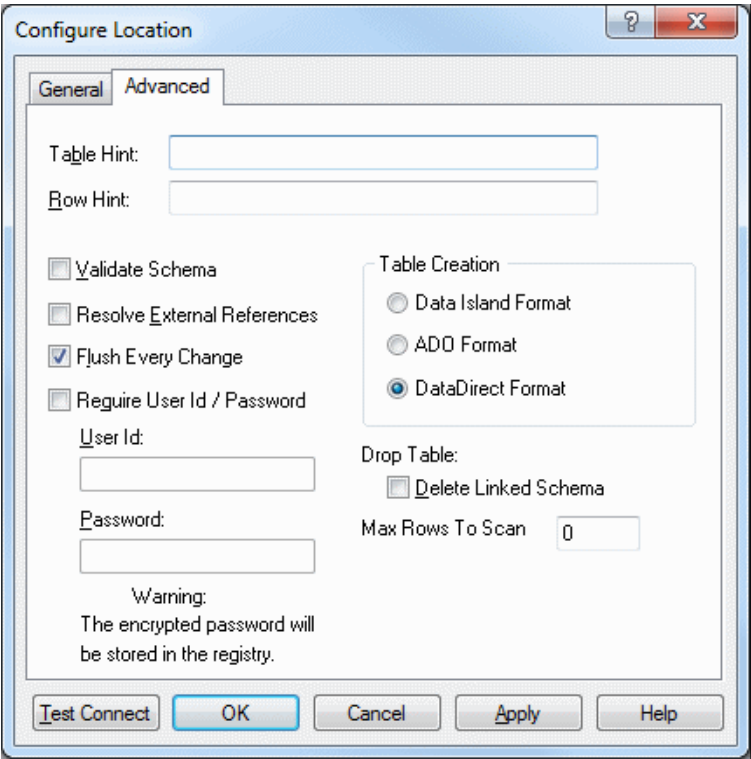
Configure Location Options: General	Default
<a href="#">Location Name (see page 649)</a>	None
<a href="#">Location Type (see page 649)</a>	None
<a href="#">Location (see page 649)</a>	localhost

**Location:** Either type the full path to the location you are defining or click the select button:



to select a path.

- 6 Optionally, click the **Advanced** tab of the Configure Location dialog box to specify additional information about the location you are defining.

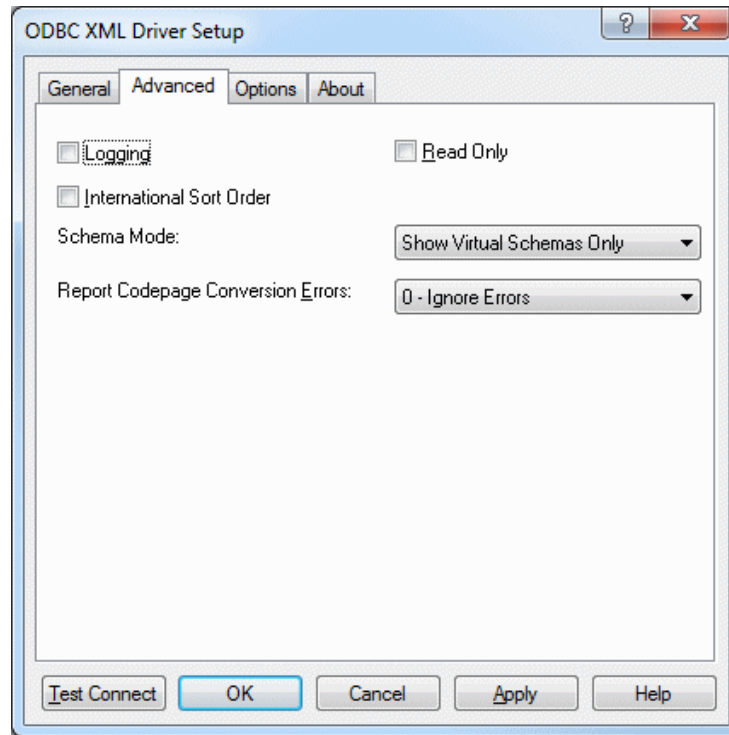


On this tab, provide any of the following optional information; then, click **Apply**.

Configure Location Options: Advanced	Default
Table Hint (see page 652)	None
Row Hint (see page 651)	None
Validate Schema (see page 652)	Disabled
Resolve External References (see page 651)	Disabled
Flush Every Change (see page 648)	Enabled
Require User ID/Password (see page 650)	Disabled
User ID (see page 652)	None
Password (see page 650)	None
Table Creation (see page 651)	DataDirect Format
Delete Linked Schema (see page 648)	Disabled
Max Rows to Scan (see page 649)	0

- 7 You can click **Test Connect** to attempt to connect to the location.
- 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 incorrect environment or connection value, it displays an appropriate error message. Click **OK**.

- 8 Click **OK** to return to the ODBC XML driver Setup dialog box or **Cancel**. If you click **OK**, the values you have specified become the defaults for this location.
- 9 Optionally, click the **Advanced** tab of the ODBC XML driver Setup dialog box to specify data source settings.



On this tab, provide any of the following optional information; then, click **Apply**.

**Connection Options: Advanced**

Logging (see page 644)

International Sort Order (see page 644)

Schema Mode (see page 645)

Report Codepage Conversion Errors (see page 645)

Read Only (see page 645)

**Default**

Disabled

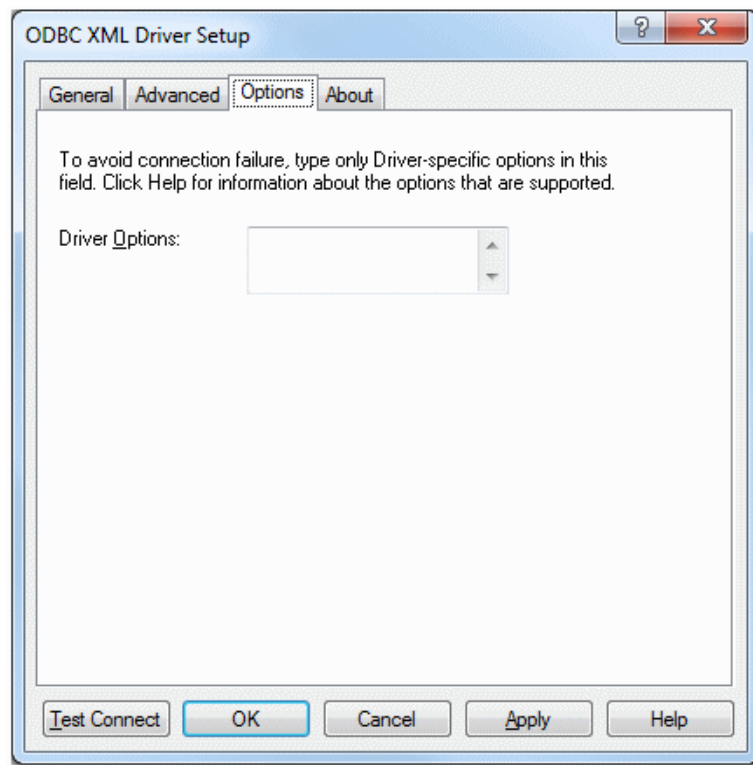
Disabled

Show Virtual Schemas Only

0 - Ignore Errors

Disabled

- 10 Optionally, click the **Options** tab to specify data source connection values.



**Driver Options:** Type configuration options specific to the XML driver.

Connection Options: Options	Default
<a href="#">Driver Options (see page 643)</a>	Disabled

WARNING: The properties you set in the Options tab override other properties for this session only and can adversely affect the operation of the XML driver. Use only authorized entries. For information about authorized entries for the Options tab, contact Progress DataDirect customer support.

- 11 At any point during the configuration process, you can click **Test Connect** to attempt to connect to the data source using the connection options specified in the driver Setup dialog box. A logon dialog box appears (see ["Using a Logon Dialog Box" on page 641](#) for details). 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 incorrect environment or connection value, it displays an appropriate error message. Click **OK**.
- 12 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.



## Using a Connection String

If you want to use a connection string for connecting to a database, or if your application requires it, you must specify either a DSN (data source name), a File DSN, or a DSN-less connection in the string. The difference is whether you use the DSN=, FILEDSN=, or the DRIVER= keyword in the connection string, as described in the ODBC specification. A DSN or FILEDSN connection string tells the driver where to find the default connection information. Optionally, you may specify *attribute=value* pairs in the connection string to override the default values stored in the data source.

The DSN connection string has the form:

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

The FILEDSN connection string has the form:

```
FILEDSN=filename.dsn[;attribute=value[;attribute=value]...]
```

The DSN-less connection string specifies a driver instead of a data source. All connection information must be entered in the connection string because the information is not stored in a data source.

The DSN-less connection string has the form:

```
DRIVER=[{]driver_name[}][;attribute=value[;attribute=value]...]
```

[Table 19-2](#) and [Table 19-3](#) give the names and descriptions of the attributes, as well as the initial default value when the driver is first installed.

An example of a DSN connection string with overriding attribute values for XML is:

```
DSN=XML FILES;LOC1.Create Type=ADO25;Logging=1
```

A FILEDSN connection string is similar except for the initial keyword:

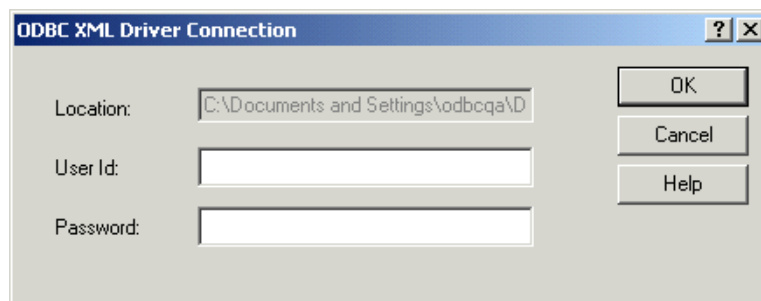
```
FILEDSN=XML.dsn;LOC1.Create Type=ADO25;Logging=1
```

A DSN-less connection string must provide all necessary connection information:

```
DRIVER=DataDirect 7.1 XML;LOC1={DataDirect Closed XML ADO Provider}
```

## Using a Logon Dialog Box

Some ODBC applications display a Logon dialog box when you are connecting to a data source. For XML, the dialog box is as follows:



This dialog box appears for each password-protected location that you have defined for the data source.

In this dialog box, provide the following information:

- 1 Type your user ID and password in the appropriate fields for the Location that appears in the Location field.
- 2 Click **OK** to connect to the data source.

---

# Connection Option Descriptions

The following connection option descriptions are listed alphabetically by the GUI name that appears on the driver Setup dialog. The connection string attribute name is listed immediately underneath the GUI name. For example:

## Report Codepage Conversion Errors

Attribute    ReportCodepageConversionErrors

In most cases, the GUI name and the attribute name are the same; however, some exceptions exist. If you need to look up an option by its connection string attribute name, please refer to the alphabetical table of connection string attribute names.

Also, a few connection string attributes, for example, Password, do not have equivalent options that appear on the GUI. They are in the list of descriptions alphabetically by their attribute names.

NOTE: XML driver connection string attributes do not use short name equivalents.

# Driver Setup Dialog Box Descriptions

Table 19-2 lists the connection string attributes associated with General, Advanced, and Options tabs of the XML driver Setup dialog box. The descriptions themselves are listed below the table.

---

**Table 19-2. XML Attribute Names**

---

Attribute	Default
DataSourceName	None
Description	None
InternationalSort	Disabled
Location	None
Logging	Disabled
ReadOnly	Disabled
ReportCodepageConversionErrors	0 (Ignore Errors)

**Table 19-2. XML Attribute Names** (cont.)

Attribute	Default
<a href="#">ShowManufacturedSchemas</a>	0 (Disabled)
<a href="#">ShowVirtualSchemas</a>	1 (Enabled)

## Data Source Name

Attribute	DataSourceName
Description	The name of a data source in your Windows Registry or odbc.ini file.
Valid Values	<i>string</i> where <i>string</i> is the name of a data source.
Default	None
GUI Tab	<a href="#">General tab</a> on page 636

## Description

Attribute	Description
Description	An optional long description of a data source. This description is not used as a runtime connection attribute, but does appear in the ODBC.INI section of the Registry and in the odbc.ini file.
Valid Values	<i>string</i> where <i>string</i> is a description of a data source.
Default	None
GUI Tab	<a href="#">General tab</a> on page 636

## Driver Options

Attribute	n/a
Description	Type configuration options specific to the XML driver.
Valid Values	WARNING: The properties you set in the Options tab override other properties for this session only and can adversely affect the operation of the XML driver. Use only authorized entries. For information about authorized entries for the Options tab, contact Progress DataDirect customer support.
Default	None
GUI Tab	<a href="#">Options tab</a> on page 640

## International Sort Order

Attribute	InternationalSort
Description	Uses international sort order as defined by your operating system when you issue a Select statement with an Order By clause.
Valid Values	0   1  If set to 1 (Enabled), this order is always alphabetic, regardless of case; the letters are sorted as "A, b, C." Refer to your operating system documentation concerning the sorting of accented characters.  If set to 0 (Disabled), ASCII sort order is used. This order sorts items alphabetically with uppercase letters preceding lowercase letters. For example, "A, b, C" is sorted as "A, C, b."
Default	0 (Disabled)
GUI Tab	<a href="#">Advanced tab</a> on page 639

## Location Names

Attribute	Location
Description	A display of all existing location names defined for the data source you are configuring.
Valid Values	<i>string</i>  where <i>string</i> is the name of a location.  The location names listed in the text box are used for connections according to the order that they are displayed. If you want to change the order or precedence, use the Up and Down buttons.
Default	None
GUI Tab	<a href="#">General tab</a> on page 636

## Logging

Attribute	Logging
Description	Creates a log file that logs the SQL execution plan. A value of 0 means no logging is performed.
Valid Values	0   1  If set to 1 (Enabled), a log file is created in the current directory. The default log file name is \Integrator.txt.  If set to 0 (Disabled), no logging is performed.
Default	0 (Disabled)
GUI Tab	<a href="#">Advanced tab</a> on page 639

## Read Only

Attribute	ReadOnly
Description	Controls whether the driver opens files for Read-Write access or Read-Only access
Valid Values	0   1
	<p>If set to 1 (Enabled), the driver opens XML files for Read-Only access. In this case, the XML driver opens XML files with a Shared Read lock. This allows other connections and applications to read the same XML file that the XML driver has open; however, they cannot write to the XML file.</p> <p>If set to 0 (Disabled), the XML driver opens XML files for Read-Write access. Opening an XML file for Read-Write access places an exclusive lock on the file. No other connections or applications can open the XML file while the driver has the file open.</p>
Default	0 (Disabled)
GUI Tab	<a href="#">Advanced tab</a> on page 639

## Report Codepage Conversion Errors

Attribute	ReportCodepageConversionErrors
Description	<p>Specifies how the driver handles code page conversion errors that occur when a character cannot be converted from one character set to another.</p> <p>An error message or warning can occur if an ODBC call causes a conversion error, or if an error occurs during code page conversions to and from the database or to and from the application. The error or warning generated is <code>Code page conversion error encountered</code>. In the case of parameter data conversion errors, the driver adds the following sentence: <code>Error in parameter x</code>, where <code>x</code> is the parameter number. The standard rules for returning specific row and column errors for bulk operations apply.</p>
Valid Values	0   1   2
	<p>If set to 0 (Ignore Errors), the driver substitutes 0x1A for each character that cannot be converted and does not return a warning or error.</p> <p>If set to 1 (Return Error), the driver returns an error instead of substituting 0x1A for unconverted characters.</p> <p>If set to 2 (Return Warning), the driver substitutes 0x1A for each character that cannot be converted and returns a warning.</p>
Default	0 (Ignore Errors)
GUI Tab	<a href="#">Advanced tab</a> on page 639

## Schema Mode

Attribute	n/a
Description	Specifies whether to show virtual schemas, manufactured schemas, or both.

Valid Values	Choose one of the following options: <ul style="list-style-type: none"><li>■ Show Virtual Schemas Only. This option returns "XML" in the Schema Name column when a SQLTables or SQLColumns operation is performed when connected to a data source. See <a href="#">Show Virtual Schemas</a>.</li><li>■ Show Manufactured Schemas Only. This option returns the manufactured schema names in the Schema Name column when a SQLTables or SQLColumns operation is performed when connected to a data source. See <a href="#">Show Manufactured Schemas</a>.</li><li>■ Show Both Virtual and Manufactured Schemas. This option returns both virtual and manufactured schema names when a SQLTables or SQLColumns operation is performed when connected to a data source.</li></ul>
Default	Show Virtual Schemas Only
GUI Tab	<a href="#">Advanced tab</a> on page 639

Show Manufactured Schemas

Attribute	ShowManufacturedSchemas										
Description	Returns the manufactured schema names in the Schema Name column when a SQLTables or SQLColumns operation is performed when connected to a data source. The Location names you define for a data source are manufactured into a schema name by adding a # symbol after the Location names. For example: <table><thead><tr><th>Schema Name</th><th>Table Name</th></tr></thead><tbody><tr><td>LOC1#</td><td>TAB1A</td></tr><tr><td>LOC1#</td><td>TAB1B</td></tr><tr><td>LOC2#</td><td>TAB2A</td></tr><tr><td>LOC2#</td><td>TAB2B</td></tr></tbody></table>	Schema Name	Table Name	LOC1#	TAB1A	LOC1#	TAB1B	LOC2#	TAB2A	LOC2#	TAB2B
Schema Name	Table Name										
LOC1#	TAB1A										
LOC1#	TAB1B										
LOC2#	TAB2A										
LOC2#	TAB2B										
Valid Values	0   1 <p>If set to 1 (Enabled), manufactured schema names are returned.</p> <p>If set to 0 (Disabled), manufactured schema names are not returned.</p> <p>To return both manufactured and virtual schema names, set this option to 1 (Enabled) and the Show Virtual Schemas option to 1 (Enabled).</p>										
Default	0 (Disabled)										
GUI Tab	<a href="#">Advanced tab</a> on page 639										

## Show Virtual Schemas

Attribute	ShowVirtualSchemas										
Description	Returns "XML" in the Schema Name column when a SQLTables or SQLColumns operation is performed when connected to a data source. For example:										
	<table> <tr> <th>Schema Name</th><th>Table Name</th></tr> <tr> <td>XML</td><td>TAB1A</td></tr> <tr> <td>XML</td><td>TAB1B</td></tr> <tr> <td>XML</td><td>TAB2A</td></tr> <tr> <td>XML</td><td>TAB2B</td></tr> </table>	Schema Name	Table Name	XML	TAB1A	XML	TAB1B	XML	TAB2A	XML	TAB2B
Schema Name	Table Name										
XML	TAB1A										
XML	TAB1B										
XML	TAB2A										
XML	TAB2B										
Valid Values	0   1										
	If set to 1 (Enabled), virtual schema names are returned.										
	If set to 0 (Disabled), virtual schema names are not returned.										
	To return both virtual and manufactured schema names, set this option to 1 (Enabled) and the Show Manufactured Schemas option to 1 (Enabled).										
Default	1 (Enabled)										
GUI Tab	<a href="#">Advanced tab</a> on page 639										

## Configure Location Dialog Box Descriptions

[Table 19-3](#) lists the connection string attributes associated with General and Advanced tabs of the XML driver Configure Location dialog box. The descriptions themselves are listed below the table. See ["Defining Locations" on page 633](#) for an explanation of locations.

NOTE: XML driver connection string attributes do not use short name equivalents.

The names of all connection options in this section are preceded by *location\_name*, where *location\_name* represents the name of a specific location that you have defined, for example, LOC1. See the description of the [Location Name](#) option for details.

---

**Table 19-3. XML Configure Location Attribute Names**

---

Attribute	Default
<a href="#">location_name</a>	None
<a href="#">location_name.Catalog Type Hint</a>	Folder
<a href="#">location_name.Create Type</a>	DataDirect
<a href="#">location_name.Delete Schema</a>	0 (Disabled)
<a href="#">location_name.Flush Every Change</a>	1 (Enabled)
<a href="#">location_name.Initial Catalog</a>	None
<a href="#">location_name.Password</a>	None
<a href="#">location_name.Require Passwd</a>	0 (Disabled)

**Table 19-3. XML Configure Location Attribute Names** (cont.)

Attribute	Default
<a href="#">location_name.Resolve External</a>	0 (Disabled)
<a href="#">location_name.Row Hint</a>	None
<a href="#">location_name.Scan Rows</a>	0
<a href="#">location_name.Table Hint</a>	None
<a href="#">location_name.User ID</a>	None
<a href="#">location_name.Validate Schema</a>	0 (Disabled)

## Delete Linked Schema

Attribute	<i>location_name.Delete Schema</i>
Description	Specifies whether an externally-linked schema file is deleted when a table is deleted. This option is valid only for Folder location types. The XML document for the table contains a link to this external schema file. By default, this check box is not selected.
Valid Values	0   1  If set to 1 (Enabled), the externally-linked schema file is deleted when the table is deleted. If multiple XML documents are linked to the same schema file, the schema file is not deleted when a table is deleted.  If set to 0 (Disabled), the externally-linked schema file is not deleted when the table is deleted.
Default	0 (Disabled)
GUI Tab	<a href="#">Configure Location Advanced tab</a> on page 638

## Flush Every Change

Attribute	<i>location_name.Flush Every Change</i>
Description	Writes the data document to disk after every insert, update, or delete operation. This option is valid only for Folder location types.
Valid Values	0   1  If set to 1 (Enabled), the driver writes the data document to disk after every change.  If set to 0 (Disabled), the driver does not write the data document to disk after every change. Disabling this option can improve performance.
Default	1 (Enabled)
GUI Tab	<a href="#">Configure Location Advanced tab</a> on page 638



## Location

Attribute	<i>location_name</i> .Initial Catalog
Description	The full path name to the location you are defining.
Valid Values	<i>location_directory</i>  where <i>location_directory</i> is the full path name of the directory in which the data files are stored. For example:  LOC1.Initial Catalog=C:\Documents\filesml
Default	None
GUI Tab	<a href="#">Configure Location General tab</a> on page 637

## Location Name

Attribute	<i>location_name</i>
Description	A unique name for the location you are defining, for example, LOC1.
Valid Values	<i>location_name</i> ={DataDirect Closed XML ADO Provider}  where <i>location_name</i> is the unique name of the location you are defining. For example, if you choose the location name LOC1, then:  LOC1={DataDirect Closed XML ADO Provider}
Default	None
GUI Tab	<a href="#">Configure Location General tab</a> on page 637

## Location Type

Attribute	<i>location_name</i> .Catalog Type Hint
Description	Specifies the type of location you are defining for the connection.
Valid Values	Folder   XML Document   HTML Document  For example:  LOC1.Catalog Type Hint=XML Document  See <a href="#">"Defining Locations" on page 633</a> for the definition of each type.
Default	Folder
GUI Tab	<a href="#">Configure Location General tab</a> on page 637

## Max Rows to Scan

Attribute	<i>location_name</i> .Scan Rows
Description	An integer that represents the maximum number of rows to scan when the XML driver is determining the data type of each column. This option is valid only for XML Document location types.

Valid Values 0 |  $x$

where  $x$  is the number of rows to scan.

If set to  $x$ , the driver scans a maximum of  $x$  rows in the table. During the scan, the driver inspects each column value in the row of a table and adjusts the data type determination for each column based on the corresponding value. The more sample column values it encounters, the more accurate the determination.

If set to 0, the driver scans all rows in the table. Disabling this option can improve performance because limiting the number of rows can reduce the amount of time it takes to determine the column information on very large documents. Because less information is available, however, the determination of the data types can be incorrect.

Default 0

GUI Tab [Configure Location Advanced tab](#) on page 638

## Password

Attribute *location\_name*.Password

Description The password used to establish a connection to the location specified by *location\_name*. A password is required only if the location to which you are connecting is password-protected.

This option is not available unless the Require User ID/Password option is enabled.

Valid Values *pwd*

where *pwd* is a valid password.

WARNING: The encrypted password is stored in the Windows Registry.

Default None

GUI Tab [Configure Location Advanced tab](#) on page 638

## Require User ID/Password

Attribute *location\_name*.Require Passwd

Description Specifies whether a User ID and password are required to establish a connection to the location you are defining.

Valid Values 0 | 1

If set to 1 (Enabled), a User ID and password are required to establish a connection to the location. You must enable this option if the location you are defining is password-protected; otherwise, the connection will fail. Enabling this option causes a [Logon dialog box](#) to appear when connecting with the driver.

If set to 0 (Disabled), no user ID and password are required to establish a connection to the location.

Default 0 (Disabled)

GUI Tab [Configure Location Advanced tab](#) on page 638

## Resolve External References

Attribute	<i>location_name</i> .Resolve External
Description	Determines whether external references such as DTDs, Schemas, Entities, and Notations are resolved for the XML documents contained within the location specified by <i>location_name</i> .
Valid Values	0   1  If set to 1 (Enabled), the documents are not processed if the XML parser cannot locate the external references.  If set to 0 (Disabled), the document is processed, even if the XML parser cannot locate the external references.
Default	0 (Disabled)
GUI Tab	<a href="#">Configure Location Advanced tab</a> on page 638

## Row Hint

Attribute	<i>location_name</i> .Row Hint
Description	A string that specifies an Extensible Stylesheet Language (XSL) pattern to identify the nodes that make up the rows in the rowset of a tabular-formatted XML document contained within the location specified by <i>location_name</i> . See <a href="#">"Using Hints for Tabular-Formatted XML Documents" on page 653</a> for details.  This option is valid only for Folder and HTML Document location types.
Valid Values	<i>row_hint</i>  where <i>row_hint</i> is an XSL pattern.
Default	None
GUI Tab	<a href="#">Configure Location Advanced tab</a> on page 638

## Table Creation

Attribute	<i>location_name</i> .Create Type
Description	Determines the style of XML that is generated when a new table is created. This option is valid only for Folder location types.
Valid Values	IE5DataIsland   ADO25   DataDirect  <ul style="list-style-type: none"> <li>■ Data Island Format (IE5DataIsland): New tables are created with the Internet Explorer 5 Data Island XML style.</li> <li>■ ADO Format (ADO25): New tables are created with the ADO 2.5 XML style.</li> <li>■ DataDirect Format (DataDirect): New tables are created with the DataDirect format. This format conforms to the W3C recommendation for XML schema, working draft April 07, 2000.</li> </ul> <p>See <a href="#">"Common Tabular Formats for XML Documents" on page 630</a> for a description of each of these formats.</p>

Default	DataDirect
GUI Tab	<a href="#">Configure Location Advanced tab</a> on page 638

## Table Hint

Attribute	<i>location_name</i> .Table Hint
Description	<p>A string that specifies an Extensible Stylesheet Language (XSL) pattern to identify the table or rowset nodes in a tabular-formatted XML document contained within the location specified by <i>location_name</i>. See <a href="#">"Using Hints for Tabular-Formatted XML Documents" on page 653</a> for details.</p> <p>This option is valid only for Folder and HTML Document location types.</p>
Valid Values	<p><i>table_hint</i></p> <p>where <i>table_hint</i> is an XSL pattern.</p>
Default	None
GUI Tab	<a href="#">Configure Location Advanced tab</a> on page 638

## User ID

Attribute	<i>location_name</i> .User ID
Description	<p>The User ID (user name) used to establish a connection to the location specified by <i>location_name</i>. A password is required only if the location to which you are connecting is password-protected.</p> <p>This option is not available unless the Require User ID/Password option is enabled.</p>
Valid Values	<p><i>userid</i></p> <p>where <i>userid</i> is a valid user name.</p>
Default	None
GUI Tab	<a href="#">Configure Location Advanced tab</a> on page 638

## Validate Schema

Attribute	<i>location_name</i> .Validate Schema
Description	Determines whether the XML documents contained within the location specified by <i>location_name</i> are validated against their schema.
Valid Values	<p>0   1</p> <p>If set to 1 (Enabled), the XML documents are validated against their schema. This allows a well-formed XML document to be processed, even if the document is not valid.</p> <p>If set to 0 (Disabled), the XML documents are not validated against their schema.</p>
Default	0 (Disabled)
GUI Tab	<a href="#">Configure Location Advanced tab</a> on page 638

## Using Hints for Tabular-Formatted XML Documents

The XML driver supports table and row hints. You can specify a table hint, a row hint, or both, when configuring an XML data source or using a connection string.

Table hints should be specified so that they resolve to a single node. If a table hint resolves to a set of nodes, the first node in the set is used as the table node. The context of the table hint is always the root node of the XML document.

Row hints define the "row" element and specify whether the rowset is element-based or attribute-based. If a table hint is supplied, the context of the row node is the node to which the table hint resolves; otherwise, the context is the root node of the XML document. The column mode identifier specifies whether the columns of a row are child nodes or attributes of the row node.

When working with hints, keep in mind that the XML driver assumes that the row nodes are the immediate children of the table node.

- If only a table hint is specified, the row nodes are the children of the node to which the hint resolves. It is assumed that all of the child nodes have the same name.
- If only a row hint is specified, the table node is the parent of the node to which the hint resolves. If the row hint resolves to a set of nodes, the nodes in that set must all have the same parent.
- If both a table hint and a row hint are specified, the row hint is taken to be relative to the node to which the table hint resolves.

The column mode identifier has the format:

```
\column mode
```

where *mode* can be one of the following options:

- child: The columns are child nodes of the row node.
- attr: The columns are attributes of the row node.

In the following examples, the columns are the children of the row nodes.

### Example 1

Table Hint:

Row Hint: //Item

The row nodes are the nodes named Item. The table node is the parent of the row nodes. Use this form only when all of the Item nodes reside under one parent.

If some Item nodes have different parents, use a table hint or a more specific row hint to select the set of Item nodes.

**Example 2**

Table Hint:

Row Hint: /Bookstore/Books/Item

The row nodes are the nodes named Item. The table node is Books, which is a child of the Bookstore node.

**Example 3**

Table Hint: /Bookstore/Books

Row Hint:

The table node is Books, which is a child of the Bookstore node. The row nodes are the children of the Books node. It is assumed that all of the child nodes under the Books nodes have the same name. If the child nodes do not all have the same name, the name of the first child node encountered is used as the row node name. In that case, it would be better to specify both a table and row hint.

**Example 4**

Table Hint: /Bookstore [@location = "Raleigh"]/Books

Row Hint: ./Item

The table node is Books, which is a child of the Bookstore node. Bookstore has a "location" attribute with the value Raleigh. The row nodes are the Item nodes that are children of the Books node.

**Column Mode Identifier**

The following examples illustrate the use of the optional column mode identifier.

**Example 5**

Table Hint:

Row Hint: //Item \column attr

The row nodes are named Item. The table node is the parent of the row nodes. The columns are attributes of the row node.

**Example 6**

Table Hint:

Row Hint: //Item \column child

The row nodes are the nodes named Item. The table node is the parent of the row nodes. The columns are attributes of the row node.

# Data Types

This section provides three tables that show how the data types for each supported tabular-formatted XML document map to the standard ODBC data types, as follows:

- [Table 19-4 "Data Islands Data Types" on page 655](#)
- [Table 19-5 "ADO 2.5 Persisted Files Data Types" on page 655](#)
- [Table 19-6 "DataDirect Format Data Types" on page 656](#)

**Table 19-4. Data Islands Data Types**

Data Islands	Internal XML Name	ODBC
binhex	bin.hex	SQL_LONGVARBINARY
boolean	boolean	SQL_BIT
currency	fixed.14.4	SQL_DECIMAL
date	date	SQL_TYPE_DATE
dateTime	dateTime	SQL_TYPE_TIMESTAMP
float	float	SQL_DOUBLE
i1	i1	SQL_TINYINT SIGNED
i2	i2	SQL_SMALLINT SIGNED
i4	i4	SQL_INTEGER SIGNED
int	int	SQL_INTEGER SIGNED
number	number	SQL_DOUBLE
r4	r4	SQL_REAL
r8	r8	SQL_DOUBLE
singleChar	singleChar	SQL_SMALLINT
string	string	SQL_WLONGVARCHAR
time	time	SQL_TYPE_TIME
ui1	ui1	SQL_TINYINT UNSIGNED
ui2	ui2	SQL_SMALLINT UNSIGNED
ui4	ui4	SQL_INTEGER UNSIGNED

**Table 19-5. ADO 2.5 Persisted Files Data Types**

ADO 2.5 Persisted Files	Internal XML Name	ODBC
binhex	bin.hex	SQL_LONGVARBINARY
boolean	boolean	SQL_BIT
currency	fixed.14.4	SQL_DECIMAL
date	date	SQL_TYPE_DATE
dateTime	dateTime	SQL_TYPE_TIMESTAMP
float	float	SQL_DOUBLE
i1	i1	SQL_TINYINT SIGNED

**Table 19-5. ADO 2.5 Persisted Files Data Types** (cont.)

<b>ADO 2.5 Persisted Files</b>	<b>Internal XML Name</b>	<b>ODBC</b>
i2	i2	SQL_SMALLINT SIGNED
i4	i4	SQL_INTEGER SIGNED
i8	i8	SQL_BIGINT SIGNED
int	int	SQL_INTEGER UNSIGNED
number	number	SQL_DOUBLE
r4	r4	SQL_REAL
r8	r8	SQL_DOUBLE
singleChar	singleChar	SQL_SMALLINT SIGNED
time	time	SQL_TYPE_TIME
ui1	ui1	SQL_TINYINT UNSIGNED
ui2	ui2	SQL_SMALLINT UNSIGNED
ui4	ui4	SQL_INTEGER UNSIGNED
ui8	ui8	SQL_BIGINT UNSIGNED
wchar	string	SQL_CHAR
wchar	string	SQL_WCHAR
wlvarchar	string	SQL_WLONGVARBINARY
wvarchar	string	SQL_WVARCHAR

**Table 19-6. DataDirect Format Data Types**

<b>DataDirect</b>	<b>Internal XML Name</b>	<b>ODBC</b>
binary	binary	SQL_BINARY
boolean	boolean	SQL_BIT
byte	byte	SQL_TINYINT SIGNED
date	date	SQL_TYPE_DATE
decimal	decimal	SQL_NUMERIC
double	double	SQL_DOUBLE
float	float	SQL_REAL
int	int	SQL_INTEGER UNSIGNED
long	long	SQL_BIGINT SIGNED
lvarbinary	binary	SQL_LONGVARBINARY
short	short	SQL_SMALLINT SIGNED
time	time	SQL_TYPE_TIME
timeInstant	timeInstant	SQL_TYPE_TIMESTAMP
unsignedByte	unsignedByte	SQL_TINYINT UNSIGNED
unsignedInt	unsignedInt	SQL_INTEGER UNSIGNED
unsignedLong	unsignedLong	SQL_BIGINT UNSIGNED



**Table 19-6. DataDirect Format Data Types** (cont.)

<b>DataDirect</b>	<b>Internal XML Name</b>	<b>ODBC</b>
unsignedShort	unsignedShort	SQL_SMALLINT UNSIGNED
varbinary	binary	SQL_VARBINARY
wchar	string	SQL_CHAR
wchar	string	SQL_WCHAR
wlvarchar	string	SQL_WLONGVARBINARY
wvarchar	string	SQL_WVARCHAR

See ["Retrieving Data Type Information" on page 59](#) for information about retrieving data types.

## Unicode Support

The driver supports the Unicode ODBC W (Wide) function calls, such as SQLConnectW. This allows the Driver Manager to transmit these calls directly to the driver. Otherwise, the Driver Manager would incur the additional overhead of converting the W calls to ANSI function calls, and vice versa.

See ["UTF-16 Applications on UNIX and Linux" on page 125](#) for related details. Also, refer to [Chapter 4 "Internationalization, Localization, and Unicode"](#) in the *DataDirect Connect Series for ODBC Reference* for a more detailed explanation of Unicode.

## Persisting a Result Set as an XML Data File

The driver allows you to persist a result as an XML data file with embedded schema. See ["Persisting a Result Set as an XML Data File" on page 60](#) for details about implementation.

## ODBC Conformance Level

The driver is Level 1 compliant, that is, it supports all ODBC Core and Level 1 functions.

In addition, the driver supports SQLSetPos.

Refer to [Chapter 2 "ODBC API and Scalar Functions"](#) in the *DataDirect Connect Series for ODBC Reference* for a list of the API functions supported by the XML driver.

---

# Number of Connections and Statements Supported

There is no limit to the number of connections and statements supported.

---

## SQL Support

This section provides information about the SQL statements that the XML driver processes, and about SQL standards and conventions that the driver supports:

- ["SQL Statements" on page 658](#)
- ["Extensions to SQL Standards" on page 658](#)
- ["Grammar Token Definitions" on page 659](#)

### SQL Statements

The SQL Engine included with the XML driver supports the following SQL statements:

- Select
- Create and Drop Table
- Insert
- Update
- Delete

NOTE: See the table at the beginning of this chapter for the SQL statements that the XML driver supports for the different types of supported file formats.

### Extensions to SQL Standards

The XML driver uses SQL grammar that is compliant with entry level ANSI SQL-92. [Table 19-7](#) summarizes significant extensions to the grammar.

---

**Table 19-7. SQL Extensions**

---

Entry Level ANSI SQL-92 Extension	Relevant Standard or Convention
Aliasing table references	Intermediate level ANSI SQL-92
ANSI date, time, and timestamp literals	Intermediate level ANSI SQL-92
Dynamic parameter specification	Full level ANSI SQL-92
GUID literals	COM
Hex string literals	Full level ANSI SQL-92
Left Outer Joins	Intermediate level ANSI SQL-92
ODBC escape support	ODBC 3.0
Scalar functions	ODBC 3.0

---

## Grammar Token Definitions

The tokens used in the XML driver SQL grammar are defined in the following sections:

- ["Regular Identifiers" on page 659](#)
- ["Delimited Identifiers" on page 659](#)
- ["Integer Numbers" on page 659](#)
- ["Real Numbers" on page 660](#)
- ["Character String Literals" on page 660](#)
- ["GUID Literals" on page 660](#)
- ["Hex Literals" on page 660](#)
- ["Time and Date Literals" on page 660](#)
- ["SQL Operators and Symbols" on page 661](#)
- ["Keywords for the XML Driver" on page 661](#)
- ["SQL Comments" on page 665](#)

### Regular Identifiers

A regular identifier must begin with a letter and may not exceed 128 characters. In addition, all ASCII characters are converted to uppercase.

The following are examples of regular identifiers:

- FOO
- COLUMN\_NAME
- SCHEMA#NAME
- Col3 (legal, but converted to COL3)

### Delimited Identifiers

Delimited identifiers may not exceed 128 characters. A double quotation character can be embedded within the string by specifying two consecutive double quotation mark characters. A delimited identifier can span multiple lines. The body of a delimited identifier can contain any character except the newline character.

The following examples show delimited identifiers:

- "\$ % ^ ( \$"
- "This is a delimited variable name"

### Integer Numbers

Examples of integer numbers are:

- 5
- 1004

## Real Numbers

Examples of real numbers are:

- .10
- 12.01
- 10.
- .01e-10
- 12E+10
- 12.01e2
- 12.01e-10
- 12.e-10

## Character String Literals

Character string literals are delimited with single quotation mark characters. A single quotation mark character can be embedded within the string by specifying two consecutive single quotation mark characters. A character string literal can span multiple lines.

Examples are:

- '\$%^('\$'
- 'This is a character string literal'

## GUID Literals

A GUID uses the following format, where *x* is a hexadecimal digit:

```
xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
```

## Hex Literals

Hex literal values are introduced with an uppercase *x* followed by a single quoted string of hexadecimal characters.

Examples are:

- X'39FA'
- X'B0F00D'

## Time and Date Literals

Date, time, and timestamp literals are date, time, and timestamp values surrounded by a standard prefix and suffix. Date literals are specified in a *YYYY-MM-DD* format. Time literals are specified in an *HH:MM:SS* format with an optional fraction component. Timestamp literals are a concatenation of date and time values.

Examples for ODBC and SQL syntax are shown in the following table.

**Table 19-8. Time and Date Literals**

<b>Literal Type</b>	<b>ODBC Syntax</b>	<b>ANSI SQL-92 Syntax</b>
Date Literal	{d '1999-09-19'}	date '1999-09-19'
Time Literal	{t '11:11:11.225'}	time '11:11:11.225'
Timestamp Literal	{ts '1999-09-19 11:11:11.225'}	timestamp '1999-09-19 11:11:11.225'
Timestamp Literal	{ts '1999-09-19'}	timestamp '1999-09-19'

NOTE: ODBC 1.x style ODBC escape sequences such as the following are not supported:

```
--(*VENDOR(Microsoft), PRODUCT(ODBC) ...*)--
```

## SQL Operators and Symbols

**Table 19-9. SQL Operators and Symbols**

<b>Symbol</b>	<b>Description</b>	<b>Symbol</b>	<b>Description</b>
':'	Colon	'<'	Less than operator
';'	Semicolon	')	Right parenthesis
':'	Period	'='	Equal operator
','	Comma	'+'	Plus operator
'<>'	Not equal operator	'-'	Minus operator
'<='	Less than or equal operator	'*'	Multiply operator
'>='	Greater than or equal operator	'/'	Divide operator
'>'	Greater than operator	'?'	Dynamic parameter
'('	Left parenthesis		

## Keywords for the XML Driver

A keyword may not be used as a regular identifier. For example, the following statement would generate a syntax error because INDICATOR is a keyword:

```
SELECT INDICATOR FROM T1
```

You can, however, enclose a keyword in double quotation marks to form a delimited identifier. For example, the following statement is valid:

```
SELECT "INDICATOR" FROM T1
```

Table 19-10 lists all of the keywords that are reserved for use in SQL statements or designated as potential future reserved words.

---

**Table 19-10. Reserved Keywords**

---

ABSOLUTE	ACTION	ADD
AFTER	ALIAS	ALL
ALLOCATE	ALTER	AND
ANY	ARE	AS
ASC	ASSERTION	ASYNC
AT	AUTHORIZATION	AVG
BEFORE	BEGIN	BETWEEN
BIT	BIT_LENGTH	BOOLEAN
BOTH	BREADTH	BY
CALL	CASCADE	CASCADE
CASE	CAST	CATALOG
CHAR	CHAR_LENGTH	CHARACTER
CHARACTER_LENGTH	CHECK	CLOSE
COALESCE	COLLATE	COLLATION
COLUMN	COLUMNS	COMMIT
COMPLETION	CONCAT	CONNECT
CONNECTION	CONSTRAINT	CONSTRAINTS
CONTINUE	CONVERT	CORRESPONDING
COUNT	CREATE	CROSS
CURDATE	CURRENT	CURRENT_DATE
CURRENT_TIME	CURRENT_TIMESTAMP	CURRENT_USER
CURSOR	CURTIME	CYCLE
DATA	DATE	DAY
DAYOFMONTH	DAYOFWEEK	DEALLOCATE
DEC	DECIMAL	DECLARE
DEFAULT	DEFERRABLE	DEFERRED
DELETE	DEPTH	DESC
DESCRIBE	DESCRIPTOR	DIAGNOSTICS
DICTIONARY	DISCONNECT	DISTINCT
DOMAIN	DOUBLE	DROP
EACH	ELSE	ELSEIF
END	END_EXEC	EQUALS
ESCAPE	EXCEPT	EXCEPTION
EXEC	EXECUTE	EXISTS
EXTERNAL	EXTRACT	FALSE
FETCH	FIRST	FLOAT

**Table 19-10. Reserved Keywords** (cont.)

FLOOR	FOR	FOREIGN
FOUND	FROM	FULL
GENERAL	GET	GLOBAL
GO	GOTO	GRANT
GROUP	HAVING	HOURL
IDENTIFY	IF	IFNULL
IGNORE	IMMEDIATE	IN
INDEX	INFO	INDICATOR
INITIALLY	INNER	INPUT
INSENSITIVE	INSERT	INT
INTEGER	INTERSECT	INTERVAL
INTO	IS	ISOLATION
JOIN	KEY	LANGUAGE
LAST	LCASE	LEADING
LEAVE	LEFT	LENGTH
LESS	LEVEL	LIKE
LIMIT	LOCAL	LOOP
LOWER	LTRIM	MATCH
MAX	MIN	MINUTE
MOD	MODIFY	MODULE
MONTH	NAMES	NATIONAL
NATURAL	NCHAR	NEW
NEXT	NO	NONE
NOT	NOW	NULL
NULLIF	NUMERIC	OBJECT
OCTET_LENGTH	OF	OFF
OID	OLD	ON
ONLY	OPEN	OPERATION
OPERATORS	OPTION	OR
ORDER	OTHERS	OUTER
OUTPUT	OVERLAPS	PAD
PARAMETERS	PARTIAL	PENDANT
POSITION	POWER	PRECISION
PREORDER	PREPARE	PRESERVE
PRIMARY	PRIOR	PRIVATE
PRIVILEGES	PROCEDURE	PROTECTED
PUBLIC	RCASE	READ
REAL	RECURSIVE	REF
REFERENCES	REFERENCING	RELATIVE

**Table 19-10. Reserved Keywords** (cont.)

REMOVE	REPLACE	RESIGNAL
RESTRICT	RETURN	RETURNS
REVOKE	RIGHT	ROLE
ROLLBACK	ROUND	ROUTINE
ROW	ROWS	RTRIM
SAVEPOINT	SCHEMA	SCROLL
SEARCH	SECOND	SECTION
SELECT	SENSITIVE	SEQUENCE
SESSION	SESSION_USER	SET
SIGNAL	SIMILAR	SIZE
SMALLINT	SOME	SPACE
SQL	SQLCODE	SQLERROR
SQLEXCEPTION	SQLSTATE	SQLWARNING
STRUCTURE	SUBSTRING	SUM
SYSTEM_USER	TABLE	TEMPORARY
TEST	THEN	THERE
TIME	TIMESTAMP	TIMEZONE_HOUR
TIMEZONE_MINUTE	TO	TRAILING
TRANSACTION	TRANSLATE	TRANSLATION
TRIGGER	TRIM	TRUE
TYPE	UCASE	UNDER
UNION	UNIQUE	UNKNOWN
UPDATE	UPPER	USAGE
USER	USING	VALUE
VALUES	VARCHAR	VARIABLE
VARYING	VIEW	VIRTUAL
VISIBLE	WAIT	WHEN
WHENEVER	WHERE	WHILE
WITH	WITHOUT	WORK
WRITE	YEAR	ZONE



## SQL Comments

ANSI SQL-92 standard comments (--) and C++ standard comments (/\*...\*/, //) are supported. Comments can be nested.

For example, in the following query columns col2, col3, and col4 are ignored:

```
SELECT col1  /* col1 comment */
/*
    col2,-- col2 comment
    col3,// col3 comment
    col4,/* col4 comment */
*/
FROM t1
```



## Part 4: The Connect XE Drivers

This part describes the Progress DataDirect Connect XE drivers. It contains the following chapters:

- [Chapter 20 “The Greenplum Wire Protocol Driver” on page 669](#)
- [Chapter 21 “The Salesforce Driver” on page 699](#)
- [Chapter 22 “The Sybase IQ Wire Protocol Driver” on page 769](#)
- [Chapter 23 “The Driver for Apache Hive™” on page 805](#)
- [Chapter 24 “The Driver for the Teradata Database” on page 821](#)



## 20 The Greenplum Wire Protocol Driver

The DataDirect Connect XE *for* ODBC and DataDirect Connect64 XE *for* ODBC Greenplum Wire Protocol driver (the Greenplum Wire Protocol driver) each support the following Greenplum database servers:

- Greenplum version 4.0, 4.1, 4.2
- Greenplum version 3.3

The Greenplum Wire Protocol driver is supported in the Windows, UNIX, and Linux environments. See ["Environment-Specific Information" on page 45](#) for detailed information about the Windows, UNIX, and Linux environments supported by this driver.

Refer to the readme file shipped with your DataDirect Connect XE product for the file name of the Greenplum Wire Protocol driver.

---

### Driver Requirements

The driver has no client requirements.

---

### Configuring and Connecting to Data Sources

After you install the driver, you configure data sources to connect to the database. See [Chapter 1 "Quick Start Connect" on page 35](#) for an explanation of different types of data sources. The data source contains connection options that allow you to tune the driver for specific performance. If you want to use a data source but need to change some of its values, you can either modify the data source or override its values at connection time through a connection string.

If you choose to use a connection string, you must use specific connection string attributes. See ["Using a Connection String" on page 676](#) and [Table 20-1 on page 678](#) for an alphabetical list of driver connection string attributes and their initial default values.

#### Data Source Configuration in the UNIX/Linux odbc.ini File

On UNIX and Linux, you must set up the proper ODBC environment before configuring data sources. See ["Environment Configuration" on page 37](#) for basic setup information and ["Environment Variables" on page 97](#) for more detail about this procedure.

Data sources for UNIX and Linux are stored in the system information file (by default, odbc.ini). If you have a Motif GUI environment on Linux, you can configure and modify data sources through the DataDirect ODBC Data Source Administrator for Linux (the Linux ODBC Administrator) using a driver Setup dialog box. (See ["Configuration Through the Administrator" on page 100](#) for a detailed explanation of the Administrator.)

If you do not have a GUI environment, you can configure and modify data sources directly by editing the `odbc.ini` file and storing default connection values there. See ["Configuration Through the System Information \(odbc.ini\) File" on page 103](#) for detailed information about the specific steps necessary to configure a data source.

[Table 20-1 on page 678](#) lists driver connection string attributes that must be used in the `odbc.ini` file to set the value of the attributes. Note that only the long name of the attribute can be used in the file. The default listed in the table is the initial default value when the driver is installed.

## Data Source Configuration through a GUI

On Windows, data sources are stored in the Windows Registry. You can configure and modify data sources through the ODBC Administrator using a driver Setup dialog box, as described in this section.

On UNIX and Linux, data sources are stored in the `odbc.ini` file. On Linux, you can configure and modify data sources through the Linux ODBC Administrator using a driver Setup dialog box, as described in this section.

**NOTE:** This book shows dialog box images that are specific to Windows. If you are using the drivers in the Linux environment, the dialog box that you see may differ slightly from the Windows version. Windows-only and UNIX-only connection options are specifically noted by icons in the Setup dialog box descriptions.

When the driver is first installed, the values of its connection options are set by default. These values appear on the driver Setup dialog box tabs when you create a new data source. You can change these default values by modifying the data source. In the following procedure, the description of each tab is followed by a table that lists the connection options for that tab and their initial default values. This table links you to a complete description of the options and their connection string attribute equivalents. The connection string attributes are used to override the default values of the data source if you want to change these values at connection time.

### To configure a Greenplum data source:

- 1 Start the ODBC Administrator:
  - On Windows, start the ODBC Administrator by selecting its icon from the DataDirect Connect program group.
  - On Linux, change to the `install_dir/tools` directory and, at a command prompt, enter:
 

```
odbcadmin
```

 where `install_dir` is the path to the product installation directory.
- 2 Select a tab:
  - **User DSN:** If you are configuring an existing user data source, select the data source name and click **Configure** to display the driver Setup dialog box.
 

If you are configuring a new user data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.

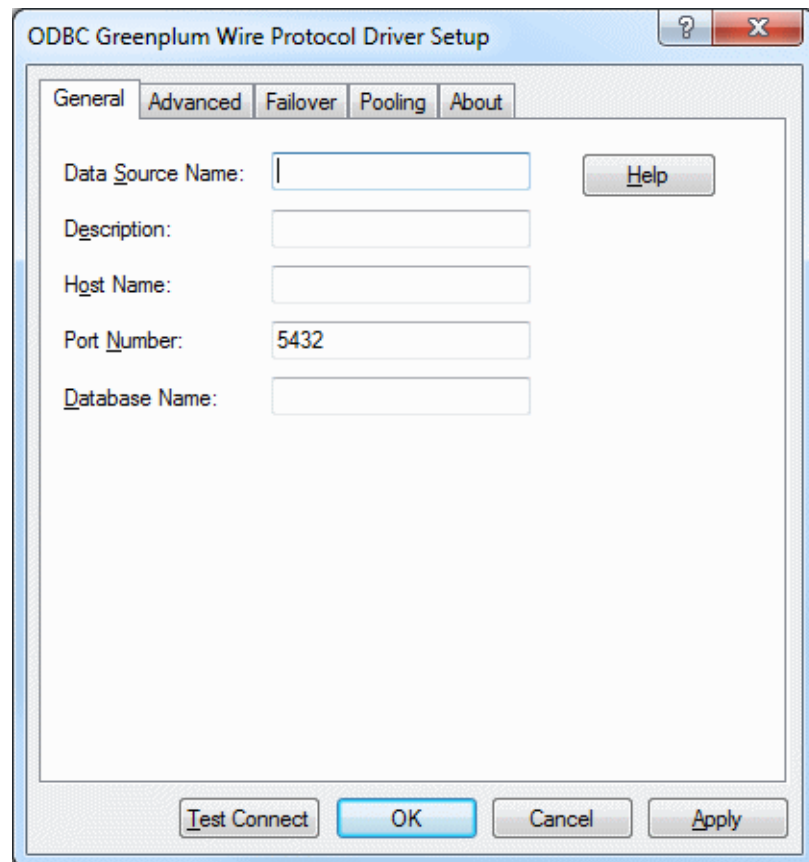
- **System DSN:** If you are configuring an existing system data source, select the data source name and click **Configure** to display the driver Setup dialog box.

If you are configuring a new system data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.

- **File DSN:** If you are configuring an existing file data source, select the data source file and click **Configure** to display the driver Setup dialog box.

If you are configuring a new file data source, click **Add** to display a list of installed drivers; then, select a driver. Click **Advanced** if you want to specify attributes; otherwise, click **Next** to proceed. Specify a name for the data source and click **Next**. Verify the data source information; then, click **Finish** to display the driver Setup dialog box.

The General tab of the Setup dialog box appears.

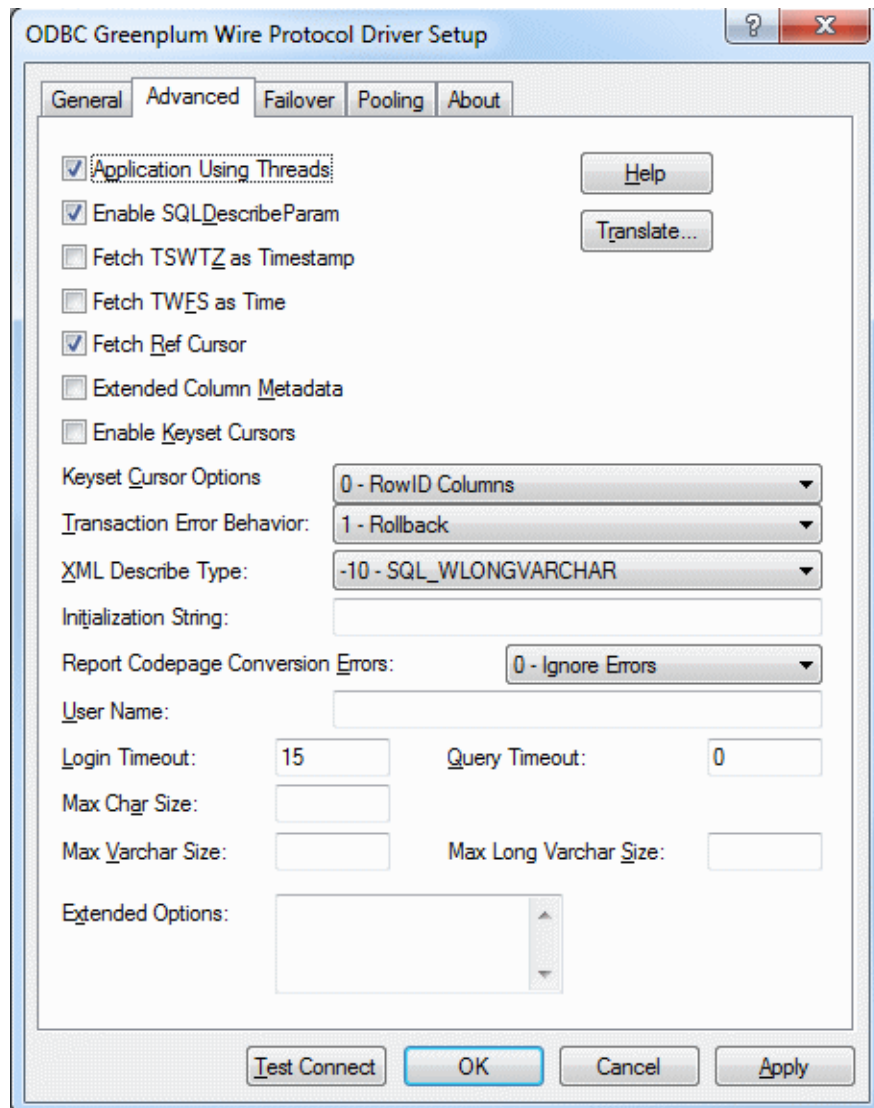


NOTE: The General tab displays only fields that are required for creating a data source. The fields on all other tabs are optional, unless noted otherwise.

- 3 On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: General	Default
<a href="#">Data Source Name</a> (see page 681)	None
<a href="#">Description</a> (see page 682)	None
<a href="#">Host Name</a> (see page 686)	None
<a href="#">Port Number</a> (see page 691)	5432
<a href="#">Database Name</a> (see page 681)	None

- 4 Optionally, click the **Advanced** tab to specify additional data source settings.





On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Advanced	Default
<a href="#">Application Using Threads (see page 679)</a>	Enabled
<a href="#">Enable SQLDescribeParam (see page 682)</a>	Enabled
<a href="#">Fetch TSWTZ as Timestamp (see page 685)</a>	Disabled
<a href="#">Fetch TWFS as Time (see page 686)</a>	Disabled
<a href="#">Fetch RefCursors (see page 685)</a>	Enabled
<a href="#">Extended Column Metadata (see page 683)</a>	Disabled
<a href="#">User Name (see page 693)</a>	None
<a href="#">Enable Keyset Cursors</a>	Disabled
<a href="#">Keyset Cursor Options</a>	0 - RowID Columns
<a href="#">Transaction Error Behavior (see page 692)</a>	1 - Rollback
<a href="#">Report Codepage Conversion Errors (see page 692)</a>	0 - Ignore Errors
<a href="#">Login Timeout (see page 689)</a>	15
<a href="#">Query Timeout (see page 691)</a>	0
<a href="#">Initialization String (see page 687)</a>	None
<a href="#">Max Char Size (see page 689)</a>	None
<a href="#">Max Long Varchar Size (see page 689)</a>	None
<a href="#">Max Varchar Size (see page 690)</a>	None
<a href="#">IANAAppCodePage (see page 686)</a>	4 (ISO 8559-1 Latin-1)
UNIX ONLY	

**Extended Options:** Type a semi-colon separated list of connection options and their values. Use this configuration option to set the value of undocumented connection options that are provided by Progress DataDirect customer support. You can include any valid connection option in the Extended Options string, for example:

```
Database=Server1;UndocumentedOption1=value[;UndocumentedOption2=value;]
```

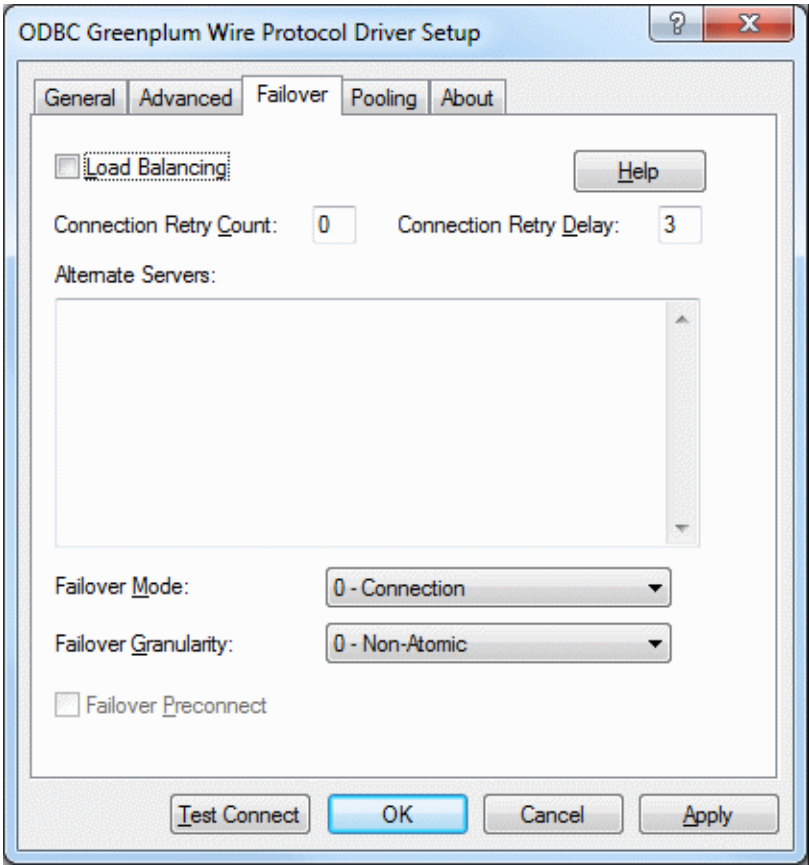
If the Extended Options string contains option values that are also set in the setup dialog or data source, the values of the options specified in the Extended Options string take precedence. However, connection options that are specified on a connection string override any option value specified in the Extended Options string.

**NOTE:** Do not specify the Extended Options configuration option in a connection string, or the driver will return an error. Instead, applications should specify the individual undocumented connection options in the connection string.

**Translate:** Click **Translate** to display the Select Translator dialog box, which lists the translators specified in the ODBC Translators section of the Registry. Progress 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.

- 5 Optionally, click the **Failover** tab to specify failover data source settings.

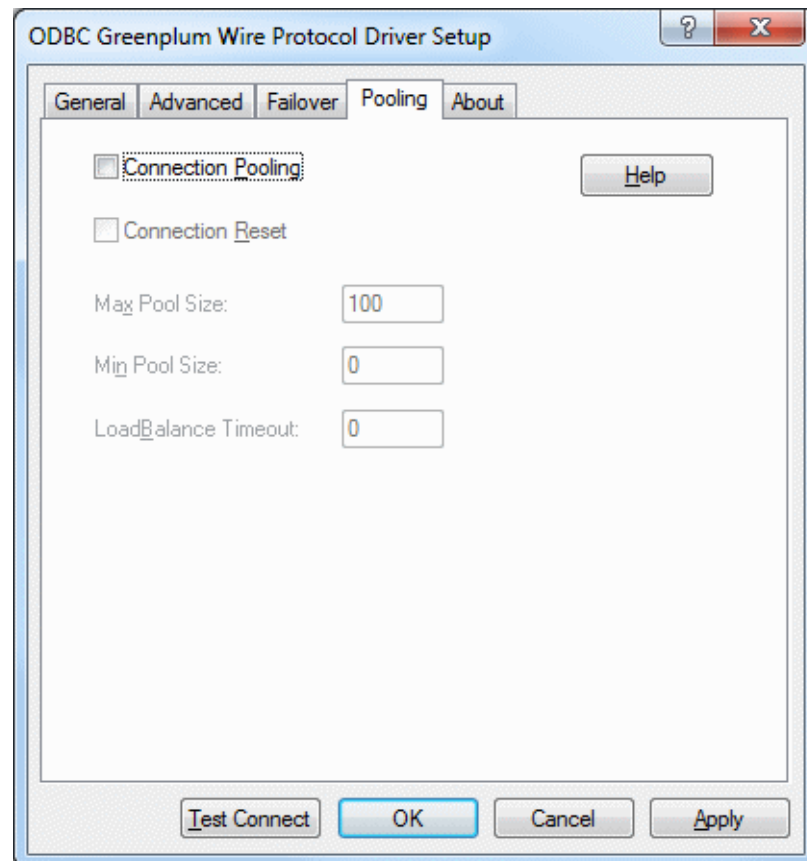


See ["Using Failover" on page 65](#) for a general description of failover and its related connection options.

On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Failover	Default
<a href="#">Load Balancing (see page 688)</a>	Disabled
<a href="#">Connection Retry Count (see page 680)</a>	0
<a href="#">Connection Retry Delay (see page 681)</a>	3
<a href="#">Alternate Servers (see page 679)</a>	None
<a href="#">Failover Mode (see page 684)</a>	0 - Connection
<a href="#">Failover Granularity (see page 683)</a>	0 - Non-Atomic
<a href="#">Failover Preconnect (see page 684)</a>	Disabled

- 6 Optionally, click the **Pooling** tab to specify pooling data source settings.



See ["Using DataDirect Connection Pooling" on page 82](#) for a general description of connection pooling.

On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Pooling	Default
<a href="#">Connection Pooling (see page 680)</a>	Disabled
<a href="#">Connection Reset (see page 680)</a>	Disabled
<a href="#">Max Pool Size (see page 690)</a>	100
<a href="#">Min Pool Size (see page 690)</a>	0
<a href="#">Load Balance Timeout (see page 688)</a>	0

- 7 At any point during the configuration process, you can click **Test Connect** to attempt to connect to the data source using the connection options specified in the driver Setup dialog box. A logon dialog box appears (see ["Using a Logon Dialog Box" on page 677](#) for details). 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 incorrect environment or connection value, it displays an appropriate error message. Click **OK**.

NOTE: If you are configuring alternate servers for use with the connection failover feature, be aware that the Test Connect button tests only the primary server, not the alternate servers.

- 8 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.

## Using a Connection String

If you want to use a connection string for connecting to a database, or if your application requires it, you must specify either a DSN (data source name), a File DSN, or a DSN-less connection in the string. The difference is whether you use the DSN=, FILEDSN=, or the DRIVER= keyword in the connection string, as described in the ODBC specification. A DSN or FILEDSN connection string tells the driver where to find the default connection information. Optionally, you may specify *attribute=value* pairs in the connection string to override the default values stored in the data source.

The DSN connection string has the form:

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

The FILEDSN connection string has the form:

```
FILEDSN=filename.dsn[;attribute=value[;attribute=value]...]
```

The DSN-less connection string specifies a driver instead of a data source. All connection information must be entered in the connection string because the information is not stored in a data source.

The DSN-less connection string has the form:

```
DRIVER=[{ }driver_name[ } ][;attribute=value[;attribute=value]...]
```

[Table 20-1](#) lists the long and short names for each attribute, as well as the initial default value when the driver is first installed. You can specify either long or short names in the connection string.

An example of a DSN connection string with overriding attribute values for Greenplum Wire Protocol is:

```
DSN=Accounting;UID=JOHN;PWD=XYZZY
```

A FILEDSN connection string is similar except for the initial keyword:

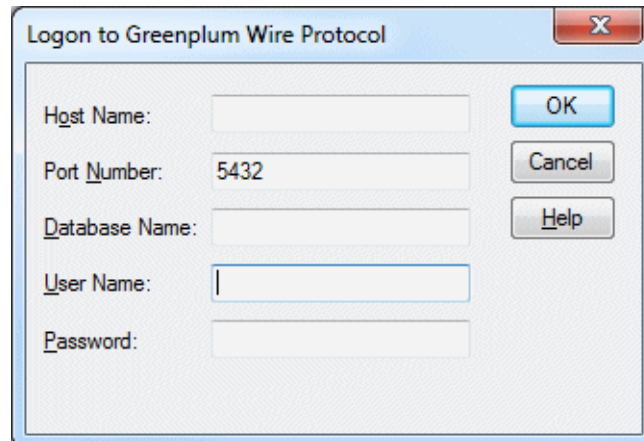
```
FILEDSN=GreenplumWP.dsn;UID=JOHN;PWD=XYZZY
```

A DSN-less connection string must provide all necessary connection information:

```
DRIVER=DataDirect 7.1 Greenplum Wire Protocol;  
HOST=GreenplumServer;PORT=5432;UID=JOHN;PWD=XYZZY;DB=Gplumdb1
```

## 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, provide the following information:

- 1 In the Host Name field, type either the name or the IP address of the server to which you want to connect. The IP address must be in IPv4 format.
- 2 In the Port Number field, type the number of your Greenplum listener. Check with your database administrator for the correct number.
- 3 In the Database Name field, type the name of the database to which you want to connect.
- 4 If required, type your Greenplum user name.
- 5 If required, type your Greenplum password.
- 6 Click **OK** to log on to the Greenplum database installed on the server you specified and to update the values in the Registry.

---

## Connection Option Descriptions

The following connection option descriptions are listed alphabetically by the GUI name that appears on the driver Setup dialog box. The connection string attribute name, along with its short name, is listed immediately underneath the GUI name. For example:

### Application Using Threads

Attribute    ApplicationUsingThreads (AUT)

In most cases, the GUI name and the attribute name are the same; however, some exceptions exist. If you need to look up an option by its connection string attribute name, please refer to the alphabetical table of connection string attribute names.

Also, a few connection string attributes, for example, Password, do not have equivalent options that appear on the GUI. They are in the list of descriptions alphabetically by their attribute names.

Table 20-1 lists the connection string attributes supported by the Greenplum Wire Protocol driver.

**Table 20-1. Greenplum Wire Protocol Attribute Names**

Attribute (Short Name)	Default
AlternateServers (ASRV)	None
ApplicationUsingThreads (AUT)	1 (Enabled)
ConnectionReset (CR)	0 (Disabled)
ConnectionRetryCount (CRC)	0
ConnectionRetryDelay (CRD)	3
Database (DB)	None
DataSourceName (DSN)	None
Description (n/a)	None
EnableDescribeParam (EDP)	1 (Enabled)
EnableKeysetCursors (EKC)	0
ExtendedColumnMetaData (ECMD)	0 (Disabled)
FailoverGranularity (FG)	0 (Non-Atomic)
FailoverMode (FM)	0 (Connection)
FailoverPreconnect (FP)	0 (Disabled)
FetchRefCursors (FRC)	1 (Enabled)
FetchTSWTZasTimestamp (FTSWTZAT)	0 (Disabled)
FetchTWFSasTime (FTWFSAT)	0 (Disabled)
HostName (HOST)	None
IANAAppCodePage (IACP)	4 (ISO 8559-1 Latin-1)
UNIX ONLY	
InitializationString (IS)	None
KeysetCursorOptions (KCO)	0 - RowID Columns
LoadBalanceTimeout (LBT)	0
LoadBalancing (LB)	0 (Disabled)
LoginTimeout (LT)	15
LogonID (UID)	None
MaxCharSize (MCS) (see page 689)	None
MaxLongVarcharSize (MLVS)	None
MaxPoolSize (MXPS)	100
MaxVarcharSize (MVS)	None
MinPoolSize (MNPS)	0
Password (PWD)	None
Pooling (POOL)	0 (Disabled)

**Table 20-1. Greenplum Wire Protocol Attribute Names** (cont.)

Attribute (Short Name)	Default
<a href="#">PortNumber (PORT)</a>	5432
<a href="#">QueryTimeout (QT)</a>	0
<a href="#">ReportCodepageConversionErrors (RCCE)</a>	0 (Ignore Errors)
<a href="#">TransactionErrorBehavior (TEB)</a>	1 (Rollback Transaction)

## Alternate Servers

Attribute	AlternateServers (ASRV)
Description	A list of alternate database servers to which the driver tries to connect if the primary database server is unavailable. Specifying a value for this option enables connection failover for the driver. The value you specify must be in the form of a string that defines the physical location of each alternate server. All of the other required connection information for each alternate server is the same as what is defined for the primary server connection.
Valid Values	(HostName=hostvalue:PortNumber=portvalue:Database=databasevalue[, . . .])  You must specify the host name, port number, and database name of each alternate server.
Example	The following Alternate Servers value defines two alternate database servers for connection failover:  <code>AlternateServers=(HostName=GreenplumServer:PortNumber=5431:Database=Pgredb1, HostName=255.201.11.24:PortNumber=5432:Database=Pgredb2)</code>
Default	None
GUI Tab	<a href="#">Failover tab</a> on page 674

## Application Using Threads

Attribute	ApplicationUsingThreads (AUT)
Description	Determines whether the driver works with applications using multiple ODBC threads.  This connection option can affect performance. See " <a href="#">Performance Considerations</a> " on <a href="#">page 693</a> for details.
Valid Values	0   1  If set to 1 (Enabled), the driver works with single-threaded and multi-threaded applications.  If set to 0 (Disabled), the driver does not work with multi-threaded applications. If using the driver with single-threaded applications, this value avoids additional processing required for ODBC thread-safety standards.
Default	1 (Enabled)
GUI Tab	<a href="#">Advanced tab</a> on page 672

## Connection Pooling

Attribute	Pooling (POOL)
Description	Specifies whether to use the driver's connection pooling.  NOTE: The application must be thread-enabled to use connection pooling.  This connection option can affect performance. See <a href="#">"Performance Considerations" on page 693</a> for details.
Valid Values	0   1  If set to 1 (Enabled), the driver uses connection pooling.  If set to 0 (Disabled), the driver does not use connection pooling.
Default	0 (Disabled)
GUI Tab	<a href="#">Pooling tab</a> on page 675

## Connection Reset

Attribute	ConnectionReset (CR)
Description	Determines whether the state of connections that are removed from the connection pool for reuse by the application is reset to the initial configuration of the connection.  This connection option can affect performance. See <a href="#">"Performance Considerations" on page 693</a> for details.
Valid Values	0   1  If set to 1 (Enabled), the state of connections removed from the connection pool for reuse by an application is reset to the initial configuration of the connection. Resetting the state can negatively impact performance because additional commands must be sent over the network to the server to reset the state of the connection.  If set to 0 (Disabled), the state of connections is not reset.
Default	0 (Disabled)
GUI Tab	<a href="#">Pooling tab</a> on page 675

## Connection Retry Count

Attribute	ConnectionRetryCount (CRC)
Description	The number of times the driver retries connection attempts to the primary database server, and if specified, alternate servers until a successful connection is established.  This option and the Connection Retry Delay connection option, which specifies the wait interval between attempts, can be used in conjunction with failover.
Valid Values	0   x  where x is a positive integer from 1 to 65535.  If set to 0, the driver does not try to connect after the initial unsuccessful attempt.



If set to  $x$ , the driver retries connection attempts the specified number of times. If a connection is not established during the retry attempts, the driver returns an error that is generated by the last server to which it tried to connect.

Default 0

GUI Tab [Failover tab](#) on page 674

## Connection Retry Delay

Attribute ConnectionRetryDelay (CRD)

Description The number of seconds the driver waits between connection retry attempts when Connection Retry Count is set to a positive integer.

This option and the Connection Retry Count connection option can be used in conjunction with failover.

Valid Values 0 |  $x$

where  $x$  is a positive integer from 1 to 65535.

If set to 0, there is no delay between retries.

If set to  $x$ , the driver waits the specified number of seconds between connection retry attempts.

Default 3

GUI Tab [Failover tab](#) on page 674

## Data Source Name

Attribute DataSourceName (DSN)

Description The name of a data source in your Windows Registry or odbc.ini file.

Valid Values *string*

where *string* is the name of a data source.

Default None

GUI Tab [General tab](#) on page 672

## Database Name

Attribute Database (DB)

Description The name of the database to which you want to connect.

Valid Values *database\_name*

where *database\_name* is the name of a valid database.

Default None

GUI Tab [General tab](#) on page 672

## Description

Attribute	Description (n/a)
Description	An optional long description of a data source. This description is not used as a runtime connection attribute, but does appear in the ODBC.INI section of the Registry and in the <code>odbc.ini</code> file.
Valid Values	<i>string</i>  where <i>string</i> is a description of a data source.
Default	None
GUI Tab	<a href="#">General tab</a> on page 672

## Enable Keyset Cursors

Attribute	EnableKeysetCursors (EKC)
Description	Determines whether the driver emulates keyset cursors to provide scrollable keyset cursors to an ODBC application.
Valid Values	0   1  If set to 1 (Enabled), the driver emulates keyset cursors.  If set to 0 (Disabled), the driver does not emulate keyset cursors. If an application requests a keyset cursor and this option is set to 0, the driver uses a static cursor and returns a message that a different value was used.
Default	0 (Disabled)
GUI Tab	<a href="#">Advanced tab</a> on page 672

## Enable SQLDescribeParam

Attribute	EnableDescribeParam (EDP)
Description	Determines whether SQLDescribeParam returns the Datatype, ParameterSize, DecimalDigits, and Nullable information for parameters in a prepared statement.
Valid Values	0   1  If set to 1 (Enabled), SQLDescribeParam returns the Datatype, ParameterSize, DecimalDigits, and Nullable information for parameters in a prepared statement.  If set to 0 (Disabled), the driver does not support SQLDescribeParam and returns the message: <code>Driver does not support this function.</code>
Default	1 (Enabled)
GUI Tab	<a href="#">Advanced tab</a> on page 672

## Extended Column Metadata

Attribute	ExtendedColumnMetaData (ECMD)
Description	Determines how the driver returns column metadata when using SQLDescribeCol and SQLColAttribute.
Valid Values	<p>0   1</p> <p>If set to 1 (Enabled), SQLDescribeCol returns the actual values for Data Type, Column Size, Decimal Digits, and Nullable. SQLColAttribute returns the actual values for:</p> <ul style="list-style-type: none"> <li>■ SQL_DESC_CATALOG_NAME: <i>catalog_name</i></li> <li>■ SQL_DESC_TABLE_NAME: <i>table_name</i></li> <li>■ SQL_DESC_BASE_COLUMN_NAME: <i>base_column_name</i></li> <li>■ SQL_DESC_LOCAL_TYPE_NAME: <i>local_type_name</i></li> <li>■ SQL_DESC_NULLABLE: <i>nullable</i></li> <li>■ SQL_DESC_AUTO_UNIQUE_VALUE: <i>auto_unique_value</i></li> </ul> <p>If set to 0 (Disabled), SQLDescribeCol returns the Data Type, Column Size, and Decimal Digits for the column. The value SQL_NULLABLE_UNKNOWN is returned for Nullable. SQLColAttribute returns the following attribute values:</p> <ul style="list-style-type: none"> <li>■ SQL_DESC_CATALOG_NAME: empty string</li> <li>■ SQL_DESC_TABLE_NAME: empty string</li> <li>■ SQL_DESC_BASE_COLUMN_NAME: empty string</li> <li>■ SQL_DESC_LOCAL_TYPE_NAME: empty string</li> <li>■ SQL_DESC_NULLABLE: SQL_NULLABLE_UNKNOWN</li> <li>■ SQL_DESC_AUTO_UNIQUE_VALUE: SQL_FALSE</li> </ul>
Default	0 (Disabled)
GUI Tab	<a href="#">Advanced tab</a> on page 672

## Failover Granularity

Attribute	FailoverGranularity (FG)
Description	<p>Determines whether the driver fails the entire failover process or continues with the process if errors occur while trying to reestablish a lost connection.</p> <p>This option applies only when Failover Mode is set to 1 (Extended Connection) or 2 (Select).</p> <p>The Alternate Servers option specifies one or multiple alternate servers for failover and is required for all failover methods.</p>
Valid Values	<p>0   1   2   3</p> <p>If set to 0 (Non-Atomic), the driver continues with the failover process and posts any errors on the statement on which they occur.</p> <p>If set to 1 (Atomic) the driver fails the entire failover process if an error is generated as the result of anything other than executing and repositioning a Select statement. If an error is generated as a result of repositioning a result set to the last row position, the driver continues with the failover process, but generates a warning that the Select statement must be reissued.</p>

If set to 2 (Atomic Including Repositioning), the driver fails the entire failover process if any error is generated as the result of restoring the state of the connection or the state of work in progress.

If set to 3 (Disable Integrity Check), the driver does not verify that the rows that were restored during the failover process match the original rows. This value applies only when Failover Mode is set to 2 (Select).

Default 0 (Non-Atomic)  
GUI Tab [Failover tab](#) on page 674

**Failover Mode**

Attribute FailoverMode (FM)  
Description Specifies the type of failover method the driver uses.  
  
The Alternate Servers option specifies one or multiple alternate servers for failover and is required for all failover methods.  
Valid Values 0 | 1 | 2  
  
If set to 0 (Connection), the driver provides failover protection for new connections only.  
  
If set to 1 (Extended Connection), the driver provides failover protection for new and lost connections, but not any work in progress.  
  
If set to 2 (Select), the driver provides failover protection for new and lost connections. In addition, it preserves the state of work performed by the last Select statement executed.  
Default 0 (Connection)  
GUI Tab [Failover tab](#) on page 674

**Failover Preconnect**

Attribute FailoverPreconnect (FP)  
Description Specifies whether the driver tries to connect to the primary and an alternate server at the same time.  
  
This attribute applies only when Failover Mode is set to 1 (Extended Connection) or 2 (Select) and at least one alternate server is specified.  
  
The Alternate Servers option specifies one or multiple alternate servers for failover and is required for all failover methods.  
Valid Values 0 | 1  
  
If set to 0 (Disabled), the driver tries to connect to an alternate server only when failover is caused by an unsuccessful connection attempt or a lost connection. This value provides the best performance, but your application typically experiences a short wait while the failover connection is attempted.

If set to 1 (Enabled), the driver tries to connect to the primary and an alternate server at the same time. This can be useful if your application is time-sensitive and cannot absorb the wait for the failover connection to succeed.

Default 0 (Disabled)

GUI Tab [Failover tab](#) on page 674

## Fetch RefCursors

Attribute FetchRefCursors (FRC)

Description Determines whether the driver returns refcursors from stored procedures as results sets.

Valid Values 0 | 1

If set to 1 (Enabled), the driver returns refcursors from stored procedures as result sets. The driver fetches all the data from the refcursor and then closes the refcursor. If a stored procedure returns multiple refcursors, the driver generates multiple result sets, one for each refcursor returned.

If set to 0 (Disabled), the driver returns the cursor name for refcursors. The application must fetch the actual data from the refcursor using the cursor name and must close the cursor before additional processing can be done on the statement. The application must close the cursor regardless of whether it actually fetches data from the cursor.

Default 1 (Enabled)

GUI Tab [Advanced tab](#) on page 672

## Fetch TSWTZ as Timestamp

Attribute FetchTSWTZasTimestamp (FTSWTZAT)

Description Determines whether the driver returns column values with the timestamp with time zone data type as the ODBC data type SQL\_TYPE\_TIMESTAMP or SQL\_VARCHAR.

Valid Values 0 | 1

If set to 1 (Enabled), the driver returns column values with the timestamp with time zone data type as the ODBC type SQL\_TYPE\_TIMESTAMP. The time zone information in the fetched value is truncated. Use this value if your application needs to process values the same way as TIMESTAMP columns.

If set to 0 (Disabled), the driver returns column values with the timestamp with time zone data type as the ODBC data type SQL\_VARCHAR. Use this value if your application requires the time zone information in the fetched value.

Default 0 (Disabled)

GUI Tab [Advanced tab](#) on page 672

## Fetch TWFS as Time

Attribute	FetchTWFSasTime (FTWFSAT)
Description	Determines whether the driver returns column values with the time data type as the ODBC data type SQL_TYPE_TIME or SQL_TYPE_TIMESTAMP.
Valid Values	0   1  If set to 1 (Enabled), the driver returns column values with the time data type as the ODBC data type SQL_TYPE_TIME. The fractional seconds portion of the value is truncated.  If set to 0 (Disabled), the driver returns column values with the time data type as the ODBC data type SQL_TYPE_TIMESTAMP. The fractional seconds portion of the value is preserved. Time columns are not searchable when they are described and fetched as timestamp  NOTE: When returning time with fractional seconds data as SQL_TYPE_TIMESTAMP, the Year, Month and Day parts of the timestamp must be set to zero.
Default	0 (Disabled)
GUI Tab	<a href="#">Advanced tab</a> on page 672

## Host Name

Attribute	HostName (HOST)
Description	The name or the IP address of the server to which you want to connect.
Valid Values	<i>server_name</i>   <i>IP_address</i>  where:  <i>server_name</i> is the name of the server to which you want to connect.  <i>IP_address</i> is the IP address of the server to which you want to connect.  The IP address must be in IPv4 format.
Default	None
GUI Tab	<a href="#">General tab</a> on page 672

## IANAAppCodePage

Attribute	IANAAppCodePage (IACP)
Description	An Internet Assigned Numbers Authority (IANA) value. You must specify a value for this option if your application is not Unicode-enabled or if your database character set is not Unicode. Refer to <a href="#">Chapter 4 “Internationalization, Localization, and Unicode”</a> in the <i>DataDirect Connect Series for ODBC Reference</i> for details.  The driver uses the specified IANA code page to convert "W" (wide) functions to ANSI.

The driver and Driver Manager both check for the value of `IANAAppCodePage` in the following order:

- In the connection string
- In the Data Source section of the system information file (odbc.ini)
- In the ODBC section of the system information file (odbc.ini)

If the driver does not find an `IANAAppCodePage` value, the driver uses the default value of 4 (ISO 8859-1 Latin-1).

Valid Values *IANA\_code\_page*

where *IANA\_code\_page* is one of the valid values listed in [Chapter 1 “Values for the Attribute IANAAppCodePage”](#) in the *DataDirect Connect Series for ODBC Reference*. The value must match the database character encoding and the system locale.

Default 4 (ISO 8559-1 Latin-1)

GUI Tab [Advanced tab](#) on page 672

## Initialization String

Attribute InitializationString (IS)

Description A SQL command that is issued immediately after connecting to the database to manage session settings.

NOTE: If the statement fails to execute, the connection fails and the driver reports the error returned from the server.

Valid Values *SQL\_command*

where *SQL\_command* is a valid SQL command that is supported by the database.

Example To set the date format on every connection, specify:

```
Set DateStyle='ISO, MDY'
```

Default None

GUI Tab [Advanced tab](#) on page 672

## Keyset Cursor Options

Attribute KeysetCursorOptions (KCO)

Description Determines which columns are used to comprise the keyset that the driver uses to create the initial keyset on which cursor operations are based. Greenplum does not offer a true row identifier column; the driver instead uses two hidden system columns provided by the Greenplum database, `ctid` and `gp_segment_id`. Because the database might reassign these IDs following a Vacuum operation, the driver can be configured to also include other columns to help ensure that data integrity is maintained.

NOTE: This option has no effect unless the [Enable Keyset Cursors](#) connection option is enabled.

Valid Values 0 | 1

If set to 1 - RowID and Searchable Columns (Enabled), the driver uses a combination of every non-LOB column in the Select list and the ctid and gp\_segment\_id hidden columns to build the keyset. By adding other Select list fields to the keyset, the driver is able to indicate the row cannot be found if the IDs change following a Vacuum operation.

If set to 0 - RowID Columns (Disabled), the driver uses the ctid and gp\_segment\_id hidden system columns.

Default 0

GUI Tab [Advanced tab](#) on page 672

## Load Balance Timeout

Attribute LoadBalanceTimeout (LBT)

Description The number of seconds to keep inactive connections open in a connection pool. An inactive connection is a database session that is not associated with an ODBC connection handle, that is, a connection in the pool that is not in use by an application.

NOTE: The Min Pool Size option may cause some connections to ignore this value.

This connection option can affect performance. See ["Performance Considerations" on page 693](#) for details.

Valid Values 0 | *x*

where *x* is a positive integer that specifies a number of seconds.

If set to 0, inactive connections are kept open.

If set to *x*, inactive connections are closed after the specified number of seconds passes.

Default 0

GUI Tab [Pooling tab](#) on page 675

## Load Balancing

Attribute LoadBalancing (LB)

Description Determines whether the driver uses client load balancing in its attempts to connect to the database servers (primary and alternate). You can specify one or multiple alternate servers by setting the Alternate Servers option.

Valid Values 0 | 1

If set to 1 (Enabled), the driver uses client load balancing and attempts to connect to the database servers (primary and alternate servers) in random order.

If set to 0 (Disabled), the driver does not use client load balancing and connects to each server based on their sequential order (primary server first, then, alternate servers in the order they are specified).

NOTE: This option has no effect unless alternate servers are defined for the Alternate Servers connection option.



Default 0 (Disabled)  
 GUI Tab [Failover tab](#) on page 674

## Login Timeout

Attribute LoginTimeout (LT)

Description The number of seconds the driver waits for a connection to be established before returning control to the application and generating a timeout error. To override the value that is set by this connection option for an individual connection, set a different value in the SQL\_ATTR\_LOGIN\_TIMEOUT connection attribute using the SQLSetConnectAttr() function.

Valid Values -1 | 0 |  $x$

where  $x$  is a positive integer that specifies a number of seconds.

If set to -1, the connection request does not time out. The driver silently ignores the SQL\_ATTR\_LOGIN\_TIMEOUT attribute.

If set to 0, the connection request does not time out, but the driver responds to the SQL\_ATTR\_LOGIN\_TIMEOUT attribute.

If set to  $x$ , the connection request times out after the specified number of seconds unless the application overrides this setting with the SQL\_ATTR\_LOGIN\_TIMEOUT attribute.

Default 15

GUI Tab [Advanced tab](#) on page 672

## Max Char Size

Attribute MaxCharSize (MCS)

Description Specifies the maximum size of columns of type SQL\_CHAR that the driver describes through result set descriptions and catalog functions.

Valid Values A positive integer from 1 to 10485760

When not specified, the actual size of the columns from the database is persisted to the application.

If you specify a value that is not in the specified range, the driver uses the maximum value of the SQL\_CHAR data type.

Default None. The actual size of the columns from the database is persisted to the application.

GUI Tab Advanced

## Max Long Varchar Size

Attribute MaxLongVarcharSize (MLVS)

Description Specifies the maximum size of columns of type SQL\_LONGVARCHAR that the driver describes through result set descriptions and catalog functions.

Valid Values	A positive integer from 1 to $x$  where $x$ is maximum size of the SQL_LONGVARCHAR data type.
Default	None. The actual size of the columns from the database is persisted to the application.
GUI Tab	Advanced

## Max Pool Size

Attribute	MaxPoolSize (MXPS)
Description	The maximum number of connections allowed within a single connection pool. When the maximum number of connections is reached, no additional connections can be created in the connection pool.  This connection option can affect performance. See <a href="#">"Performance Considerations" on page 693</a> for details.
Valid Values	An integer from 1 to 65535  For example, if set to 20, the maximum number of connections allowed in the pool is 20.
Default	100
GUI Tab	<a href="#">Pooling tab</a> on page 675

## Max Varchar Size

Attribute	MaxVarcharSize (MVS)
Description	Specifies the maximum size of columns of type SQL_VARCHAR that the driver describes through result set descriptions and catalog functions.
Valid Values	A positive integer from 1 to $x$  where $x$ is maximum size of the SQL_VARCHAR data type.
Default	None. The actual size of the columns from the database is persisted to the application.
GUI Tab	Advanced

## Min Pool Size

Attribute	MinPoolSize (MNPS)
Description	The minimum number of connections that are opened and placed in a connection pool, in addition to the active connection, when the pool is created. The connection pool retains this number of connections, even when some connections exceed their Load Balance Timeout value.  This connection option can affect performance. See <a href="#">"Performance Considerations" on page 693</a> for details.
Valid Values	0   $x$  where $x$ is an integer from 1 to 65535.

For example, if set to 5, the start-up number of connections in the pool is 5 in addition to the current existing connection.

If set to 0, no connections are opened in addition to the current existing connection.

Default 0

GUI Tab [Pooling tab](#) on page 675

## Password

Attribute Password (PWD)

Description The password that the application uses to connect to your database. The Password option cannot be specified through the driver Setup dialog box and should not be stored in a data source. It is specified through the Logon dialog box or a connection string.

Valid Values *pwd*

where *pwd* is a valid password.

Default None

GUI Tab n/a

## Port Number

Attribute PortNumber (PORT)

Description The port number of the server listener.

Valid Values *port\_name*

where the *port\_name* is the port number of the server listener. Check with your database administrator for the correct number.

Default 5432

GUI Tab [General tab](#) on page 672

## Query Timeout

Attribute QueryTimeout (QT)

Description The number of seconds for the default query timeout for all statements that are created by a connection. To override the value set by this connection option for an individual statement, set a different value in the SQL\_ATTR\_QUERY\_TIMEOUT statement attribute on the SQLSetStmtAttr() function.

Valid Values -1 | 0 |  $x$

where  $x$  is a positive integer that specifies a number of seconds.

If set to -1, the query does not time out. The driver silently ignores the SQL\_ATTR\_QUERY\_TIMEOUT attribute.

If set to 0, the query does not time out, but the driver responds to the SQL\_ATTR\_QUERY\_TIMEOUT attribute.

If set to  $x$ , all queries time out after the specified number of seconds unless the application overrides this value by setting the SQL\_ATTR\_QUERY\_TIMEOUT attribute.

Default 0

GUI Tab [Advanced tab](#) on page 672

## Report Codepage Conversion Errors

Attribute ReportCodepageConversionErrors (RCCE)

Description Specifies how the driver handles code page conversion errors that occur when a character cannot be converted from one character set to another.

An error message or warning can occur if an ODBC call causes a conversion error, or if an error occurs during code page conversions to and from the database or to and from the application. The error or warning generated is `Code page conversion error encountered`. In the case of parameter data conversion errors, the driver adds the following sentence: `Error in parameter  $x$` , where  $x$  is the parameter number. The standard rules for returning specific row and column errors for bulk operations apply.

Valid Values 0 | 1 | 2

If set to 0 (Ignore Errors), the driver substitutes 0x1A for each character that cannot be converted and does not return a warning or error.

If set to 1 (Return Error), the driver returns an error instead of substituting 0x1A for unconverted characters.

If set to 2 (Return Warning), the driver substitutes 0x1A for each character that cannot be converted and returns a warning.

Default 0 (Ignore Errors)

GUI Tab [Advanced tab](#) on page 672

## Transaction Error Behavior

Attribute TransactionErrorBehavior (TEB)

Description Determines how the driver handles errors that occur within a transaction. When an error occurs in a transaction, the Greenplum server does not allow any operations on the connection except for rolling back the transaction.

Valid Values 0 | 1

If set to 0 (None), the driver does not roll back the transaction when an error occurs. The application must handle the error and roll back the transaction. Any operation on the statement other than a rollback results in an error.

If set to 1 (Rollback Transaction), the driver rolls back the transaction when an error occurs. In addition to the original error message, the driver posts an error message indicating that the transaction has been rolled back.

Default 1 (Rollback Transaction)

GUI Tab [Advanced tab](#) on page 672

## User Name

Attribute LogonID (UID)

Description The default user ID that is used to connect to your database. Your ODBC application may override this value or you may override it in the logon dialog box or connection string.

Valid Values *userid*

where *userid* is a valid user ID with permissions to access the database.

Default None

GUI Tab [Advanced tab](#) on page 672

---

## Performance Considerations

The following connection options can enhance driver performance. You can also enhance performance through efficient application design. Refer to [Chapter 5 “Designing ODBC Applications for Performance Optimization”](#) in the *DataDirect Connect Series for ODBC Reference* for details.

**Application Using Threads (ApplicationUsingThreads):** The driver coordinates concurrent database operations (operations from different threads) by acquiring locks. Although locking prevents errors in the driver, it also decreases performance. If your application does not make ODBC calls from different threads, the driver has no reason to coordinate operations. In this case, the ApplicationUsingThreads attribute should be disabled (set to 0).

NOTE: If you are using a multi-threaded application, you must enable the Application Using Threads option.

**Connection Pooling (ConnectionPooling):** If you enable the driver to use connection pooling, you can set additional options that affect performance:

- **Load Balance Timeout:** You can define how long to keep connections in the pool. The time that a connection was last used is compared to the current time and, if the timespan exceeds the value of the Load Balance Timeout option, the connection is destroyed. The Min Pool Size option can cause some connections to ignore this value.

- **Connection Reset:** Resetting a re-used connection to the initial configuration settings impacts performance negatively because the connection must issue additional commands to the server.
- **Max Pool Size:** Setting the maximum number of connections that the pool can contain too low might cause delays while waiting for a connection to become available. Setting the number too high wastes resources.
- **Min Pool Size:** A connection pool is created when the first connection with a unique connection string connects to the database. The pool is populated with connections up to the minimum pool size, if one has been specified. The connection pool retains this number of connections, even when some connections exceed their Load Balance Timeout value.

**Failover Mode (FailoverMode):** Although high availability that replays queries after a failure provides increased levels of protection, it can adversely affect performance because of increased overhead.

---

# Data Types

Table 20-2 shows how the Greenplum data types are mapped to the standard ODBC data types.

Table 20-2. Greenplum Data Types	
Greenplum	ODBC
Bigint	SQL_BIGINT
Bigserial	SQL_BIGINT
Bit <sup>1</sup>	SQL_BIT
Bit varying	SQL_VARBINARY
Boolean	SQL_BIT
Bytea	SQL_VARBINARY
Character	SQL_CHAR
Character varying	SQL_VARCHAR
Date	SQL_TYPE_DATE
Double Precision	SQL_DOUBLE
Integer	SQL_INTEGER
Money	SQL_DOUBLE
Name	SQL_VARCHAR
Numeric <sup>2</sup>	SQL_NUMERIC
Real	SQL_REAL
Serial	SQL_INTEGER
Smallint	SQL_SMALLINT
Text	SQL_LONGVARCHAR

**Table 20-2. Greenplum Data Types (cont.)**

Greenplum	ODBC
Time <sup>3</sup>	SQL_TYPE_TIME
Timestamp	SQL_TYPE_TIMESTAMP
Timestamp with timezone <sup>4</sup>	SQL_VARCHAR

1. Bit maps to SQL\_BIT when the length for the bit is 1. If the length is greater than 1, the driver maps the column to SQL\_BINARY.
2. Numeric maps to SQL\_NUMERIC if the precision of the Numeric is less than or equal to 38. If the precision is greater than 38, the driver maps the column to SQL\_VARCHAR.
3. Time mapping changes based on the setting of the Fetch TWFS as Time option.
4. Timestamp with timezone mapping changes based on the setting of the Fetch TSWTZ as Timestamp option.

See ["Retrieving Data Type Information" on page 59](#) for more information about data types.

## Unicode Support

The Greenplum Wire Protocol driver automatically determines whether the Greenplum database is a Unicode database.

## Advanced Features

The driver supports the following advanced features:

- Failover
- Connection pooling

### Failover

The driver supports failover and its related connection options. Failover connection options are located on the [Failover tab](#) of the driver Setup dialog box. See ["Using Failover" on page 65](#) for a general description of failover and its implementation.

### Connection Pooling

The driver supports connection pooling and its related connection options. Connection pooling connection options are located on the [Pooling tab](#) of the driver Setup dialog box. See ["Using DataDirect Connection Pooling" on page 82](#) for a general description of connection pooling and its implementation.

## Stored Procedure Results

Greenplum provides functionality to create user-defined functions. Greenplum does not make a distinction between user-defined functions and stored procedures. To Greenplum, everything is a user-defined function. Greenplum does not define a call mechanism for invoking a user-defined function. User-defined functions must be invoked via a SQL statement.

For example, a function defined as:

```
CREATE table foo (intcol int, varcharcol varchar(123))
CREATE or REPLACE FUNCTION insertFoo
(IN idVal int, IN nameVal varchar) RETURNS void
AS $$
    insert into foo values ($1, $2);
$$
LANGUAGE SQL;
```

must be invoked natively as:

```
SELECT * FROM insertFoo(100, 'Mark')
```

even though the function does not return a value or results. The Select SQL statement returns a result set that has one column named insertFoo and no row data.

The Greenplum Wire Protocol driver supports invoking user-defined functions using the ODBC call Escape. The previously described function can be invoked using:

```
{call insertFoo(100, 'Mark')}
```

Greenplum functions return data from functions as a result set. If multiple output parameters are specified, the values for the output parameters are returned as columns in the result set. For example, the function defined as:

```
CREATE or REPLACE FUNCTION addValues(in v1 int, in v2 int)
RETURNS int
AS $$
    SELECT $1 + $2;
$$
LANGUAGE SQL;
```

returns a result set with a single column of type SQL\_INTEGER, whereas the function defined as:

```
CREATE or REPLACE FUNCTION selectFooRow2
(IN idVal int, OUT id int, OUT name varchar)
AS $$
    select intcol, varcharcol from foo where intcol = $1;
$$
LANGUAGE SQL
```

returns a result set that contains two columns, a SQL\_INTEGER id column and a SQL\_VARCHAR name column.



In addition, when calling Greenplum functions that contain output parameters, the native syntax requires that the output parameter values be omitted from the function call. This, in addition to output parameter values being returned as a result set, makes the Greenplum behavior of calling functions different from most other databases.

The Greenplum Wire Protocol driver provides a mechanism that makes the invoking of functions more consistent with how other databases behave. In particular, the Greenplum Wire Protocol driver allows parameter markers for output parameters to be specified in the function argument list when the Escape call is used. The driver allows buffers to be bound to these output parameters. When the function is executed, the output parameters are removed from the argument list sent to the server. The driver extracts the output parameter values from the result set returned by the server and updates the bound output parameter buffers with those values. For example, the function `selectFooRow2` described previously can be invoked as:

```
sql = L"{call selectFooRow2(?, ?, ?)}";
retVal = SQLPrepare(hPrepStmt, sql, SQL_NTS);
retVal = SQLBindParameter(
    hPrepStmt, 1, SQL_PARAM_INPUT, SQL_C_LONG,
    SQL_INTEGER, 0, 0, &idBuf, 0, &idInd);
retVal = SQLBindParameter(
    hPrepStmt, 2, SQL_PARAM_OUTPUT, SQL_C_LONG,
    SQL_INTEGER, 0, 0, &idBuf2, 4, &idInd2);
retVal = SQLBindParameter(
    hPrepStmt, 3, SQL_PARAM_INPUT, SQL_C_WCHAR,
    SQL_VARCHAR, 30, 0, &nameBuf, 123, &nameInd);
retVal = SQLExecute(hPrepStmt);
```

The values of the id and name output parameters are returned in the `idBuf2` and `nameBuf` buffers.

If output parameters are bound to a function call, the driver returns the output parameters in the bound buffers. An error is returned if the number of output parameters bound when the function is executed is less than the number of output parameters defined in the function. If no output parameters are bound to a function call, the driver returns the output parameters as a result set.

Greenplum can also return results from a function as a refcursor. There can be, at most, one refcursor per result; however, a function can return multiple results where each result is a refcursor. A connection option defines how the driver handles refcursors. See ["Fetch RefCursors" on page 685](#) for details about this option.

---

## Isolation and Lock Levels Supported

Greenplum supports isolation level 0 (read uncommitted), level 1 (read committed), 2 (Repeatable read), and level 3 (serializable). Greenplum supports record-level locking.

Refer to [Chapter 7 "Locking and Isolation Levels"](#) in the *DataDirect Connect Series for ODBC Reference* for details.

---

## SQL Support

The driver supports the core SQL grammar.

---

## ODBC Conformance Level

The driver is Level 1 compliant, that is, it supports all ODBC Core and Level 1 functions

In addition, the following functions are supported:

- SQLColumnPrivileges
- SQLDescribeParam (if EnableDescribeParam=1)
- SQLForeignKeys
- SQLTablePrivileges

Refer to [Chapter 2 “ODBC API and Scalar Functions”](#) in the *DataDirect Connect Series for ODBC Reference* for a list of the API functions supported by the Greenplum Wire Protocol driver.

---

## Number of Connections and Statements Supported

The Greenplum Wire Protocol driver supports multiple connections and multiple statements per connection.

---

## Using Arrays of Parameters

Greenplum supports returning a set of output parameters or return values, but no ODBC standard method exists for returning arrays of output parameters or return values. If the call Escape is used to invoke a function that returns a set of output parameters and buffers are bound for those output parameters, the Greenplum Wire Protocol driver places the first set of output parameters in the bound buffers. If no output parameters are bound for functions that return a set of results or output parameters, the driver returns a result set with a row for each set of output parameters.

# 21 The Salesforce Driver

The DataDirect Connect XE *for* ODBC and DataDirect Connect64 XE *for* ODBC Salesforce driver (the Salesforce driver) supports the standard SQL query language to fetch, insert, update, and delete data from Salesforce.com, Force.com, and Database.com.

NOTE: For the Salesforce Web Service API versions supported by the Salesforce driver, refer to the product matrix on the Progress DataDirect Web site:

<http://www.datadirect.com/products/odbc/matrix/connectodbc.htm>

NOTE: You can query the SYSTEM\_REMOTE\_SESSIONS system table to get the version of the Web Service API the driver supports.

The driver translates the SQL statements provided by the application to Salesforce queries (SOQL) and Web service calls. Refer to [Chapter 10 “SQL Statements and Extensions for the Salesforce Driver”](#) in the *DataDirect Connect Series for ODBC Reference* for the SQL statements that the driver supports.

The driver maps the Salesforce data model into a set of related relational tables. The mapping representation is stored in XML files external to the driver. This allows the sharing of map files among different client machines.

The driver uses a client-side data cache for improved performance. You can define rules that specify which data to cache on the client as well as when the cached data becomes invalid and needs to be refreshed (see [“Client-Side Caches” on page 745](#) for details).

The Salesforce driver can be used with industry standard tools, which means that developers can leverage their existing SQL knowledge instead of having to learn the Salesforce query language and APIs. Examples include the following tools:

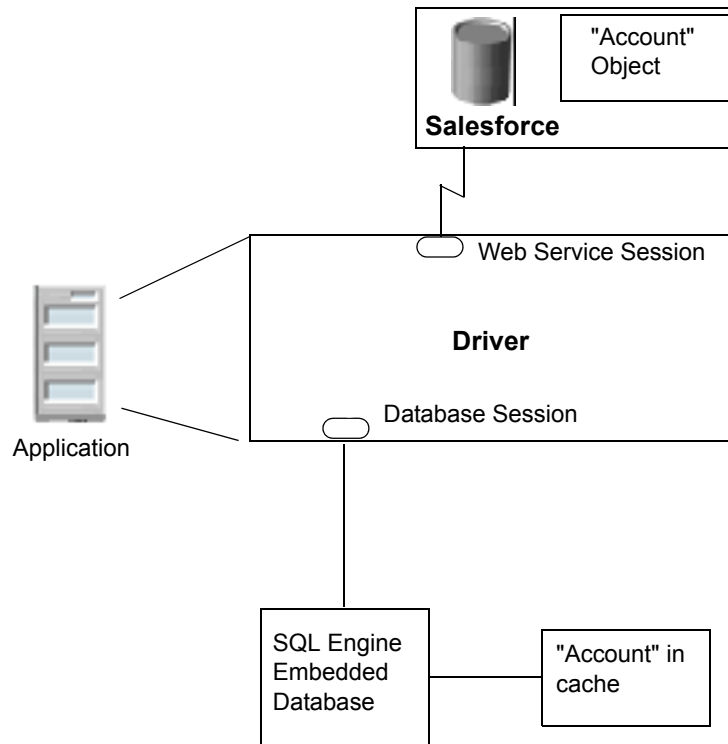
- Business Objects Crystal Reports
- Microsoft Access
- Oracle Gateway
- SAS/Access for ODBC
- Tableau
- Cognos
- Microsoft Excel
- Oracle Business Intelligence (OBIEE)
- SQL Server Linked Server

The driver is supported in the Windows, UNIX, and Linux environments. See [“Environment-Specific Information” on page 45](#) for detailed information about the environments supported by this driver.

Refer to the readme file shipped with your DataDirect Connect XE product for the file name of the driver.

Figure 21-1 shows the different components of an environment that uses the Salesforce driver to access Salesforce. Depending on your license from Salesforce.com, the driver could instead connect to Force.com or Database.com.

**Figure 21-1. Salesforce Driver**



When an application connects to Salesforce through the driver, connectivity to Salesforce is real-time, out of the box. In the background, the driver establishes a Web service session with the Salesforce instance and a database session that opens an embedded database. The application can establish multiple sessions with Salesforce; however, additional Web service and database sessions are always opened in the ratio of one database session per Web service session. The database session maintains the object-to-relational table mapping. In addition, it maintains cached tables and local tables, as well as maintaining views. See ["Database Configuration File" on page 755](#) for more details.

On Windows, the SQL Engine can be run within the same process space as the ODBC application, or it can be run as a separate process. Some applications may experience problems loading the JVM required for the SQL Engine because the process exceeds the available heap space. If your application experiences problems loading the JVM, you can configure the Salesforce driver to run in a separate 32-bit process within its own JVM. See ["Configuring the SQL Engine Server" on page 759](#) for more information.

Salesforce has certain standard objects that always exist, even if they do not contain anything. Salesforce administrators can also create custom objects using the Salesforce browser interface. The relationships among these objects are tabular, like those among the tables in a database. The Salesforce driver recognizes the relationships among both standard Salesforce objects and custom objects and can access, create, and update both. The relationships among objects can be reported through the ODBC SQLForeignKeys and SQLPrimaryKeys functions. The driver leverages Salesforce mechanisms for joining data, minimizing the amount of data that needs to be fetched over the network.

---

## Driver Requirements

The driver requires a Java Virtual Machine (JVM): J2SE 5 or higher.

---

## Configuring and Connecting to Data Sources

After you install the driver, you configure data sources to connect to the database. See [Chapter 1 “Quick Start Connect” on page 35](#) for an explanation of different types of data sources. The data source contains connection options that allow you to tune the driver for specific performance. If you want to use a data source but need to change some of its values, you can either modify the data source or override its values at connection time through a connection string.

If you choose to use a connection string, you must use specific connection string attributes. See ["Using a Connection String" on page 716](#) for an alphabetical list of driver connection string attributes and their initial default values.

### Data Source Configuration in the UNIX/Linux odbc.ini File

On UNIX and Linux, you must set up the proper ODBC environment before configuring data sources. See ["Setting the Library Path Environment Variable \(Salesforce Driver on UNIX/Linux\)" on page 38](#) for basic setup information and ["Environment Variables" on page 97](#) for more detailed information about this procedure.

Data sources for UNIX and Linux are stored in the system information file (by default, odbc.ini).

You can configure and modify data sources directly by editing the odbc.ini file and storing default connection values there. See ["Configuration Through the System Information \(odbc.ini\) File" on page 103](#) for detailed information about the specific steps necessary to configure a data source.

lists driver connection string attributes that must be used in the odbc.ini file to set the value of the attributes. Note that only the long name of the attribute can be used in the file. The default listed in the table is the initial default value when the driver is installed.

## Data Source Configuration through a GUI

On Windows, data sources are stored in the Windows Registry. You can configure and modify data sources through the ODBC Administrator using a driver Setup dialog box, as described in this section.

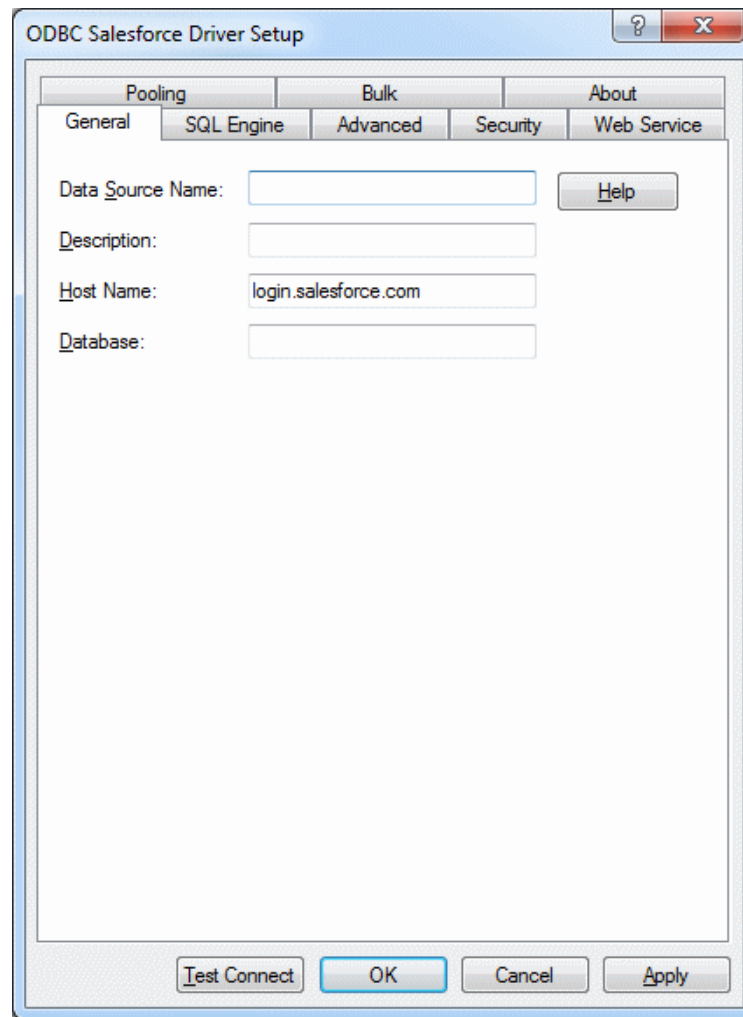
When the driver is first installed, the values of its connection options are set by default. These values appear on the driver Setup dialog box tabs when you create a new data source. You can change these default values by modifying the data source. In the following procedure, the description of each tab is followed by a table that lists the connection options for that tab and their initial default values. This table links you to a complete description of the options and their connection string attribute equivalents. The connection string attributes are used to override the default values of the data source if you want to change these values at connection time.

NOTE:

### To configure a Salesforce data source:

- 1 Start the ODBC Administrator by selecting its icon from the DataDirect Connect program group.
- 2 Select a tab:
  - **User DSN:** If you are configuring an existing user data source, select the data source name and click **Configure** to display the driver Setup dialog box.  
  
If you are configuring a new user data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.
  - **System DSN:** If you are configuring an existing system data source, select the data source name and click **Configure** to display the driver Setup dialog box.  
  
If you are configuring a new system data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.
  - **File DSN:** If you are configuring an existing file data source, select the data source file and click **Configure** to display the driver Setup dialog box.  
  
If you are configuring a new file data source, click **Add** to display a list of installed drivers; then, select a driver. Click **Advanced** if you want to specify attributes; otherwise, click **Next** to proceed. Specify a name for the data source and click **Next**. Verify the data source information; then, click **Finish** to display the driver Setup dialog box.

The General tab of the Setup dialog box appears by default.

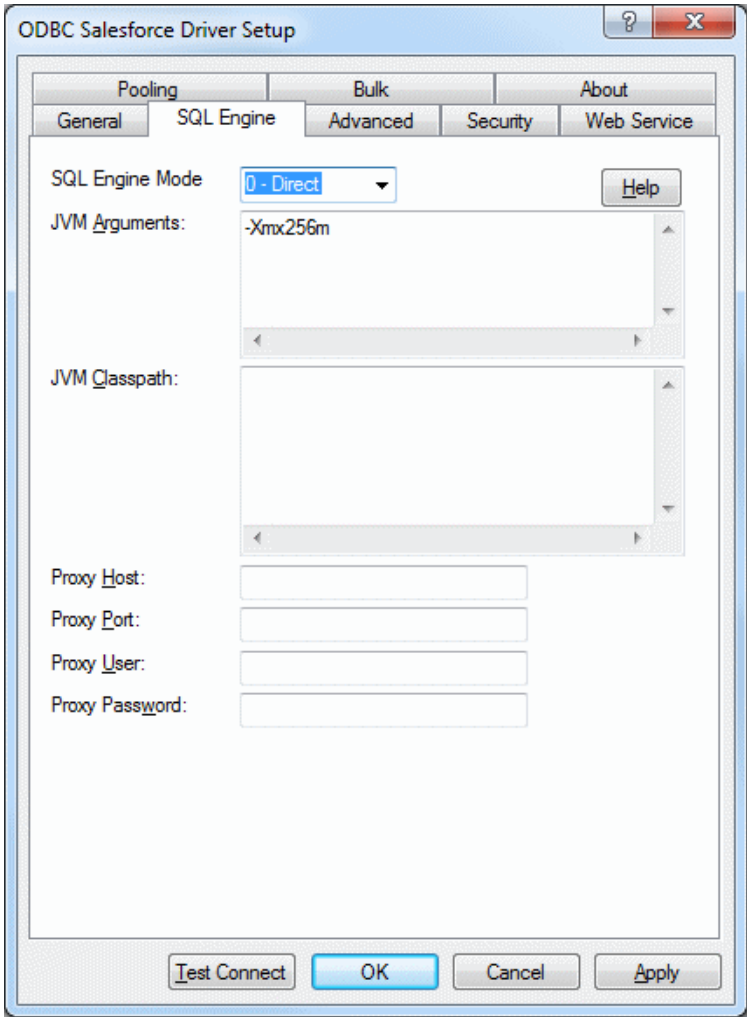


NOTE: The General tab displays only fields that are required for creating a data source. The fields on all other tabs are optional, unless noted otherwise.

On this tab, provide values for the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: General	Default
Data Source Name ( <a href="#">see page 726</a> )	None
Description ( <a href="#">see page 727</a> )	None
Host Name ( <a href="#">see page 729</a> )	login.salesforce.com
Database ( <a href="#">see page 727</a> )	None

- 3 Optionally, click the **SQL Engine** tab to specify additional data source settings.



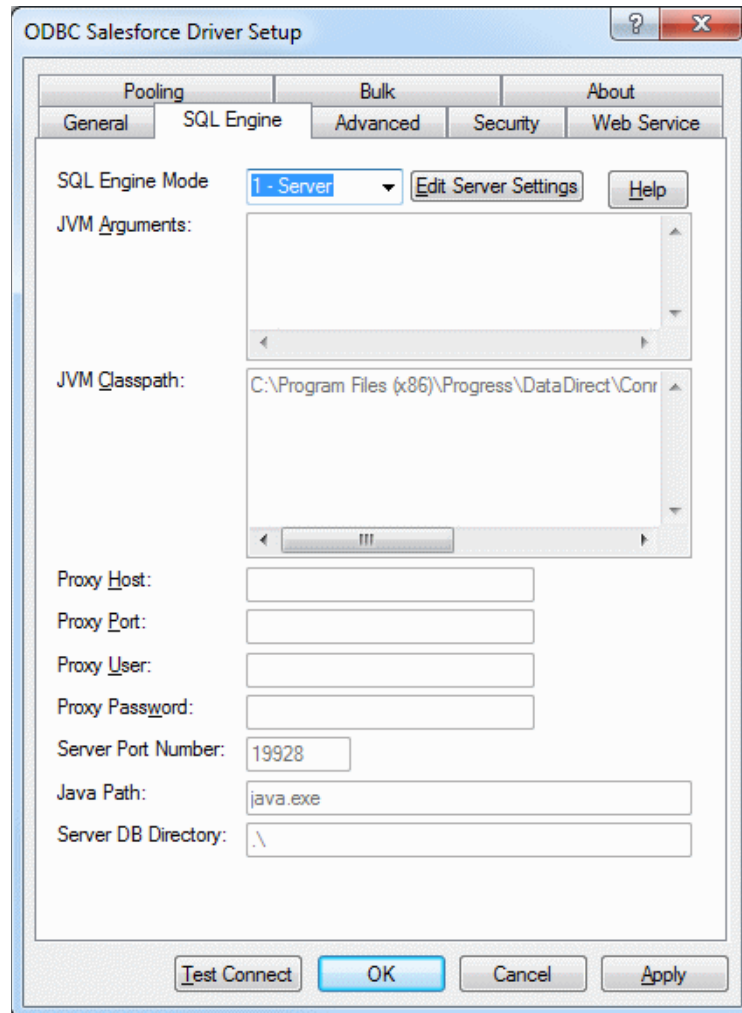
On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: SQL Engine	Default (Direct Mode)
<a href="#">JVM Arguments (see page 730)</a>	-Xmx256m
<a href="#">JVM Classpath (see page 731)</a>	None
<a href="#">Proxy Host (see page 734)</a>	None
<a href="#">Proxy Port (see page 735)</a>	None
<a href="#">Proxy User (see page 735)</a>	None
<a href="#">Proxy Password (see page 734)</a>	None

By default, the Salesforce driver operates in direct mode, with both the driver and its SQL engine running in the ODBC application's address space. Some applications may experience problems loading the JVM because the process exceeds the available heap space. You can configure the Salesforce driver to operate in server mode. Server mode allows the driver to connect to a 32-bit SQL engine JVM running as a separate service.



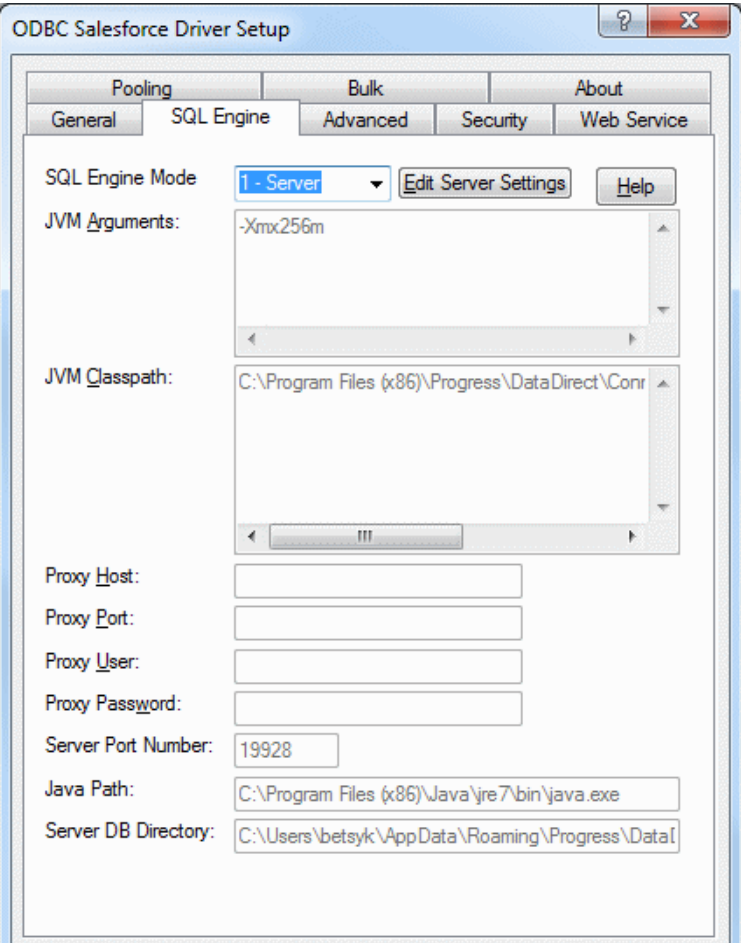
To run the driver in server mode, set **SQL Engine Mode** to 1-Server. Additional configuration settings appear on the SQL Engine tab. All fields except SQL Engine Mode are read only.



To define the settings for server mode, click **Edit Server Settings** from the SQL Engine tab. The SQL Engine Service Setup dialog box appears.

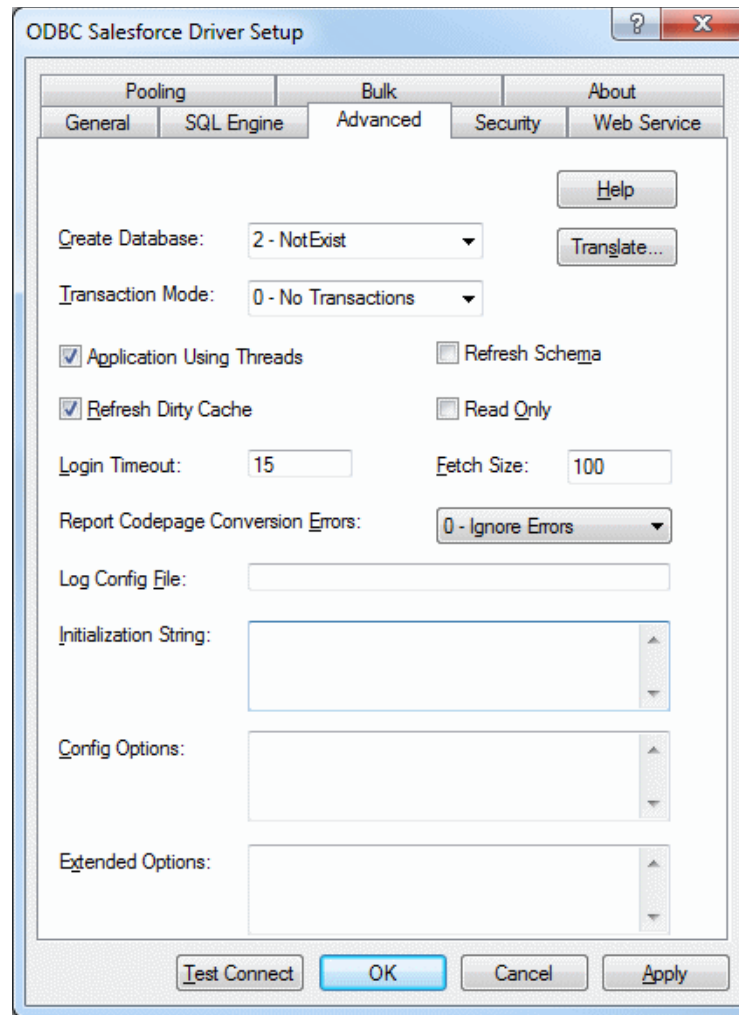
**NOTE:** You must be an administrator to modify the server mode settings. Otherwise, the Edit Server Settings button does not appear on the SQL Engine tab.

You use the SQL Engine Service Setup dialog box to configure server mode and to start or stop the service. See ["Configuring Server Mode" on page 759](#) for detailed information.



Configuration Options: SQL Engine Service	Default
<a href="#">JVM Arguments (see page 730)</a>	None
<a href="#">JVM Classpath (see page 731)</a>	<code>install_dir\java\lib\sforce.jar</code>
<a href="#">Proxy Host (see page 734)</a>	None
<a href="#">Proxy Port (see page 735)</a>	None
<a href="#">Proxy User (see page 735)</a>	None
<a href="#">Proxy Password (see page 734)</a>	None
<a href="#">Server Port Number (see page 738)</a>	19928
<a href="#">Java Path (see page 759)</a>	Fully qualified path to the J2SE 5 or higher JVM executable (java.exe)
<a href="#">Server DB Directory (see page 759)</a>	Path of the working directory for the SQL Engine service
<a href="#">Services (see page 759)</a>	Progress DataDirect Salesforce SQL Engine

- 4 Optionally, click the **Advanced** tab to specify additional data source settings.



On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

#### Connection Options: Advanced

Create Database (see page 726)

Transaction Mode (see page 739)

Application Using Threads (see page 719)

Refresh Dirty Cache (see page 736)

Refresh Schema (see page 736)

Read Only (see page 735)

Login Timeout (see page 732)

Fetch Size (see page 728)

Report Codepage Conversion Errors (see page 737)

Log Config File (see page 731)

Initialization String (see page 729)

Config Options (see page 722)

#### Default

2 -NotExist

0 - No Transactions

Enabled

Enabled

Disabled

Disabled

15

100

0 - Ignore Errors

None

None

None

**Extended Options:** Type a semi-colon separated list of connection options and their values. Use this configuration option to set the value of undocumented connection options that are provided by Progress DataDirect customer support. You can include any valid connection option in the Extended Options string, for example:

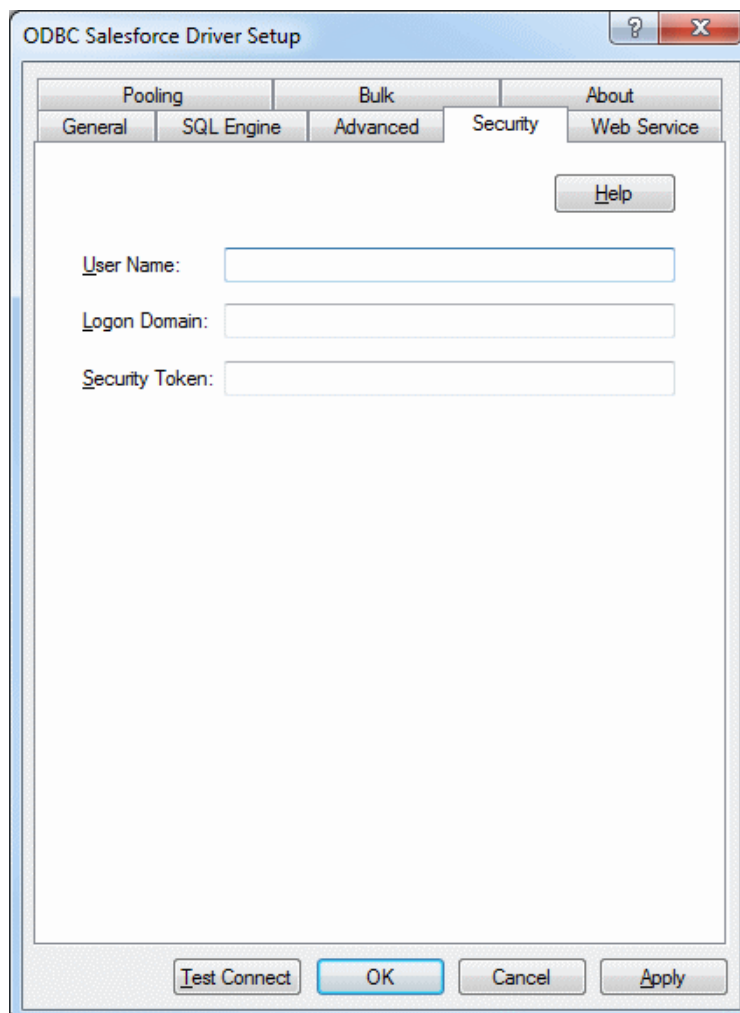
```
HostName=login.salesforce.com;UndocumentedOption1=value
[;UndocumentedOption2=value;]
```

If the Extended Options string contains option values that are also set in the setup dialog or data source, the values of the options specified in the Extended Options string take precedence. However, connection options that are specified on a connection string override any option value specified in the Extended Options string.

**Translate:** Click **Translate** to display the Select Translator dialog box, which lists the translators specified in the ODBC Translators section of the Registry. Progress 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.

- 5 Optionally, click the **Security** tab to specify security data source settings.



See ["Using Security" on page 77](#) for a general description of authentication and encryption and their configuration requirements.

On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Security	Default
<a href="#">User Name (see page 740)</a>	None
<a href="#">Logon Domain (see page 732)</a>	None
<a href="#">Security Token (see page 737)</a>	None

- 6 Optionally, click the **Web Service** tab to specify additional data source settings.

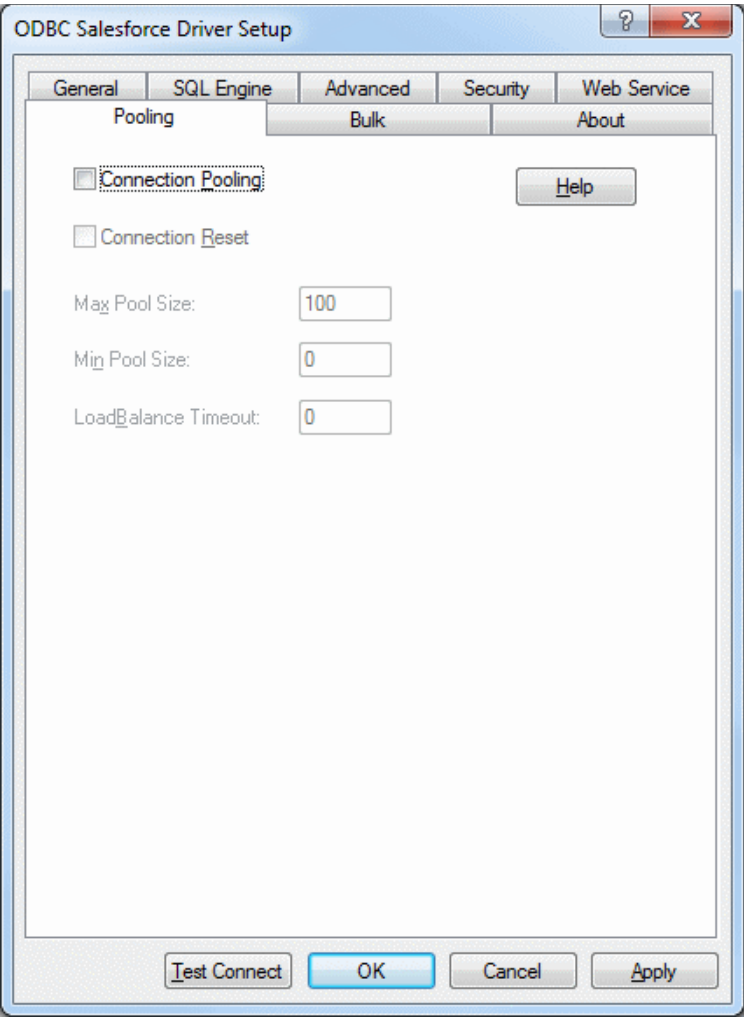
The screenshot shows the 'ODBC Salesforce Driver Setup' dialog box with the 'Web Service' tab selected. The dialog has a title bar with a question mark and a close button. Below the title bar are tabs for 'Pooling', 'Bulk', 'About', 'General', 'SQL Engine', 'Advanced', 'Security', and 'Web Service'. The 'Web Service' tab is active, showing a 'Help' button and several input fields: 'Statement Call Limit' (text box with '20'), 'Stmt Call Limit Behavior' (dropdown menu with '1 - ErrorAlways'), 'WSFetch Size' (text box with '0'), 'WSRetry Count' (text box with '0'), and 'WSTimeout' (text box with '120'). At the bottom are buttons for 'Test Connect', 'OK', 'Cancel', and 'Apply'.

On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Web Service	Default
<a href="#">Statement Call Limit (see page 739)</a>	20
<a href="#">Stmt Call Limit Behavior (see page 739)</a>	1- ErrorAlways
<a href="#">WSFetch Size (see page 740)</a>	0

Connection Options: Web Service	Default
<a href="#">WSRetry Count (see page 740)</a>	0
<a href="#">WSTimeout (see page 741)</a>	120

- Optionally, click the **Pooling** tab to specify connection pooling data source settings.

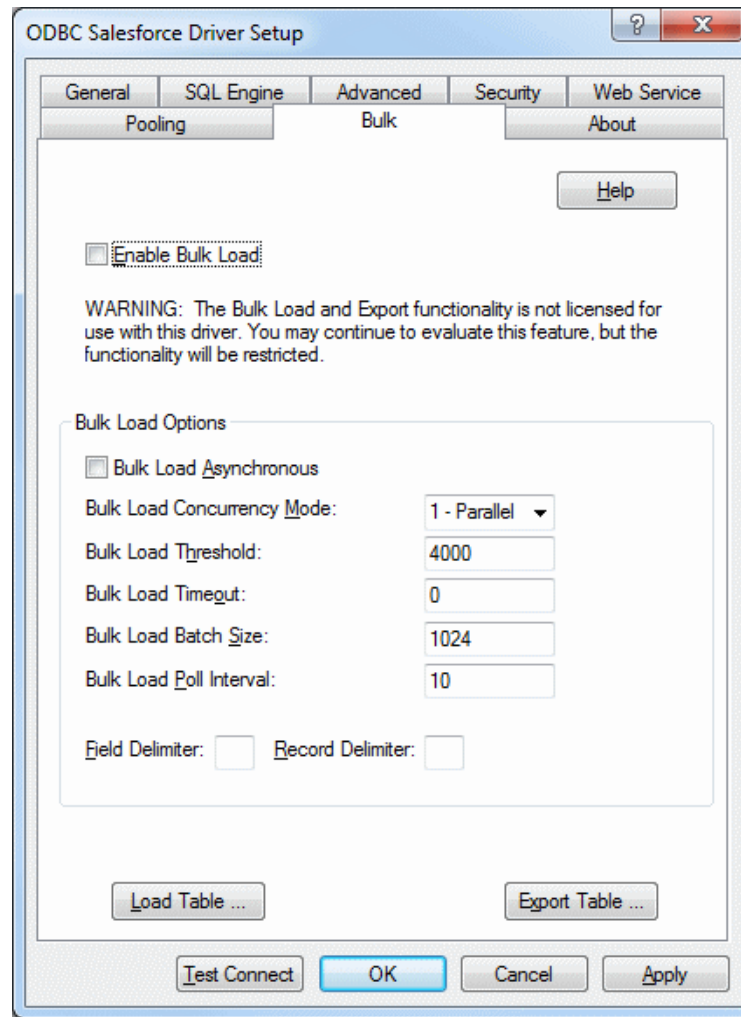


See ["Using DataDirect Connection Pooling" on page 82](#) for a general description of connection pooling.

On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Pooling	Default
<a href="#">Connection Pooling (see page 725)</a>	Disabled
<a href="#">Connection Reset (see page 725)</a>	Disabled
<a href="#">Max Pool Size (see page 733)</a>	100
<a href="#">Min Pool Size (see page 733)</a>	0
<a href="#">LoadBalance Timeout (see page 731)</a>	0

- 8 Optionally, click the **Bulk** tab to specify DataDirect Bulk Load data source settings.



See ["Using DataDirect Bulk Load" on page 85](#) for a general description of DataDirect Bulk Load.

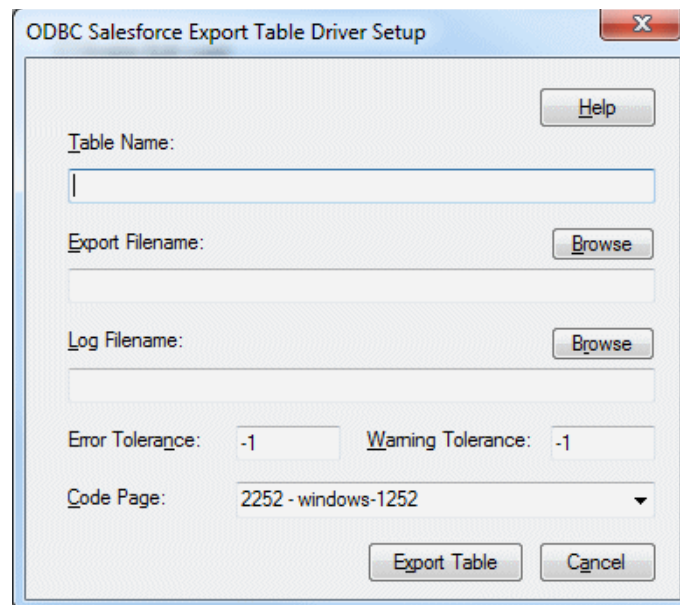
On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Bulk	Default
<a href="#">Enable Bulk Load (see page 728)</a>	Disabled
<a href="#">Bulk Load Asynchronous (see page 720)</a>	Disabled
<a href="#">Bulk Load Concurrency Mode (see page 720)</a>	1 (Parallel)
<a href="#">Bulk Load Threshold (see page 721)</a>	4000
<a href="#">Bulk Load Timeout (see page 722)</a>	0
<a href="#">Bulk Load Batch Size (see page 720)</a>	1024
<a href="#">Bulk Load Poll Interval (see page 721)</a>	10
<a href="#">Field Delimiter (see page 728)</a>	None
<a href="#">Record Delimiter (see page 735)</a>	None

If your application is already coded to use parameter array batch functionality, you can leverage DataDirect Bulk Load features through the Enable Bulk Load connection option. Enabling this option automatically converts the parameter array batch operation to use the database bulk load protocol.

If you are not using parameter array batch functionality, you can export data to a bulk load data file, verify the metadata of the bulk load configuration file against the structure of the target table, and bulk load data to a table. Use the following steps to accomplish these tasks.

- a To export data from a table to a bulk load data file, click **Export Table** from the Bulk tab. The Export Table dialog box appears.



Both a bulk data file and a bulk configuration file are produced by exporting a table. The configuration file has the same name as the data file, but with an XML extension. See ["Using DataDirect Bulk Load" on page 85](#) for details about these files.

The bulk export operation can create a log file and can also export to external files. See ["External Overflow Files" on page 95](#) for more information. The export operation can be configured such that if any errors or warnings occur:

- The operation always completes.
- The operation always terminates.
- The operation terminates after a certain threshold of warnings or errors is exceeded.

**Table Name:** A string that specifies the name of the source database table and, optionally, the columns containing the data to be exported. The driver uses the table name in the FROM clause of a `SELECT * FROM tablename` SQL statement. If you want to only export certain columns from your Salesforce table, then you can enter a SELECT statement in this field using the format:

```
(SELECT column1, column2, ... FROM tablename)
```



For example, to export data from the Salesforce ACCOUNT table excluding some of the audit columns, enter the following SQL in the Table Name field:

```
(SELECT SYS_NAME, TYPE, BILLINGSTREET, BILLINGCITY, BILLINGSTATE,
BILLINGPOSTALCODE, BILLINGCOUNTRY, SHIPPINGSTREET, SHIPPINGCITY,
SHIPPINGSTATE, SHIPPINGPOSTALCODE, SHIPPINGCOUNTRY, PHONE, FAX, WEBSITE,
INDUSTRY, ANNUALREVENUE, NUMBEROFEMPLOYEES, DESCRIPTION FROM ACCOUNT)
```

**Export Filename:** A string that specifies the path (relative or absolute) and file of the bulk load data file to which the data is to be exported. It also specifies the file name of the bulk configuration file. These files must not already exist; if one of both of them already exists, an error is returned.

**Log Filename:** A string that specifies the path (relative or absolute) and file name of the bulk log file. The log file is created if it does not exist. Events logged to this file are:

- Total number of rows fetched
- A message for each row that failed to export
- Total number of rows that failed to export
- Total number of rows successfully exported

Information about the load is written to this file, preceded by a header. Information about the next load is appended to the end of the file.

If you do not supply a value for Log Filename, no log file is created.

**Error Tolerance:** A value that specifies the number of errors to tolerate before an operation terminates. A value of 0 indicates that no errors are tolerated; the operation fails when the first error is encountered.

The default of -1 means that an infinite number of errors is tolerated.

**Warning Tolerance:** A value that specifies the number of warnings to tolerate before an operation terminates. A value of 0 indicates that no warnings are tolerated; the operation fails when the first warning is encountered.

The default of -1 means that an infinite number of warnings is tolerated.

**Code Page:** A value that specifies the code page value to which the driver must convert all data for storage in the bulk data file. See ["Character Set Conversions" on page 94](#) for more information.

The default value on Windows is the current code page of the machine.

Click **Export Table** to connect to the database and export data to the bulk data file or click **Cancel**.

- b** To bulk load data from the bulk data file to a database table, click **Load Table** from the Bulk tab. The Load File dialog box appears.

The load operation can create a log file and can also create a discard file that contains rows rejected during the load. The discard file is in the same format as the bulk load data file. After fixing reported issues in the discard file, the bulk load can be reissued using the discard file as the bulk load data file.

The export operation can be configured such that if any errors or warnings occur:

- The operation always completes.
- The operation always terminates.
- The operation terminates after a certain threshold of warnings or errors is exceeded.

If a load fails, the Load Start and Load Count options can be used to control which rows are loaded when a load is restarted after a failure.

**Table Name:** A string that specifies the name of the target database table and, optionally, the columns into which the data is loaded.

The fields defined in the load data file must have the same ordering of the fields defined in the Salesforce destination table. Because Salesforce defines additional audit columns that are managed by the database, your load data file may not contain data to load into these fields.

In this case, you can specify the exact columns that you want for the data to be inserted into using a Table Name string of the format:

```
table(column1, column2, ...)
```

For example, if your load data file contains only five fields of billing data that you wanted to load into the Salesforce ACCOUNT table, then the Table Name field would contain:

```
ACCOUNT(BILLINGSTREET, BILLINGCITY, BILLINGSTATE, BILLINGPOSTALCODE, BILLINGCOUNTRY)
```

**Load Data Filename:** A string that specifies the path (relative or absolute) and file name of the bulk data file from which the data is loaded.

**Configuration Filename:** A string that specifies the path (relative or absolute) and file name of the bulk configuration file.

**Log Filename:** A string that specifies the path (relative or absolute) and file name of the bulk log file. Specifying a value for Log Filename creates the file if it does not already exist. Events logged to this file are:

- Total number of rows read
- Message for each row that failed to load
- Total number of rows that failed to load
- Total number of rows successfully loaded

Information about the load is written to this file, preceded by a header. Information about the next load is appended to the end of the file.

If you do not specify a value for Log Filename, no log file is created.

**Discard Filename:** A string that specifies the path (relative or absolute) and file name of the bulk discard file. Any row that cannot be inserted into the database as result of bulk load is added to this file, with the last row rejected added to the end of the file.

Information about the load is written to this file, preceded by a header. Information about the next load is appended to the end of the file.

If you do not specify a value for Discard Filename, a discard file is not created.

**Error Tolerance:** A value that specifies the number of errors to tolerate before an operation terminates. A value of 0 indicates that no errors are tolerated; the operation fails when the first error is encountered.

The default of -1 means that an infinite number of errors is tolerated.

**Load Start:** A value that specifies the first row to be loaded from the data file. Rows are numbered starting with 1. For example, when Load Start is 10, the first 9 rows of the file are skipped and the first row loaded is row 10. This option can be used to restart a load after a failure.

The default value is 1.

**Read Buffer Size (KB):** A value that specifies the size, in KB, of the buffer that is used to read the bulk data file for a bulk load operation.

The default value is 2048.

**Warning Tolerance:** A value that specifies the number of warnings to tolerate before an operation terminates. A value of 0 indicates that no warnings are tolerated; the operation fails when the first warning is encountered.

The default of -1 means that an infinite number of warnings is tolerated.

**Load Count:** A value that specifies the number of rows to be loaded from the data file. The bulk load operation loads rows up to the value of Load Count from the file to the database. It is valid for Load Count to specify more rows than exist in the data file. The bulk load operation completes successfully when either the number of rows specified by the Load Count value has been loaded or the end of the data file is reached. This option can be used in conjunction with Load Start to restart a load after a failure.

The default value is the maximum value for SQLULEN. If set to 0, no rows are loaded.

Click **Load Table** to connect to the database and load the table or click **Cancel**.

At any point during the configuration process, you can click **Test Connect** to attempt to connect to the data source using the connection options specified in the driver Setup dialog box. A logon dialog box appears (see ["Using a Logon Dialog Box" on page 717](#) for details). 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 incorrect environment or connection value, it displays an appropriate error message. Click **OK**.
- 9 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.

## Using a Connection String

If you want to use a connection string for connecting to a database, or if your application requires it, you must specify either a DSN (data source name), a File DSN, or a DSN-less connection in the string. The difference is whether you use the `DSN=`, `FILEDSN=`, or the `DRIVER=` keyword in the connection string, as described in the ODBC specification. A DSN or FILEDSN connection string tells the driver where to find the default connection information. Optionally, you may specify *attribute=value* pairs in the connection string to override the default values stored in the data source.

The DSN connection string has the form:

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

The FILEDSN connection string has the form:

```
FILEDSN=filename.dsn[;attribute=value[;attribute=value]...]
```

The DSN-less connection string specifies a driver instead of a data source. All connection information must be entered in the connection string because the information is not stored in a data source.

The DSN-less connection string has the form:

```
DRIVER={[driver_name]}[;attribute=value[;attribute=value]...]
```

[Table 21-1 "Salesforce Attribute Names" on page 718](#) lists the long and short names for each attribute, as well as the initial default value when the driver is first installed. You can specify either long or short names in the connection string.

An example of a DSN connection string with overriding attribute values for Salesforce for Linux/UNIX/Windows is:

```
DSN=Salesforce;UID=JOHN@MYCOMPANY.COM;PWD=XYZZY
```

A FILEDSN connection string is similar except for the initial keyword:

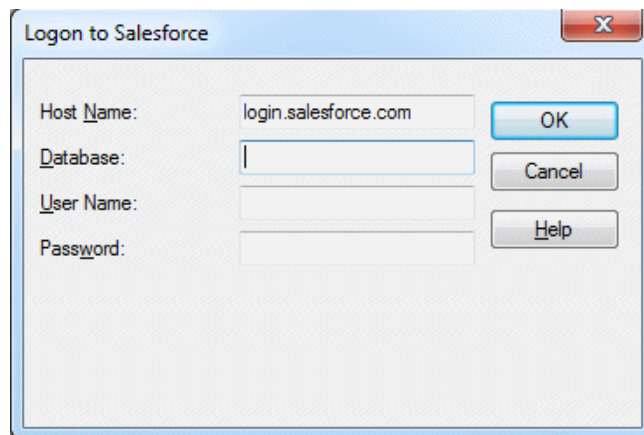
```
FILEDSN=Salesforce;UID=JOHN@MYCOMPANY.COM;PWD=XYZZY
```

A DSN-less connection string must provide all necessary connection information:

```
DRIVER=DataDirect 7.1 Salesforce;UID=JOHN@MYCOMPANY.COM;PWD=XYZZY
```

## 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 host name has already been specified.



In this dialog box, provide the following information:

- 1 In the Host Name field, type the root of the Salesforce URL to which you want to connect. The default is login.salesforce.com.
- 2 Type the file name prefix the driver uses to create or locate the set of files that define the embedded database per connection. See ["Mapping Objects to Tables" on page 745](#) for an explanation of embedded database.
- 3 Type your logon ID in the User Name field.

- 4 Type your password in the Password field.
- 5 Click **OK** to complete the logon.

# Connection Option Descriptions

The following connection option descriptions are listed alphabetically by the GUI name that appears on the driver Setup dialog box. The connection string attribute name, along with its short name, is listed immediately underneath the GUI name. For example:

## Application Using Threads

Attribute    ApplicationUsingThreads (AUT)

In most cases, the GUI name and the attribute name are the same; however, some exceptions exist. If you need to look up an option by its connection string attribute name, please refer to the alphabetical table of connection string attribute names.

Also, a few connection string attributes, for example, Password, do not have equivalent options that appear on the GUI. They are in the list of descriptions alphabetically by their attribute names.

The following table lists the connection string attributes supported by the Salesforce driver.

**Table 21-1. Salesforce Attribute Names**

Attribute (Short Name)	Default
ApplicationUsingThreads (AUT)	1 (Enabled)
BulkLoadAsync (BLA)	0 (unselected)
BulkLoadBatchSize (BLBS)	1024
BulkLoadConcurrencyMode (BLCM)	1 (Parallel)
BulkLoadFieldDelimiter (BLFD)	None
BulkLoadPollInterval (BLPI)	10
BulkLoadRecordDelimiter (BLRD)	None
BulkLoadThreshold (BLTH)	4000
BulkLoadTimeout (BLTO)	0
ConfigOptions (CO)	Empty string
ConnectionReset (CR)	0 (Disabled)
CreateDB (CDB)	2 (NotExist)
DataSourceName (DSN)	None
Database	The file name prefix the driver uses to create or locate the set of files that define the embedded database per connection.
EnableBulkLoad (EBL)	0 (Disabled)
FetchSize (FS)	100

**Table 21-1. Salesforce Attribute Names** (cont.)

Attribute (Short Name)	Default
HostName (HOST)	Empty string (logs into the default Salesforce instance, login.salesforce.com).
InitializationString (IS)	Empty string
JVMArgs (JVMA)	-Xmx256m
JVMClasspath (JVMC)	<i>install_dir\java\lib\sforce.jar</i>
LogConfigFile (LCF)	Empty string
LoginTimeout (LT)	15
LogonDomain (LD)	Empty string
LogonID (UID)	None
MaxPoolSize (MXPS)	100
MinPoolSize (MNPS)	0
Pooling (POOL)	0 (Disabled)
ProxyHost (PXHN)	Empty string
ProxyPassword (PXPW)	Empty string
ProxyPort (PXPT)	Empty string
ProxyUser (PXUN)	Empty string
ReadOnly (RO)	0
RefreshDirtyCache (RDC)	1 (Enabled)
RefreshSchema (RS)	0 (Disabled)
ReportCodepageConversionErrors (RCCE)	0 (Ignore Errors)
SecurityToken (STK)	Empty string
ServerPortNumber (SPN)	19928
SQLEngineMode (SEM)	0 (Direct)
StmtCallLimit (SCL)	20
StmtCallLimitBehavior (SCLB)	1 (ErrorAlways)
TransactionMode (TM)	0 (No Transactions)
WSFetchSize (WSFS)	0
WSRetryCount (WSRC)	0
WSTimeout (WST)	120

## Application Using Threads

Attribute	ApplicationUsingThreads (AUT)
Description	Determines whether the driver works with applications using multiple ODBC threads.  This connection option can affect performance. See <a href="#">"Performance Considerations" on page 741</a> for details.

Valid Values 0 | 1

If set to 1 (Enabled), the driver works with single-threaded and multi-threaded applications.

If set to 0 (Disabled), the driver does not work with multi-threaded applications. If using the driver with single-threaded applications, this value avoids additional processing required for ODBC thread-safety standards.

Default 1 (Enabled)

GUI Tab [Advanced tab](#) on page 707

## Bulk Load Asynchronous

Attribute BulkLoadAsync (BLA)

Description Determines whether the driver treats bulk load operations as synchronous or asynchronous.

Valid Values 0 | 1

If set to 0 (Disabled), bulk load operations are synchronous. The driver does not return from the function that invoked an operation until the operation is complete or the BulkLoadTimeout period has expired. If the operation times out, the driver returns an error.

If set to 1 (Enabled), bulk load operations are asynchronous. The driver returns from the function that invoked an operation after the operation is submitted to the server. The driver does not verify the completion status of the bulk load operation.

Default 0 (Disabled)

GUI Tab [Bulk tab](#) on page 711

## Bulk Load Batch Size

Attribute BulkLoadBatchSize (BLBS)

Description The number of rows that the driver sends to the database at a time during bulk operations. This value applies to all methods of bulk loading.

Valid Values  $x$ 

where  $x$  is a positive integer that specifies the number of rows to be sent.

Default 1024

GUI Tab [Bulk tab](#) on page 711

## Bulk Load Concurrency Mode

Attribute BulkLoadConcurrencyMode (BLCM)

Description Determines whether multiple batches associated with a bulk load operation are processed by Salesforce in parallel or one at a time. See ["Using DataDirect Bulk Load" on page 85](#) for more information.



Valid Values	0   1
	If set to 0 (Serial), multiple batches associated with a bulk load operation are processed one at a time.
	If set to 1 (Parallel), multiple batches associated with a bulk load operation are processed in parallel. The order in which the batches are processed can vary.
Default	1 (Parallel)
GUI Tab	<a href="#">Bulk tab</a> on page 711

## Bulk Load Poll Interval

Attribute	BulkLoadPollInterval (BLPI)
Description	Specifies the number of seconds the driver waits to request bulk operation status. This interval is used by the driver the first time it requests status and for all subsequent status requests. See <a href="#">"Using DataDirect Bulk Load" on page 85</a> for more information.
Valid Values	$x$
	where $x$ is a positive integer that represents the number of seconds the driver waits before requesting bulk operation status.
Default	10
GUI Tab	<a href="#">Bulk tab</a> on page 711

## Bulk Load Threshold

Attribute	BulkLoadThreshold (BLTH)
Description	Determines when the driver uses bulk load for insert, update, delete, or batch operations. If the <a href="#">Enable Bulk Load</a> option is set to <code>True</code> and the number of rows affected by an insert, update, delete, or batch operation exceeds the threshold specified by this option, the driver uses the Salesforce Bulk API to perform the operation.
	NOTES:
	<ul style="list-style-type: none"> <li>■ If the <a href="#">Enable Bulk Load</a> option is set to <code>false</code>, this option is ignored.</li> <li>■ Do not set the Bulk Load Threshold option to a value greater than the Web service call limit set by the <a href="#">Statement Call Limit</a> option. If the value set for Bulk Load Threshold is greater than the value of Statement Call Limit, the driver would never use the Salesforce Bulk API because the Web service call limit is reached before the driver reaches the threshold to switch to the Salesforce Bulk API.</li> </ul>
Valid Values	0   $x$
	where $x$ is a positive integer that represents a threshold (number of rows).
	If set to 0, the driver always uses bulk load to execute insert, update, delete, or batch operations.
	If set to $x$ , the driver only uses bulk load if the <a href="#">Enable Bulk Load</a> option is set to a value of <code>True</code> and the number of rows to be updated by an insert, update, delete, or batch operation exceeds the threshold. If the operation times out, the driver returns an error.

Default 4000

GUI Tab [Bulk tab](#) on page 711

## Bulk Load Timeout

Attribute BulkLoadTimeout (BLTO)

Description The time, in seconds, that the driver waits for a Salesforce bulk job to complete. A value of zero means there is no timeout.

Valid Values  $x$

where  $x$  is a positive integer that represents a number of seconds the driver waits before requesting bulk operation status.

Default 0

GUI Tab [Bulk tab](#) on page 711

## Config Options

Attribute ConfigOptions (CO)

Description Determines how the embedded database and the mapping of the remote data model to the relational data model is configured, customized, and updated.

NOTE: This option is primarily used for initial configuration of the driver for a particular user. It is not intended for use with every connection. By default, the driver configures itself and this option is normally not needed. If Config Options is specified on a connection after the initial configuration, the values specified for Config Options must match the values specified for the initial configuration. The preferred method for setting the configuration options for a particular user is through the database configuration file. See ["Database Configuration File" on page 755](#) for details.

Valid Values `(key=value[; key=value])`

where *key* is one of the following values: AuditColumns, CustomSuffix, MapSystemColumnNames, NumberFieldMapping, or UppercaseIdentifiers.

The value is a set of key value pairs separated by a semicolon (;). The value must be enclosed in parentheses. For example:

```
(AuditColumns=AuditOnly;UppercaseIdentifiers=false)
```

**AuditColumns:** Determines whether the driver includes audit fields, which Salesforce adds to all objects defined in a Salesforce instance, as table columns when it defines the remote data model to relational table mapping.

The audit columns added by Salesforce are:

```
IsDeleted
CreatedById
CreatedDate
LastModifiedById
LastModifiedDate
SystemModestamp
```

Salesforce also adds the field MasterRecordId.

Valid values for AuditColumns are:

Value	Description
All	The driver includes the all of the audit columns and the master record id column in its table definitions.
AuditOnly	The driver adds only the audit columns in its table definitions.
MasterOnly	The driver adds only the MasterRecordId column in its table definitions.
None	The driver does not add the audit columns or the MasterRecordId column in its table definitions.

The default value for AuditColumns is `None`.

In a typical Salesforce instance, not all users are granted access to the Audit or MasterRecordId columns. If AuditColumns is set to a value other than `None` and the driver cannot include the columns requested, the connection fails and the driver generates a `SQLException` with a `SQLState` of 08001.

**CustomSuffix** (Custom objects and fields only): Determines whether the driver includes or strips the "\_\_c" suffix from the table and column names when mapping the remote data model to the relational data model. Salesforce adds the suffix to all custom objects and fields.

Valid values for CustomSuffix are:

Value	Description
Include	The driver includes the "__c" suffix.
Strip	The driver strips the "__c" suffix.

The default value for CustomSuffix is `Strip`.

**KeywordConflictSuffix**: Specifies a string of up to five alphanumeric characters that the driver appends to any object or field name that conflicts with a SQL engine keyword. For example, if you specify `KeywordConflictSuffix=TAB`, then the driver maps the Case object in Salesforce to CASETAB.

Do not use a string that matches the suffix of a custom table, for example, CASEOFFICE. If you specify `KeywordConflictSuffix=OFFICE`, a name collision occurs with the Standard object CASE and the custom table CASEOFFICE, or a table with a column called CASEOFFICE. In this situation, the standard object CASE is returned. The custom object is ignored.

Valid values for KeywordConflictSuffix are:

Value	Description
string	One to five alphanumeric characters.

The default value for KeywordConflictSuffix is an empty string.

**MapSystemColumnNames:** Determines how the driver maps Salesforce system columns. Valid values for MapSystemColumnNames are:

Value	Description																						
0	The driver does not change the names of the Salesforce system columns.																						
1	The driver changes the names of the Salesforce system columns as described in the following table:																						
	<table><tr><th>Field Name</th><th>Mapped Name</th></tr><tr><td>Id</td><td>ROWID</td></tr><tr><td>Name</td><td>SYS_NAME</td></tr><tr><td>IsDeleted</td><td>SYS_ISDELETED</td></tr><tr><td>CreatedDate</td><td>SYS_CREATEDDATE</td></tr><tr><td>CreatedById</td><td>SYS_CREATEDBYID</td></tr><tr><td>LastModifiedDate</td><td>SYS_LASTMODIFIEDDATE</td></tr><tr><td>LastModifiedId</td><td>SYS_LASTMODIFIEDID</td></tr><tr><td>SystemModstamp</td><td>SYS_SYSTEMMODSTAMP</td></tr><tr><td>LastActivityDate</td><td>SYS_LASTACTIVITYDATE</td></tr><tr><td>OwnerId</td><td>SYS_OWNERID</td></tr></table>	Field Name	Mapped Name	Id	ROWID	Name	SYS_NAME	IsDeleted	SYS_ISDELETED	CreatedDate	SYS_CREATEDDATE	CreatedById	SYS_CREATEDBYID	LastModifiedDate	SYS_LASTMODIFIEDDATE	LastModifiedId	SYS_LASTMODIFIEDID	SystemModstamp	SYS_SYSTEMMODSTAMP	LastActivityDate	SYS_LASTACTIVITYDATE	OwnerId	SYS_OWNERID
Field Name	Mapped Name																						
Id	ROWID																						
Name	SYS_NAME																						
IsDeleted	SYS_ISDELETED																						
CreatedDate	SYS_CREATEDDATE																						
CreatedById	SYS_CREATEDBYID																						
LastModifiedDate	SYS_LASTMODIFIEDDATE																						
LastModifiedId	SYS_LASTMODIFIEDID																						
SystemModstamp	SYS_SYSTEMMODSTAMP																						
LastActivityDate	SYS_LASTACTIVITYDATE																						
OwnerId	SYS_OWNERID																						

The default value for MapSystemColumnNames is 1.

**NumberFieldMapping:** Defines how the driver maps fields defined as NUMBER in Salesforce. The Salesforce API uses DOUBLE values to transfer data to and from NUMBER fields, which can cause problems when the precision of the NUMBER field is greater than the precision of a DOUBLE value. Rounding can occur when converting large values to and from DOUBLE. By default, the driver maps smaller fields (precision of 9 or less) to the INTEGER SQL type when the scale of the NUMBER field is 0 and maps all other NUMBER fields to the DOUBLE SQL type to match the type that Salesforce transfers the value to or from the driver. This key can be used to direct the driver to map all NUMBER fields to DOUBLE regardless of the precision of the field.

Valid values for NumberFieldMapping are:

Value	Description
1	The driver maps NUMBER fields with a precision of 9 or less and a scale of 0 to the INTEGER SQL type and maps all other NUMBER fields to the DOUBLE SQL type.
2	The driver maps NUMBER fields to the DOUBLE SQL type.

The default value for NumberFieldMapping is 1.

**UppercaseIdentifiers:** Defines how the driver maps identifiers. By default, the driver maps all identifier names to uppercase.

NOTE: Do not change the value of UppercaseIdentifiers unless the data source you are connecting to has objects with names that differ only by case.

Valid values for UppercaseIdentifiers are:

Value	Description
true	The driver maps identifiers to uppercase.
false	The driver maps identifiers to the mixed case name of the object being mapped. If mixed case identifiers are used, SQL statements must enclose those identifiers in double quotes, and the case of the identifier, must exactly match the case of the identifier name.  For example, if UppercaseIdentifiers=false, to query the Account table you specify:  <pre>SELECT "id", "name" FROM "Account"</pre>

The default value for UppercaseIdentifiers is `true`.

Default `AuditColumns=none;CustomSuffix=strip;KeywordConflictSuffix=;  
MapSystemColumnNames=1;NumberFieldMapping=1;UppercaseIdentifiers=true;`

GUI Tab [Advanced tab](#) on page 707

## Connection Pooling

Attribute Pooling (POOL)

Description Specifies whether to use the driver's connection pooling.

NOTE: The application must be thread-enabled to use connection pooling.

This connection option can affect performance. See ["Performance Considerations" on page 741](#) for details.

Valid Values 0 | 1

If set to 1 (Enabled), the driver uses connection pooling.

If set to 0 (Disabled), the driver does not use connection pooling.

Default 0 (Disabled)

GUI Tab [Pooling tab](#) on page 710

## Connection Reset

Attribute ConnectionReset (CR)

Description Determines whether the state of connections that are removed from the connection pool for reuse by the application is reset to the initial configuration of the connection.

This connection option can affect performance. See ["Performance Considerations" on page 741](#) for details.

Valid Values 0 | 1

If set to 1 (Enabled), the state of connections removed from the connection pool for reuse by an application is reset to the initial configuration of the connection. Resetting the state can negatively impact performance because additional commands must be sent over the network to the server to reset the state of the connection.

If set to 0 (Disabled), the state of connections is not reset.

Default 0 (Disabled)

GUI Tab [Pooling tab](#) on page 710

## Create Database

Attribute CreateDB (CDB)

Description Determines whether the driver creates a new embedded database when establishing the connection.

Valid Values 0 | 1 | 2

If set to 0 (No), the driver uses the current embedded database specified by `DatabaseName`. If one does not exist, the connection fails.

If set to 1 (ForceNew), the driver deletes the current embedded database specified by [Database](#) and creates a new one at the same location.

WARNING: This causes all views, data caches, and map customizations defined in the current database to be lost.

If set to 2 (NotExist), the driver uses the current embedded database specified by `DatabaseName`. If one does not exist, the driver creates one.

Default 2 (NotExist)

GUI Tab [Advanced tab](#) on page 711

## Data Source Name

Attribute DataSourceName (DSN)

Description The name of a data source in your Windows Registry or `odbc.ini` file.

Valid Values *string*

where *string* is the name of a data source.

Default None

GUI Tab [General tab](#) on page 703

## Database

Attribute	Database (DBN)
Description	Specifies the file name prefix the driver uses to create or locate the set of files that define the Object mapping and the embedded database used by the connection. See <a href="#">"Mapping Objects to Tables" on page 745</a> for an explanation of embedded database.
Valid Values	<i>prefix</i>   <i>path+prefix</i>  where:  <i>prefix</i> is the file name prefix for the embedded database. For example, if Database is set to a value of JohnQPublic, the embedded database files that are created or loaded have the form johnqpublic.xxx.  <i>path+prefix</i> is a relative or absolute path appended to the file name prefix. The path defines the directory the driver uses to store the newly created database files or locate the existing database files. For example, if Database is set to a value of C:\data\db\johnqpublic, the driver either creates or looks for the database johnqpublic.xxx in the directory C:\data\db. If you do not specify a path, the current working directory is used.  NOTES:  <ul style="list-style-type: none"> <li>■ The driver parses the User ID value and removes all non-alphanumeric characters. For example, if User ID is specified as John.Q.Public, the value used for Database is JohnQPublic.</li> <li>■ When SQL Engine Mode is set to Server, the <i>path+prefix</i> value overrides the value specified by the Server DB Directory configuration setting (see <a href="#">"Configuring Server Mode" on page 759</a>).</li> </ul>
Default	The user ID specified for the connection.
GUI Tab	<a href="#">General tab</a> on page 703

## Description

Attribute	Description (n/a)
Description	An optional long description of a data source. This description is not used as a runtime connection attribute, but does appear in the ODBC.INI section of the Registry and in the odbc.ini file.
Valid Values	<i>string</i>  where <i>string</i> is a description of a data source.
Default	None
GUI Tab	<a href="#">General tab</a> on page 703

## Enable Bulk Load

Attribute	EnableBulkLoad (EBL)
Description	Specifies whether the driver can use the bulk load protocol for insert, update, delete, and batch operations. Bulk load can reduce the number of Web service calls used to execute a statement when compared to statements that are executed individually and may improve performance. Whether the driver actually uses bulk load is determined by the <a href="#">Bulk Load Threshold</a> connection option.
Valid Values	True   False  If set to <code>True</code> , the driver can use the bulk load protocol for insert, update, delete, and batch operations.  If set to <code>False</code> , the driver cannot use the bulk load protocol for insert, update, delete, and batch operations.
Default	False
GUI Tab	<a href="#">Bulk tab</a> on page 710

## Fetch Size

Attribute	FetchSize (FS)
Description	The number of rows that the driver processes before returning data to the application. Smaller fetch sizes can improve the initial response time of the query. Larger fetch sizes improve overall fetch times at the cost of additional memory.  FetchSize is related to, but different from, <a href="#">WSFetch Size</a> . WS Fetch Size specifies the number of rows of raw data that the driver fetches from the remote data source, while Fetch Size specifies how many of these raw data rows the driver processes before returning data to the application. Processing the data includes converting from the remote data source data type to the driver SQL data type used by the application. If Fetch Size is greater than WS Fetch Size, the driver makes multiple round trips to the data source to get the requested number of rows before returning control to the application.
Valid Values	0   <i>x</i>  where <i>x</i> is a positive integer that specifies the number of rows that the driver processes before returning data to the application.  If set to 0, the driver fetches and processes all of the rows of the result before returning control to the application.
Default	100
GUI Tab	<a href="#">Advanced tab</a> on page 711

## Field Delimiter

Attribute	BulkLoadFieldDelimiter (BLFD)
Description	Specifies the character that the driver will use to delimit the field entries in a bulk load data file.
Valid Values	<i>x</i>



where *x* is any printable character.

For simplicity, avoid using a value that can be in the data, including all alphanumeric characters, the dash(-), the colon(:), the period (.), the forward slash (/), the space character, the single quote (') and the double quote ("). You can use some of these characters as delimiters if all of the data in the file is contained within double quotes.

NOTE: The Bulk Load Field Delimiter character must be different from the Bulk Load Record Delimiter.

Default None  
GUI Tab [Bulk tab](#) on page 711

## Host Name

Attribute HostName (HOST)  
Description The base Salesforce URL to use for logging in. If you are logging into a Salesforce instance other than the default, you must provide the root of the Salesforce URL.

Valid Values *url*

where:

*url* is the is the root of the Salesforce URL to which you want to connect.

Example Suppose you have a Salesforce instance that is configured with a production instance and a sandbox instance. You can specify `login.salesforce.com` as the value for the HostName attribute to connect to the production instance or `test.salesforce.com` to connect to the sandbox instance:

Salesforce Instance	URL
Production	<code>login.salesforce.com</code>
Sandbox	<code>test.salesforce.com</code>

Default None if you use the default Salesforce URL, `login.salesforce.com`  
GUI Tab [General tab](#) on page 703

## Initialization String

Attribute InitializationString (IS)  
Description One or multiple SQL commands to be executed by the driver after it has established the connection to the database and has performed all initialization for the connection. If the execution of a SQL command fails, the connection attempt also fails and the driver returns an error indicating which SQL command or commands failed.

Valid Values *string*

where *string* is one or multiple SQL commands.

Multiple commands must be separated by semicolons. In addition, if this option is specified in a connection URL, the entire value must be enclosed in parentheses when multiple commands are specified.

**Example** Because fetching metadata and generating mapping files can significantly increase the time it takes to connect to Salesforce, the driver caches this information on the client the first time the driver connects on behalf of each user. The cached metadata is used in subsequent connections made by the user instead of re-fetching the metadata from Salesforce. To force the driver to re-fetch the metadata information for a connection, use the `InitializationString` property to pass the `REFRESH SCHEMA SFORCE` command in the connection URL. For example:

```
DSN=Salesforce;UID={test@abccorp.com};PWD=secret;InitializationString=(REFRESH
SCHEMA SFORCE)
```

**Default** None

**GUI Tab** [Advanced tab](#) on page 707

## JVM Arguments

**Attribute** `JVMArgs (JVMA)`

**Description** A string that contains the arguments that are passed to the JVM that the driver is starting. The location of the JVM must be specified on the driver library path. For information on setting the location of the JVM in your environment, see:

- ["Setting the Library Path Environment Variable \(Salesforce Driver on Windows\)" on page 35](#)
- ["Setting the Library Path Environment Variable \(Salesforce Driver on UNIX/Linux\)" on page 38](#).

When specifying the heap size for the JVM, note that the JVM tries to allocate the heap memory as a single contiguous range of addresses in the application's memory address space. If the application's address space is fragmented so that there is no contiguous range of addresses big enough for the amount of memory specified for the JVM, the driver fails to load, because the JVM cannot allocate its heap. This situation is typically encountered only with 32-bit applications, which have a much smaller application address space. If you encounter problems with loading the driver in an application, try reducing the amount of memory requested for the JVM heap. If possible, switch to a 64-bit version of the application.

**Valid Values** *string*

where the string contains arguments that are defined by the JVM. Values that include special characters or spaces must be enclosed in curly braces { } when used in a connection string.

**Examples** To set the heap size used by the JVM to 256 MB and the http proxy information, specify:

```
{-Xmx256m -Dhttp.proxyHost=johndoe -Dhttp.proxyPort=808}
```

To set the heap size to 256 MB and configure the JVM for remote debugging, specify:

```
{-Xmx256m
-Xrunjdwp:transport=dt_socket, address=9003,server=y,suspend=n -Xdebug}
```

**Default** `-Xmx256m`

**GUI Tab** [SQL Engine tab](#) on page 704

## JVM Classpath

Attribute	JVMClasspath (JVMC)
Description	Specifies the CLASSPATH for the Java Virtual Machine (JVM) used by the driver. The CLASSPATH is the search string the JVM uses to locate the Java jar files the driver needs.
Valid Values	<p><i>string</i></p> <p>where the string specifies the CLASSPATH. Separate multiple jar files by a semi-colon on Windows platforms and by a colon on Linux and UNIX platforms. CLASSPATH values with multiple jar files must be enclosed in curly braces { } when used in a connection string.</p>
Example	<p>On Windows:</p> <pre>{.;c:\install_dir\java\lib\}</pre> <p>On UNIX:</p> <pre>{./home/user1/install_dir/java/lib/sforce.jar}</pre>
Default	<i>install_dir\java\lib\sforce.jar</i>
GUI Tab	<a href="#">SQL Engine tab</a> on page 704

## LoadBalance Timeout

Attribute	LoadBalanceTimeout (LBT)
Description	<p>The number of seconds to keep inactive connections open in a connection pool. An inactive connection is a database session that is not associated with an ODBC connection handle, that is, a connection in the pool that is not in use by an application.</p> <p>NOTE: The Min Pool Size option may cause some connections to ignore this value.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 741</a> for details.</p>
Valid Values	<p>0   <i>x</i></p> <p>where <i>x</i> is a positive integer that specifies a number of seconds.</p> <p>If set to 0, inactive connections are kept open.</p> <p>If set to <i>x</i>, inactive connections are closed after the specified number of seconds passes.</p>
Default	0
GUI Tab	<a href="#">Pooling tab</a> on page 710

## Log Config File

Attribute	LogConfigFile (LCF)
Description	<p>Specifies the filename of the configuration file used to initialize the driver logging mechanism.</p> <p>If the driver cannot locate the specified file when establishing the connection, the connection fails and the driver returns an error.</p>

Valid Values *string*

where *string* is the relative or fully qualified path of the configuration file used to initialize the driver logging mechanism. If the specified file does not exist, the driver continues searching for an appropriate configuration file as described in "Using Logging" in the *DataDirect Connect Series for ODBC Reference*.

Default Empty string

GUI Tab [Advanced tab](#) on page 710

## Login Timeout

Attribute LoginTimeout (LT)

Description The number of seconds the driver waits for a connection to be established before returning control to the application and generating a timeout error. To override the value that is set by this connection option for an individual connection, set a different value in the SQL\_ATTR\_LOGIN\_TIMEOUT connection attribute using the SQLSetConnectAttr() function.

Valid Values 0 | *x*

where *x* is a positive integer that specifies a number of seconds.

If set to 0, inactive connections are kept open.

If set to *x*, inactive connections are closed after the specified number of seconds passes.

Default 15

GUI Tab [Advanced tab](#) on page 710

## Logon Domain

Attribute LogonDomain (LD)

Description Specifies the domain part of the Salesforce user id. If Logon Domain is not an empty string, the driver first appends the @ character to the end of the User Name value and then appends the value of Logon Domain.

Valid Values *string*

where *string* is a valid user ID domain.

Default Empty string

GUI Tab [Security tab](#) on page 708

## Max Pool Size

Attribute	MaxPoolSize (MXPS)
Description	<p>The maximum number of connections allowed within a single connection pool. When the maximum number of connections is reached, no additional connections can be created in the connection pool.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 741</a> for details.</p>
Valid Values	<p>An integer from 1 to 65535</p> <p>For example, if set to 20, the maximum number of connections allowed in the pool is 20.</p>
Default	100
GUI Tab	<a href="#">Pooling tab</a> on page 710

## Min Pool Size

Attribute	MinPoolSize (MNPS)
Description	<p>The minimum number of connections that are opened and placed in a connection pool, in addition to the active connection, when the pool is created. The connection pool retains this number of connections, even when some connections exceed their Load Balance Timeout value.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 741</a> for details.</p>
Valid Values	<p>0   <i>x</i></p> <p>where <i>x</i> is an integer from 1 to 65535.</p> <p>For example, if set to 5, the start-up number of connections in the pool is 5 in addition to the current existing connection.</p> <p>If set to 0, no connections are opened in addition to the current existing connection.</p>
Default	0
GUI Tab	<a href="#">Pooling tab</a> on page 710

## Password

Attribute	Password (PWD)
Description	<p>Specifies the password to use to connect to your Salesforce instance. A password is required. Contact your system administrator to obtain your password.</p> <p>IMPORTANT: Setting the password using a data source is not recommended. The data source persists all options, including the Password option, in clear text.</p>
Valid Values	<p><i>password</i>   <i>password+securitytoken</i></p> <p>where:</p> <p><i>password</i> is a valid password. The password is case-sensitive.</p>

*password+securitytoken* is a valid password appended by the security token required to connect to the Salesforce instance, for example, `secretXaBARTsLZReM4Px47qPLOS`, where `secret` is the password and the remainder of the value is the security token. Both the password and security token are case-sensitive.

Optionally, you can specify the security token in the [Security Token](#) option. Do not specify the security token in both options.

- Default     None
- See Also    ["Security Token" on page 737](#)

**Proxy Host**

- Attribute    ProxyHost (PXHN)
- Description   Specifies the Hostname and possibly the Domain of the Proxy Server. The value specified can be a host name, a fully qualified domain name, or an IPv4 or IPv6 address.
- Valid Values   *server\_name* | *IP\_address*  
  
where:  
  
*server\_name* is the name of the server or a fully qualified domain name to which you want to connect.  
  
The IP address can be specified in either IPv4 or IPv6 format, or a combination of the two. See ["Using IP Addresses" on page 54](#) for details about these formats.
- Default     Empty string
- GUI Tab     [SQL Engine tab](#) on page 704

**Proxy Password**

- Attribute    ProxyPassword (PXPW)
- Description   Specifies the password needed to connect to the Proxy Server.
- Valid Values   String  
  
Specifies the password to use to connect to the Proxy Server. Contact your system administrator to obtain your password.
- Default     Empty string
- GUI Tab     [SQL Engine tab](#) on page 704

## Proxy Port

Attribute	ProxyPort (PXPT)
Description	Specifies the port number where the Proxy Server is listening for HTTP and/or HTTPS requests.
Valid Values	<i>port_name</i>  where the <i>port_name</i> is the port number of the server listener. Check with your system administrator for the correct number.
Default	0
GUI Tab	<a href="#">SQL Engine tab</a> on page 704

## Proxy User

Attribute	ProxyUser (PXUN)
Description	Specifies the user name needed to connect to the Proxy Server.
Valid Values	The default user ID that is used to connect to the Proxy Server.
Default	Empty string
GUI Tab	<a href="#">SQL Engine tab</a> on page 704

## Read Only

Attribute	ReadOnly (RO)
Description	Specifies whether the connection has read-only access to the data source.
Valid Values	0   1  If set to 1, the connection has read-only access. The following commands are the only commands that you can use when a connection is read-only: <ul style="list-style-type: none"> <li>■ Call (if the procedure does not update data)</li> <li>■ Explain Plan</li> <li>■ Select (except Select Into)</li> <li>■ Set Schema</li> </ul> <p>The driver returns an error if any other command is executed.</p> <p>If set to 0, the connection is opened for read/write access, and you can use all commands supported by the product.</p>
Default	0
GUI Tab	<a href="#">Advanced tab</a> on page 710

## Record Delimiter

Attribute	BulkLoadRecordDelimiter (BLRD)
Description	Specifies the character that the driver will use to delimit the record entries in a bulk load data file.

Valid Values *x*

where *x* is any printable character.

For simplicity, avoid using a value that can be in the data, including all alphanumeric characters, the dash(-), the colon(:), the period (.), the forward slash (/), the space character, the single quote (') and the double quote ("). You can use some of these characters as delimiters if all of the data in the file is contained within double quotes.

NOTE: The Bulk Load Record Delimiter character must be different from the Bulk Load Field Delimiter.

Default None

GUI Tab [Bulk tab](#) on page 711

## Refresh Dirty Cache

Attribute RefreshDirtyCache (RDC)

Description Specifies whether the driver refreshes a dirty cache on the next fetch operation from the cache. A cache is marked as dirty when a row is inserted into or deleted from a cached table or a row in the cached table is updated.

Valid Values 1 | 0

If set to 1 (Enabled), a dirty cache is refreshed when the cache is referenced in a fetch operation. The cache state is set to initialized if the refresh succeeds.

If set to 0 (Disabled), a dirty cache is not refreshed when the cache is referenced in a fetch operation.

Default 1 (Enabled)

See Also ["Refreshing Cache Data" on page 747](#)

GUI Tab [Advanced tab](#) on page 707

## Refresh Schema

Attribute RefreshSchema (RS)

Description Determines whether the driver automatically refreshes the information in a remote schema (rebuilds the database map for the schema) the first time a user connects to the specified embedded database. The database is opened when the user first makes a connection with the application. When all connections associated with that user are closed, then the driver closes the database. The database must be reopened before it can be used again.

NOTE: This connection option is functionally equivalent to executing the Refresh Schema statement (refer to ["Refresh Schema \(EXT\)"](#) in the *DataDirect Connect Series for ODBC Reference*). You can refresh a schema manually at any time by using the Refresh Schema statement.



Valid Values 1 | 0

If set to 1 (Enabled), the driver automatically refreshes the schema the first time a user connects to the specified database. Any schema objects that have changed since the last time the database map was rebuilt are reflected in the metadata. See ["Database Configuration File" on page 755](#) for information about embedded databases and map files.

If set to 0 (Disabled), the driver does not automatically refresh the schema the first time a user connects to the specified database.

Default 0 (Disabled)

GUI Tab [Advanced tab](#) on page 710

## Report Codepage Conversion Errors

Attribute ReportCodepageConversionErrors (RCCE)

Description Specifies how the driver handles code page conversion errors that occur when a character cannot be converted from one character set to another.

An error message or warning can occur if an ODBC call causes a conversion error, or if an error occurs during code page conversions to and from the database or to and from the application. The error or warning generated is `Code page conversion error encountered`. In the case of parameter data conversion errors, the driver adds the following sentence: `Error in parameter x`, where `x` is the parameter number. The standard rules for returning specific row and column errors for bulk operations apply.

Valid Values 0 | 1 | 2

If set to 0 (Ignore Errors), the driver substitutes 0x1A for each character that cannot be converted and does not return a warning or error.

If set to 1 (Return Error), the driver returns an error instead of substituting 0x1A for unconverted characters.

If set to 2 (Return Warning), the driver substitutes 0x1A for each character that cannot be converted and returns a warning.

Default 0 (Ignore Errors)

GUI Tab [Advanced tab](#) on page 707

## Security Token

Attribute SecurityToken (STK)

Description Specifies the security token required to make a connection to a Salesforce instance that is configured for a security token. If a security token is required and you do not supply one, the driver returns an error indicating that an invalid user or password was supplied. Contact your Salesforce administrator to find out if a security token is required.

NOTE: When setting the security token using a data source on Windows, the Security Token option is encrypted.

Valid Values	<i>string</i>  where <i>string</i> is the value of the security token assigned to the user.  Optionally, you can specify the security token in the <a href="#">Password</a> option by appending the security token to the password, for example, <code>secretXaBARTsLZReM4Px47qPL0S</code> , where <code>secret</code> is the password and the remainder of the value is the security token. Do not specify the security token in both options.
Default	Empty string
GUI Tab	<a href="#">Security tab</a> on page 708

**Server Port Number**

Attribute	ServerPortNumber (SPN)
Description	Specifies a valid port on which the SQL engine listens for requests from the driver.  NOTE: This option is ignored unless SQL Engine Mode is set to 1 (Server).
Valid Values	<i>port_name</i>  where the <i>port_name</i> is the port number of the server listener. Check with your system administrator for the correct number.
Default	19928
GUI Tab	<a href="#">SQL Engine tab</a> on page 704

**SQL Engine Mode**

Attribute	SQLEngineMode (SEM)
Description	Specifies whether the driver's SQL engine runs in the same 32-bit process as the driver (direct mode) or runs in a process that is separate from the driver (server mode). You must be an administrator to modify the server mode configuration values, and to start or stop the SQL engine service.
Valid Values	0   1  If set to 0 (Direct), the SQL engine runs in direct mode. The driver and its SQL engine run in a single process within the same JVM.  If set to 1 (Server), the SQL engine runs in server mode. The SQL engine operates in a separate process from the driver within its own JVM. You must start the SQL Engine service before using the driver (see <a href="#">"Starting the SQL Engine Server" on page 761</a> for more information). Multiple drivers on different clients can use the same service.  IMPORTANT: Changes you make to the server mode configuration affect all DSNs sharing the service.
Default	0 - Direct
GUI Tab	<a href="#">SQL Engine tab</a> on page 704

## Statement Call Limit

Attribute	StmtCallLimit (SCL)
Description	Specifies the maximum number of Web service calls the driver can make when executing any single SQL statement or metadata query.
Valid Values	0   $x$  where $x$ is a positive integer that defines the maximum number of Web service calls the driver can make when executing any single SQL statement or metadata query.  If set to 0, there is no limit.  If set to $x$ , the driver uses this value to set the maximum number of Web service calls on a single connection that can be made when executing a SQL statement. This limit can be overridden by changing the STMT_CALL_LIMIT session attribute using the ALTER SESSION statement. For example, the following statement sets the statement call limit to 10 Web service calls:  <pre>ALTER SESSION SET STMT_CALL_LIMIT=10</pre> If the Web service call limit is exceeded, the behavior of the driver depends on the value specified for the <a href="#">Stmt Call Limit Behavior</a> option.
Default	20
GUI Tab	<a href="#">Web Service tab</a> on page 709

## Stmt Call Limit Behavior

Attribute	StmtCallLimitBehavior (SCLB)
Description	Specifies the behavior of the driver when the maximum Web service call limit specified by the <a href="#">Statement Call Limit</a> option is exceeded.
Valid Values	1   2  If set to 1 (ErrorAlways), the driver returns an error if the maximum Web service call limit is exceed.  If set to 2 (ReturnResults), the driver returns any partial results it received prior to the call limit being exceeded. The driver generates a warning that not all of the results were fetched.
Default	1 (ErrorAlways)
GUI Tab	<a href="#">Web Service tab</a> on page 709

## Transaction Mode

Attribute	TransactionMode (TM)
Description	Specifies how the driver handles manual transactions.
Valid Values	0   1  If set to 1 - Ignore, the data source does not support transactions and the driver always operates in auto-commit mode. Calls to set the driver to manual commit mode and to commit transactions are ignored. Calls to rollback a transaction cause the driver to return an

error indicating that no transaction is started. Metadata indicates that the driver supports transactions and the ReadUncommitted transaction isolation level.

If set to 0 - No Transactions, the data source and the driver do not support transactions. Metadata indicates that the driver does not support transactions.

Default 0 (No Transactions)

GUI Tab [Advanced tab](#) on page 707

### User Name

Attribute LogonID (UID)

Description The default user ID that is used to connect to your database. Your ODBC application may override this value or you may override it in the logon dialog box or connection string.

Valid Values *userid*

where *userid* is a valid user ID with permissions to access the database.

Default None

GUI Tab [Security tab](#) on page 710

See Also ["Logon Domain" on page 732](#)

### WSFetch Size

Attribute WSFetchSize (WSFS)

Description Specifies the number of rows of data the driver attempts to fetch for each ODBC call.

Valid Values 0 | *x*

where *x* is a positive integer from 1 to 2000 that defines a number of rows.

If set to 0, the driver attempts to fetch up to a maximum of 2000 rows. This value typically provides the maximum throughput.

If set to *x*, the driver attempts to fetch up to a maximum of the specified number of rows. Setting the value lower than 2000 can reduce the response time for returning the initial data. Consider using a smaller WSFetch Size for interactive applications only.

Default 0 (up to a maximum of 2000 rows)

GUI Tab [Web Service tab](#) on page 709

See Also ["Fetch Size" on page 728](#)

["WSTimeout" on page 741](#)

### WSRetry Count

Attribute WSRetryCount (WSRC)

Description The number of times the driver retries a timed-out Select request. Insert, Update, and Delete requests are never retried. The timeout period is specified by the [WSTimeout](#) connection option.

Valid Values	0   $x$  where $x$ is a positive integer.  If set to 0, the driver does not retry timed-out requests after the initial unsuccessful attempt.  If set to $x$ , the driver retries the timed-out request the specified number of times.
Default	0
GUI Tab	<a href="#">Web Service tab</a> on page 709
See Also	<a href="#">"WSTimeout" on page 741</a>

## WSTimeout

Attribute	WSTimeout (WST)
Description	Specifies the time, in seconds, that the driver waits for a response to a Web service request.
Valid Values	0   $x$  where $x$ is a positive integer that defines the number of seconds the driver waits for a response to a Web service request.  If set to 0, the driver waits indefinitely for a response; there is no timeout.  If set to $x$ , the driver uses the value as the default timeout for any statement created by the connection.  If a Select request times out and <a href="#">WSRetry Count</a> is set to retry timed-out requests, the driver retries the request the specified number of times.
Default	120 (seconds)
GUI Tab	<a href="#">Web Service tab</a> on page 710
See Also	<a href="#">"WSRetry Count" on page 740</a>

---

## Performance Considerations

**Application Using Threads (ApplicationUsingThreads):** The driver coordinates concurrent database operations (operations from different threads) by acquiring locks. Although locking prevents errors in the driver, it also decreases performance. If your application does not make ODBC calls from different threads, the driver has no reason to coordinate operations. In this case, the ApplicationUsingThreads attribute should be disabled (set to 0).

NOTE: If you are using a multi-threaded application, you must enable the Application Using Threads option.

**Connection Pooling (ConnectionPooling):** If you enable the driver to use connection pooling, you can set additional options that affect performance:

- **Load Balance Timeout:** You can define how long to keep connections in the pool. The time that a connection was last used is compared to the current time and, if the

timespan exceeds the value of the Load Balance Timeout option, the connection is destroyed. The Min Pool Size option can cause some connections to ignore this value.

- **Connection Reset:** Resetting a re-used connection to the initial configuration settings impacts performance negatively because the connection must issue additional commands to the server.
- **Max Pool Size:** Setting the maximum number of connections that the pool can contain too low might cause delays while waiting for a connection to become available. Setting the number too high wastes resources.
- **Min Pool Size:** A connection pool is created when the first connection with a unique connection string connects to the database. The pool is populated with connections up to the minimum pool size, if one has been specified. The connection pool retains this number of connections, even when some connections exceed their Load Balance Timeout value.

**Enable Bulk Load:** For batch inserts and individual inserts, updates, and deletes, the driver can use the Salesforce Bulk API instead of the Web service API. Using the Bulk API significantly reduces the number of Web service calls the driver uses to transfer data to Salesforce and may improve performance.

**Fetch Size/WS Fetch Size:** The connection options Fetch Size and WSFetch Size can be used to adjust the trade-off between throughput and response time. In general, setting larger values for WSFetch Size and Fetch Size will improve throughput, but can reduce response time.

For example, if an application attempts to fetch 100,000 rows from the remote data source and WSFetch Size is set to 500, the driver must make 200 Web service calls to get the 100,000 rows. If, however, WSFetch Size is set to 2000 (the maximum), the driver only needs to make 50 Web service calls to retrieve 100,000 rows. Web service calls are expensive, so generally, minimizing Web service calls increases throughput. In addition, many Cloud data sources impose limits on the number of Web service calls that can be made in a given period of time. Minimizing the number of Web service calls used to fetch data also can help prevent exceeding the data source call limits.

For many applications, throughput is the primary performance measure, but for interactive applications, such as Web applications, response time (how fast the first set of data is returned) is more important than throughput. For example, suppose that you have a Web application that displays data 50 rows to a page and that, on average, you view three or four pages. Response time can be improved by setting Fetch Size to 50 (the number of rows displayed on a page) and WSFetch Size to 200. With these settings, the driver fetches all of the rows from the remote data source that you would typically view in a single Web service call and only processes the rows needed to display the first page.

---

## Data Types

The following table lists the data types supported by the Salesforce driver for local tables, how the Salesforce data types exposed by the Salesforce Web Service API map to them, and how the Salesforce Web Service API data types map to the ODBC data types.

**Table 21-2. Salesforce Data Types for Local Tables**

Salesforce Data Type	Web Service API Data Type	ODBC Data Type
ANYTYPE	anytype	SQL_WVARCHAR
AUTONUMBER	string	SQL_WVARCHAR
BINARY	binary	SQL_LONGVARBINARY
CHECKBOX	boolean	SQL_BIT
COMBOBOX	combobox	SQL_WVARCHAR
CURRENCY	currency	SQL_DOUBLE
DATE	date	SQL_TYPE_DATE
DATETIME	datetime	SQL_TYPE_TIMESTAMP
EMAIL	email	SQL_WVARCHAR
ENCRYPTEDTEXT	encryptedtext	SQL_WVARCHAR
HTML	html	SQL_WLONGVARCHAR
ID	id	SQL_WVARCHAR
INT	double	SQL_INTEGER
LONGTEXTAREA	longtextarea	SQL_WLONGVARCHAR
MULTISELECTPICKLIST	multipicklist	SQL_WVARCHAR
NUMBER	double	SQL_DOUBLE if scale does not = 0 or precision > 9 or the NumberFieldMapping key of the ConfigOptions connection option is set to 2.  SQL_INTEGER if scale = 0 and precision <= 9 and the NumberFieldMapping key of the ConfigOptions connection option is set to 1.
PERCENT	percent	SQL_DOUBLE
PHONE	phone	SQL_WVARCHAR
PICKLIST	picklist	SQL_WVARCHAR
REFERENCE	reference	SQL_WVARCHAR
TEXT	string	SQL_WVARCHAR
TEXTAREA	textarea	SQL_WVARCHAR
TIME	time	SQL_TYPE_TIME
URL	url	SQL_WVARCHAR

The following table lists the data types supported by the Salesforce driver for remote tables, how the Salesforce data types exposed by the Salesforce Web Service API map to them, and how the Salesforce Web Service API data types map to the ODBC data types.

**Table 21-3. Salesforce Data Types for Remote Tables**

Salesforce Data Type	Web Service API Data Type	ODBC Data Type
ANYTYPE	anytype	SQL_WVARCHAR
AUTONUMBER	string	SQL_WVARCHAR
BINARY	binary	SQL_LONGVARBINARY
CHECKBOX	boolean	SQL_BIT
COMBOBOX	combobox	SQL_WVARCHAR
CURRENCY	currency	SQL_DOUBLE
DATACATEGORYGROUPREFERENCE	DataCategoryGroupReference	SQL_WVARCHAR
DATE	date	SQL_TYPE_DATE
DATETIME	datetime	SQL_TYPE_TIMESTAMP
EMAIL	email	SQL_WVARCHAR
HTML	html	SQL_WLONGVARCHAR
ID	id	SQL_WVARCHAR
INT	double	SQL_INTEGER
LONGTEXTAREA	longtextarea	SQL_WLONGVARCHAR
MULTISELECTPICKLIST	multipicklist	SQL_WVARCHAR
NUMBER	double	SQL_DOUBLE if scale does not = 0 or precision > 9 or the NumberFieldMapping key of the ConfigOptions connection option is set to 2.  SQL_INTEGER if scale = 0 and precision <= 9 and the NumberFieldMapping key of the ConfigOptions connection option is set to 1.
PERCENT	percent	SQL_DOUBLE
PHONE	phone	SQL_WVARCHAR
PICKLIST	picklist	SQL_WVARCHAR
REFERENCE	reference	SQL_WVARCHAR
TEXT	string	SQL_WVARCHAR
TEXTAREA	textarea	SQL_WVARCHAR
TIME	time	SQL_TYPE_TIME
URL	url	SQL_WVARCHAR



---

## Mapping Objects to Tables

The driver automatically maps Salesforce objects and fields to tables and columns the first time it connects to a Salesforce instance. The driver maps both standard and custom objects and includes any relationships defined between objects.

The driver uses a local embedded database to instantiate the mapping of the remote data source objects to tables and the metadata associated with those tables. The driver creates a database per user. The embedded database is created in the directory from which the application is run and uses the user ID specified for the connection as the name of the database. If the user ID contains punctuation or other non-alphanumeric characters, the driver strips those characters from the user ID to form the name of the database. The driver provides connection options that you can use to override the default setting for the name and location of the database (Database).

---

## Client-Side Caches

The Salesforce driver can implement a client-side data cache for improved performance. Data is cached from the remote data source to the local machine on which the driver is located.

The driver caches data on a per-table basis, as opposed to caching the result of a particular query. Caching data on a table level allows the caches to be queried, filtered, and sorted in other queries. Once a cache is created, its use is transparent to the application. For example, if a cache is created on the Account table, then all subsequent queries that reference Account access the Account cache. Disabling or dropping the cache allows references to the Account table to access the remote data again. Because the use of the cache is transparent, no changes to the application are required to take advantage of the cache.

You must specifically create a cache before it can be populated; caches are not created automatically. After you have created a cache on a table, the cache will be populated as a result of the next operation on the table. For example, after creating a cache on Account, data is returned from the Salesforce data source and stored locally in the cache when you first execute the following statement:

```
SELECT ROWID, SYS_NAME FROM Account
```

Any subsequent queries against the Account table return data from the cache, which reduces response time. SQL queries can access both cached data and remote data (data stored in Salesforce that has not been assigned to a cache) in the same statement.

The caches maintained by the Salesforce driver are write-through caches. This means that, for any operation that modifies data in a table that is cached, the driver performs the operation on the remote data first and then updates the cache as much as possible.

To create, modify, refresh, or delete client-side data caches, use the following SQL statement extensions:

- Create Cache
- Alter Cache
- Refresh Cache
- Drop Cache

See the following sections for overviews of each extension. Refer to [Chapter 10 “SQL Statements and Extensions for the Salesforce Driver”](#) in the *DataDirect Connect Series for ODBC Reference* for descriptions of the syntax of these extensions.

## Creating a Cache

You create a cache using the Create Cache statement (refer to [“Create Cache \(EXT\)” on page 94](#) in the *DataDirect Connect Series for ODBC Reference*). A cache can be created on a single table or on a set of related tables. When creating a cache on a single table, you specify the name of the table to cache and can optionally specify a filter for the table. The filter determines whether the cache holds all of the data in the remote table or a subset of the data that matches the filter. You can also specify attributes for the Create Cache statement that determine:

- Whether the cache data is held on disk or in memory
- How often the cache data is refreshed
- Whether the cache is initially enabled
- Whether the driver checks to see if a refresh is needed at connect time

Creating a cache for a set of related tables is similar to creating a cache on a single table except that a primary table and one or more referencing tables are specified. This is useful if you want to cache a subset of data for a table and also cache data related to that subset of data. For example, you might have three tables, Account, Contact, and Opportunity, where both a contact and an opportunity belong to a particular account. Using a relational cache, you could specify that accounts that have had activity in the past year be cached, as well as caching the opportunities and contacts for only those cached accounts.

## Modifying a Cache Definition

Once a cache has been created, you can modify the definition of the cache or set of related caches with the Alter Cache statement (refer to [“Alter Cache \(EXT\)” on page 86](#) in the *DataDirect Connect Series for ODBC Reference*). Only the attributes of the cache can be modified through the Alter Cache statement; the table or related set of tables cannot be changed and a single table cache cannot be changed to a relational cache.

**WARNING:** Changing the attributes of a cache may cause the current data in the cache to be discarded and refetched from the remote data source.

## Disabling and Enabling a Cache

When a cache is defined on a table, all fetch operations performed on that table access the cache, essentially hiding the remote table from the application. At times, you may want an application to query the remote data instead of the cached data. For example, assume that

a cache was created on Account with a filter set to cache accounts that have had activity in the past year. You may want to run a query to get information about an account that has not been active for two years. One alternative would be to drop the Account cache, run the query, and then recreate the cache on Account, but this can be problematic. First, you must recreate the cache and make sure it had the same attributes as before. Second, the data in the cache is discarded and needs to be refetched when the cache is recreated. Depending on the amount of cached data, this could take a significant amount of time. To address this type of issue, the Salesforce driver can temporarily disable a cache. When a cache is disabled, its definition and data are maintained. Any queries that reference a table with a disabled cache access the remote table. When you want to access cached data again, the cache can be enabled.

## Refreshing Cache Data

To prevent the data in a cache from becoming out of date, the driver must periodically refresh the cache data with data from the remote data source. To minimize the amount of data that needs to be moved when a cache is refreshed, and therefore the time required to refresh it, the driver checks to see which records in the remote table have been added, modified, or deleted since the last time the cache was refreshed. The driver retrieves only data for added or modified records and removes only deleted records from the cache. You or the application can refresh the cache manually or the driver can refresh the cache automatically.

You can refresh a cache manually at any time by using the Refresh Cache statement (refer to ["Refresh Cache \(EXT\)" on page 115](#) in the *DataDirect Connect Series for ODBC Reference*). The Refresh Cache statement can also be used to perform a Clean (complete) refresh in addition to the standard optimized refresh. A Clean refresh discards all of the data from the cache and repopulates it with data from the remote data source.

The driver can refresh a cache automatically in one of two ways. When you create a cache, one of the attributes that you set is the refresh interval for the cache. During each cache query, the driver checks to see whether the time elapsed since the last refresh exceeds the refresh interval for the cache. If it has, the driver refreshes the cache before satisfying the query.

Update operations to a table that is cached can trigger the driver to refresh the cache automatically. The caches maintained by the Salesforce driver are write-through caches. For any operation that modifies data in a table that is cached, the driver performs the operation on the remote data first and then updates the cache as much as possible. The driver may not be able to update the cache with all of the modifications because some of the modified data may have been generated by the remote data source. For example, if a row is inserted but a value for all columns in the row is not required, any default values generated by the remote data source for columns not specified in the Insert statement would not be set in the cache. Because the driver cannot reflect all of the changes made when a cached table is modified, it sets the cache state to dirty. When a cache state is dirty, the next query that attempts to fetch data from that cache causes the driver to refresh the cache before the fetch operation is performed. This allows the fetch to see the values populated by the remote data source.

Automatically refreshing a dirty cache is not always desirable. For example, if an application alternates fetches and inserts on a table, and the insert does not depend on any remote data source generated values, then the refresh between fetches is unnecessary. The

[Refresh Dirty Cache](#) connection option (see ["Connection Option Descriptions" on page 718](#)) controls whether the driver automatically refreshes a cache with a dirty state. The state of a cache can be viewed by selecting the STATUS column of the SYSTEM\_CACHES catalog table. See ["SYSTEM\\_CACHES Catalog Table" on page 748](#) for more information.

## Dropping a Cache

You can drop an existing cache using the Drop Cache statement (refer to ["Drop Cache \(EXT\)" on page 110](#) in the *DataDirect Connect Series for ODBC Reference*). If a cache is a relational cache, the Drop Cache statement drops the cache for the primary table as well as the caches for the related tables.

NOTE: When a cache is dropped, all of the data in that cache is discarded.

## Cache MetaData

The Salesforce driver maintains information about the caches that have been created. The driver provides two system tables to expose the cache information, the SYSTEM\_CACHES table and the SYSTEM\_CACHE\_REFERENCES table.

The SYSTEM\_CACHES and SYSTEM\_CACHE\_REFERENCES system tables exist in the INFORMATION\_SCHEMA schema. See ["Catalog Tables" on page 748](#) for a complete description of the contents of these system tables.

---

## Catalog Tables

The Salesforce driver provides a standard set of catalog tables that maintain the information returned by various ODBC catalog functions such as SQLTables, SQLColumns, SQLDescribeParam and SQLDescribeCol. If possible use the ODBC catalog functions to obtain this information instead of querying the catalog tables directly.

The driver also provides additional catalog tables that maintain metadata specific to the Salesforce driver. This section defines the catalog tables that provide Salesforce driver-specific information. The catalog tables are defined in the INFORMATION\_SCHEMA schema.

### SYSTEM\_CACHES Catalog Table

The SYSTEM\_CACHES catalog table stores the definitions of the caches created on remote tables. The data in the SYSTEM\_CACHES table provides the name, type (single table or relational), status, and other information for each defined cache. The table name returned for a remote relational cache is the name of the primary table of the relational cache; however, its type is REMOTE RELATIONAL. You can query SYSTEM\_CACHES to determine the caches currently defined by the driver. The values in the SYSTEM\_CACHES table are read-only. The referenced tables of a relational cache can be determined by querying the SYSTEM\_CACHE\_REFERENCES catalog table (see ["SYSTEM\\_CACHE\\_REFERENCES Catalog Table" on page 750](#)).

The following table describes the columns of the SYSTEM\_CACHES table, which is sorted on the following columns: CACHE\_TYPE, TABLE\_SCHEMA, and TABLE\_NAME.

**Table 21-4. SYSTEM\_CACHES Catalog Table**

Column Name	Data Type	Description
TABLE_CAT	VARCHAR (128), NULLABLE	The catalog that contains the remote table on which the cache is defined. It is NULL for the Salesforce driver.
TABLE_SCHEM	VARCHAR (128), NULLABLE	The schema that contains the remote table on which the cache is defined.
TABLE_NAME	VARCHAR (128), NOT NULL	The name of the remote table on which the cache is defined.
CACHE_TYPE	VARCHAR (20), NOT NULL	The type cache, which can be either REMOTE TABLE or REMOTE RELATIONAL.
REFRESH_INTERVAL	INTEGER, NOT NULL	The refresh interval (in minutes).
INITIAL_CHECK	VARCHAR(20), NOT NULL	The value that defines when the initial refresh check is performed: ONFIRSTCONNECT or FIRSTUSE.
PERSIST	VARCHAR(20), NOT NULL	The value that defines whether the data in the cache is persisted past the lifetime of the connection: TEMPORARY, MEMORY, or DISK.
ENABLED	BOOLEAN, NOT NULL	The value that defines whether the cache is enabled for use with SQL statements: TRUE or FALSE.
CALL_LIMIT	INTEGER, NOT NULL	The maximum number of Web service calls that can be made when refreshing the cache. The value 0 indicates no call limit.
REFRESH_MODE	INTEGER, NOT NULL	For internal use only.
FILTER	VARCHAR(128), NULLABLE	The Where clause used to filter the rows that are cached.
LAST_REFRESH	DATETIME, NULLABLE	The time, in Coordinated Universal Time (UTC), the cache was last refreshed.
STATUS	VARCHAR(30)	<p>The Cache status. Valid values are:</p> <p>New: The cache has been created, but the data has not been populated.</p> <p>Initialized: The cache has been created and the data has been populated.</p> <p>Load aborted: The cache has been created, but the last attempt to populate the data failed. The cache is still valid. The next access attempts to populate the data again.</p> <p>Invalid: The cache is invalid. The second attempt to populate the data failed.</p> <p>Dirty: An insert or update operation has been performed on the cache and the cache has not been refreshed.</p>

## SYSTEM\_CACHE\_REFERENCES Catalog Table

The referenced tables in a relational cache can be determined by querying the SYSTEM\_CACHE\_REFERENCES system table. This table contains the names of the referenced tables as well as the name of the primary table with which they are associated.

The following table defines the columns of the SYSTEM\_CACHES table, which is sorted on the following columns: TABLE\_SCHEMA, TABLE\_NAME, and REF\_TABLE\_NAME.

**Table 21-5. SYSTEM\_CACHE\_REFERENCES**

Column	Data Type	Description
PRIMARY_TABLE_CAT	VARCHAR (128), NULLABLE	The catalog that contains the primary table of the relational cache. It is NULL for the Salesforce driver.
PRIMARY_TABLE_SCHEM	VARCHAR (128), NULLABLE	The schema that contains the primary table of the relational cache.
PRIMARY_TABLE_NAME	VARCHAR (128), NOT NULL	The primary table of the relational cache.
REF_TABLE_NAME	VARCHAR (128), NOT NULL	The name of the referenced table.
RELATIONSHIP_NAME	VARCHAR(128), NOT NULL	The name of the foreign key relationship used to relate this table to the primary table or one of the other tables in the relational cache.

## SYSTEM\_REMOTE\_SESSIONS Catalog Table

The system table named SYSTEM\_REMOTE\_SESSIONS stores information about the each of the remote sessions that are active for a given database. The values in the SYSTEM\_REMOTE\_SESSION table are read-only.

The following table defines the columns of the SYSTEM\_REMOTE\_SESSIONS table, which is sorted on the following columns: SESSION\_ID and SCHEMA.

**Table 21-6. SYSTEM\_REMOTE\_SESSIONS Catalog Table**

Column Name	Data Type	Description
SESSION_ID	INTEGER, NOT NULL	The connection (session) id with which the remote session is associated.
SCHEMA	VARCHAR(128), NOT NULL	The schema name that is mapped to the remote session.
TYPE	VARCHAR(30), NOT NULL	The remote session type. The current valid type is Salesforce.

**Table 21-6. SYSTEM\_REMOTE\_SESSIONS Catalog Table (cont.)**

Column Name	Data Type	Description
INSTANCE	VARCHAR(128)	<p>The remote session instance name or null if the remote data source does not have multiple instances.</p> <p>The Salesforce value for INSTANCE has the following form:</p> <p><i>Organization_Name</i> [Sandbox]</p> <p>where <i>Organization_Name</i> is the organization name of the Salesforce instance to which the connection is established. If the connection is established to a sandbox of the organization, then the word Sandbox is added to the end of the name.</p>
VERSION	VARCHAR(30), NOT NULL	<p>The version of the remote data source to which the session is connected.</p> <p>For Salesforce, this is the version of the Web Service API the driver is using to connect to Salesforce.</p>
CONFIG_OPTIONS	LONGVARCHAR, NOT NULL	The configuration options used to define the remote data model to relational data model mapping.
SESSION_OPTIONS	LONGVARCHAR, NOT NULL	The options used to establish the remote connection. This typically is information needed to log into the remote data source. The password value is not displayed.
WS_CALL_COUNT	INTEGER, NOT NULL	The number of Web service calls made through this remote session. The value of the WS_CALL_COUNT column can be reset using the ALTER SESSION statement.
WS_AGGREGATE_CALL_COUNT	INTEGER, NOT NULL	The total of all of the Web service calls made to the same remote data source by all active connections using the same server name and user ID.
REST_AGGREGATE_CALL_COUNT	INTEGER, NOT NULL	The number of REST calls made by this connection. REST calls are used for bulk operations, invoking reports, and describing report parameters.

## SYSTEM\_SESSIONINFO Catalog Table

The system table named SYSTEM\_SESSIONINFO describes details about your connection to Salesforce.

The following table defines the keys for the SYSTEM\_SESSIONINFO table. The values change based on your data source settings.

**Table 21-7. SYSTEM\_SESSIONINFO Catalog Table**

Key	Description
AUTOCOMMIT	Autocommit is always enabled.
DATABASE	The location and the filename prefix for the data mapping and configuration files.
DATABASE_READONLY	Indicates whether the database the session is connected to is read only
DB_FILE_LOCATION	The fully qualified path to the directory or folder that contains the database and mapping files.
DB_FILE_PREFIX	The filename prefix of the database and mapping files the driver is using.
IDENTITY	Currently always zero
MAXROWS	Currently always zero
LOG_CONFIG_FILE	The fully qualified path to the directory or folder that contains the logging configuration file.
SCHEMA	The name of the remote Salesforce schema.
SESSION_ID	The ID for this session
SESSION_READONLY	Indicates whether the session is read only
USER	The user that is associated with this session.

## SYSTEM\_SESSIONS Catalog Table

The system table named SYSTEM\_SESSIONS stores information about current system sessions. The values in the SYSTEM\_SESSIONS table are read-only.

The following table defines the columns of the SYSTEM\_SESSIONS table.

**Table 21-8. SYSTEM\_SESSIONS**

Column	Data Type	Description
SESSION_ID	INTEGER, NOT NULL	A unique ID that identifies this session. The system function CURSESSIONID( ) returns the session ID associated with the connection. Refer to <a href="#">Chapter 10 “SQL Statements and Extensions for the Salesforce Driver”</a> in the <i>DataDirect Connect Series for ODBC Reference</i> for details on CURSESSIONID().
CONNECTED	DATETIME, NOT NULL	The date and time the session was established.



**Table 21-8. SYSTEM\_SESSIONS** (cont.)

USER_NAME	VARCHAR (128), NOT NULL	The name of the embedded database that the session is using.
IS_ADMIN	BOOLEAN	For internal use only.
AUTOCOMMIT	BOOLEAN, NOT NULL	For future use.
READONLY	BOOLEAN, NOT NULL	True if the connection is in read-only mode. The READONLY status is based on whether the connection has been explicitly set to read-only mode by the Read Only connection option.
MAXROWS	INTEGER, NOT NULL	For future use.
LAST_IDENTITY	BIGINT, NULLABLE	For future use.
TRANSACTION_SIZE	INTEGER, NOT NULL	For future use.
CURRENT_SCHEMA	VARCHAR (128), NOT NULL	The current schema for the session. The current schema may be changed using the ALTER SESSION SET CURRENT_SCHEMA statement.
STMT_CALL_LIMIT	INTEGER, NOT NULL	The maximum number of Web service calls that the driver uses in attempting to execute a query to a remote data source. The statement call limit for the session may be changed via the ALTER SESSION SET STMT_CALL_LIMIT statement.

## Timeouts

The following types of timeout situations can occur when connecting to Salesforce:

- **Session timeouts.** Most remote data sources impose a limit on the duration of active sessions, meaning a session can fail with a session timeout error if the session extends past the limit. This is particularly true when connection pooling is used. The driver automatically attempts to re-establish a new session if the driver receives a session timeout error from a data source. The driver uses the initial servername, port (if appropriate), remote user ID, and remote password (encrypted) to re-establish the session. If the attempt fails, the driver returns an error indicating that the session timed out and the attempt to re-establish the session failed.
- **Web service request timeouts.** You can configure the driver to never time out while waiting for a response to a Web service request or to wait for a specified interval before timing out by setting the connection option [WSTimeout](#). For fetch requests only, if the request times out, you can configure driver to retry the request a specified number of times by setting the [WSRetry Count](#) connection option. If all subsequent attempts to retry a request fails, the driver returns an error indicating that the service request timed out and the subsequent requests failed. See "[Connection Option Descriptions](#)" on [page 718](#) for details on the WS Timeout and WS Retry Count connection options.

---

## Views and Remote/Local Tables

You can create views with the Create View statement. A view is like a named query. The view can refer to any combination of remote and local tables as well as other views.

You can create a remote or local table using the Create Table statement. A remote table is a Salesforce object and is exposed in the SFORCE schema. A local table is maintained by the driver and is local to the machine on which the driver is running. A local table is exposed in the PUBLIC schema.

Refer to [Chapter 10 “SQL Statements and Extensions for the Salesforce Driver”](#) in the *DataDirect Connect Series for ODBC Reference* for details on the Create View and Create Table statements and other SQL statements supported by the driver.

---

## Using Identifiers

Identifiers are used to refer to objects exposed by the driver, such as tables, columns, or caches. The driver supports both unquoted and quoted identifiers for naming objects. An unquoted identifier must start with an ASCII alpha character and can be followed by zero or more ASCII alpha or numeric characters. Unquoted identifiers are converted to uppercase before being used.

Quoted identifiers must be enclosed in double quotation marks ("" ). A quoted identifier can contain any Unicode character, including the space character, and is case-sensitive. The Salesforce driver recognizes the Unicode escape sequence \uxxxx as a Unicode character. You can specify a double quotation mark in a quoted identifier by escaping it with a double quotation mark.

The maximum length of both quoted and unquoted identifiers is 128 characters.

NOTE: When object names are passed as arguments to catalog functions, the case of the value must match the case of the name in the database. If an unquoted identifier name was used when the object was created, the value passed to the catalog function must be uppercase because unquoted identifiers are converted to uppercase before being used. If a quoted identifier name was used when the object was created, the value passed to the catalog function must match the case of the name as it was defined. Object names in results returned from catalog functions are returned in the case that they are stored in the database.

# Database Configuration File

You can configure an embedded database and data mapping using a database configuration file in XML format. Some of these values you can set in the file are the same as those you can set using the [Config Options](#) connection option (see ["Connection Option Descriptions" on page 718](#)). Some database configuration values can be set only using a configuration file.

The name of the database configuration file has the format:

*databaseName.config*

where *databaseName* is the name of the database to be configured. For example, if your environment has a database named mydb or a database configuration file named mydb.config, when the driver establishes a connection, it performs the following tasks:

- Checks to see if an embedded database named mydb exists (or a database using the default *databaseName* if one is not specified). If mydb exists, the driver connects to the remote data source using the mydb database.
- If mydb does not exist and the driver is configured to create a database, the driver looks for a database configuration file named mydb.config. If the database configuration file exists, the driver creates the database and mapping using the properties specified in the database configuration file.
- If mydb.config does not exist, the driver generates a database configuration file with default settings and uses those settings to create the database and its mapping.

The following is an example database configuration file:

```
<?xml version='1.0' encoding='UTF-8'?>
<Database xmlns="http://datadirect.com/cloud/config/1.0">
  <User name="CONNECT2" defaultSchema="SFORCE">
    <UseSchema name="SFORCE"/>
    <UseSchema name="PUBLIC"/>
  </User>
  <Schema name="SFORCE" type="Salesforce">
    <ConfigOptions>uppercaseidentifiers=true;localtables=0;auditcolumns=none;
      customsuffix=strip;KeywordConflictSuffix=;</ConfigOptions>
    <SessionOptions>loginhost=test.salesforce.com;userid=
      connect2@progress.com</SessionOptions>
  </Schema>
  <Schema name="PUBLIC" type="local">
  </Schema>
</Database>
```

The following are descriptions of the elements of the database configuration file.

## Database

Child Elements User, Schema

Description The Database element is the root element of the database configuration file. It does not define any configuration; it contains all of the elements that do define the database configuration. One and only one Database element must exist.

**User**

Parent Element	Database
Child Element	UseSchema
Description	Specifies the User ID used by the driver. At least one User element must exist.
Attributes	<p>name [required]: The user name is a string with a maximum length of 128 characters. The default is name=<i>userid</i>, where <i>userid</i> is the User ID used by the driver.</p> <p>defaultSchema: The name of the schema to use for unqualified table and column identifiers. If defaultSchema is not specified, the schema specified in the first useSchema child element is used as the default schema. The default is defaultSchema=SFORCE.</p>

**UseSchema**

Parent Element	User
Child Element	None
Description	<p>The UseSchema element specifies a schema that is visible to the user of this element. A schema contains the mapping between the remote data model and the relational tables the driver exposes. Multiple schemas can be associated with a user. At least one UseSchema element must exist.</p> <p>A basic User definition typically has two UseSchema elements: one that specifies the mapping to the remote data source and one for the local schema.</p>
Attributes	name [required]: The name of the schema to associate with the user. The schema name is a string with a maximum length of 128 characters. The defaults are name=SFORCE [remote] and name=PUBLIC [local].

**Schema**

Parent Element	Database
Child Elements	ConfigOptions, SessionOptions
Description	<p>The Schema element defines the schema that contains the mapping for a remote data source. The database configuration file must contain at least one schema definition and may contain multiple schema definitions. Each schema definition defines the type of the data source to which the schema maps, the information to connect to the remote database (except password), and the information needed to configure the remote data model to relational table mapping. At least one Schema element must exist.</p>
Attributes	<p>name [required]: The name of the schema that defines the data model to relational mapping. This attribute can be any valid identifier name. The defaults are name=SFORCE for the remote data source and name=PUBLIC for the local database.</p> <p>type [required]: The type of remote data source for which the schema defines mapping. This attribute must be type=Salesforce for the remote data source and type=local for the local database.</p>

## ConfigOptions

Parent Element	Schema
Child Element	None
Description	The ConfigOptions element is a string that specifies the configuration options used to define how the remote data source data model is mapped to relational tables. The ConfigOption string has the same keys, values, and syntax as the Config Options connection option (see <a href="#">"Connection Option Descriptions" on page 718</a> ) except that the enclosing parentheses are not required (see example configuration file). The default is an empty string.
Attributes	None

## SessionOptions

Parent Element	Schema
Child Element	None
Description	The SessionOptions element is a string of key value pairs that specifies the information needed to connect to the remote data source. SessionOptions includes the server name and remote user id, for example:  loginhost=login.salesforce.com;userid=john.public@abccorp.com
Attributes	None

---

## Mapping Objects to Tables

The Salesforce driver automatically maps Salesforce data source objects and fields to tables and columns the first time it connects to a data source instance. The driver maps both standard and custom objects and includes any relationships defined between objects. You can use SQLPrimaryKeys and SQLForeignKeys to report relationships among objects.

By default, the Salesforce driver does not include audit columns in table definitions when mapping Salesforce objects to tables. The [Config Options](#) connection option can be used to include audit columns. The following columns can be included or excluded:

- CreatedById
- CreatedByDate
- LastModifiedId
- LastModifiedDate
- SystemModestamp
- MasterRecordId

When mapping custom objects and fields, the Salesforce driver strips the standard "\_\_c" suffix from the names of the custom objects and fields by default. You can set the CustomSuffix key of the [Config Options](#) connection option to prevent the driver from stripping the "\_\_c" suffix. When mapping Salesforce system fields to columns in a table, the driver changes the column name to make it evident that the column is a system column. If you do not want the driver to change the names of system columns, set the MapSystemColumnNames key of the Config Options connection option to 0.

The [Create Database](#) connection option allows you to update or re-create the embedded database that defines and handles the object-to-table mapping.

---

## Reports

The Salesforce driver exposes reports defined on a Salesforce instance as stored procedures. An application can obtain a list of the reports defined on a Salesforce instance by calling the `SQLProcedures` catalog function. The names of the reports that can be invoked through the driver are listed in the `PROCEDURE_NAME` name column of the `SQLProcedures` results.

Salesforce organizes reports into folders. The Salesforce driver incorporates the folder name and report name into the procedure name reported by `SQLProcedures`. The driver creates the reported procedure name by prepending the folder name to the report name using an underscore to join them. Additionally, any spaces in the report or folder names are replaced with an underscore character. Like all identifier name metadata returned by the driver, the procedure name is uppercase. For example, if a report named Opportunity Pipeline is in the folder Opportunity Reports, it would be rendered as:

`OPPORTUNITY_REPORTS_OPPORTUNITY_PIPELINE`

An application invokes a report using the standard Call escape syntax, `{call report name}`, and ODBC mechanisms for calling a stored procedure that returns a resultset. The following example shows one way to invoke the Opportunity Pipeline report:

```
SQLRETURN      retVal;
HSTMT          hStmt = NULL;
SQLWCHAR*      sql;

sql = L"{call OPPORTUNITY_REPORTS_OPPORTUNITY_PIPELINE}";
retVal = SQLExecDirect(hStmt, sql, SQL_NTS);
if (SQL_SUCCESS == retVal) {
    // process results
}
```

### NOTES:

- The API used by the driver to obtain the list of reports and execute the reports is not an API that is documented by Salesforce. This API may change or may not be supported in the future.
- When passing parameters to stored procedures, reports are not supported.

---

## Connecting Through a Proxy Server

In some environments, your application may need to connect through a proxy server, for example, if your application accesses an external resource such as a Web service. At a minimum, your application needs to provide the following connection information when you invoke the JVM if the application connects through a proxy server:

- Server name or IP address of the proxy server
- Port number on which the proxy server is listening for HTTP/HTTPS requests

In addition, if authentication is required, your application may need to provide a valid user ID and password for the proxy server. Consult with your system administrator for the required information.

For example, the following command invokes the JVM while specifying a proxy server named `pserver`, a port of 808, and provides a user ID and password for authentication:

```
java -Dhttp.proxyHost=pserver -Dhttp.proxyPort=808 -Dhttp.proxyUser=smith
-Dhttp.proxyPassword=secret -cp sfce.jar com.acme.myapp.Main
```

Alternatively, you can use the `ProxyHost`, `ProxyPort`, `ProxyUser`, and `ProxyPassword` connection properties, but these options are applied only for the first connection. See ["Connection Option Descriptions" on page 718](#) for details about these properties.

---

## Configuring the SQL Engine Server

Some applications may experience problems loading the JVM required for the SQL engine because the process exceeds the available heap space. If your application experiences problems loading the JVM, you can configure the Salesforce driver to operate in server mode.

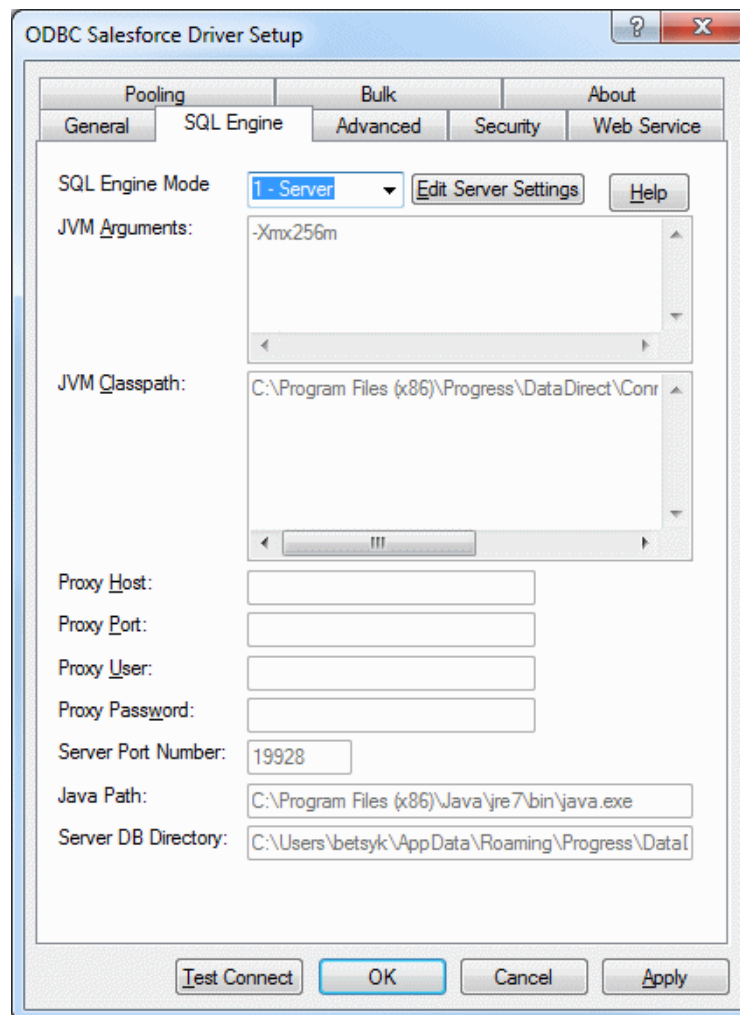
By default, the Salesforce driver operates in direct mode, with the SQL engine and JVM running in a single 32-bit process within the same JVM. In server mode, the driver's SQL engine runs in a separate 32-bit process with its own JVM instead of trying to load the SQL engine and JVM in the same process used by the driver.

NOTE: You must be an administrator to start or stop the service, or to configure any settings for the service.

### Configuring Server Mode

- 1 Set the SQL Engine Mode connection property to a value of `1 - Server`. All fields on the SQL Engine tab become readonly, and the **Edit Server Settings** button appears.
- 2 Click **Edit Server Setting** to display the ODBC Salesforce SQL Engine Service Setup dialog box. Use this dialog box to define settings for Server Mode and to start and stop the Progress DataDirect Salesforce SQL Engine service.

The SQL Engine Service Setup dialog box appears.



**JVM Arguments:** A string that contains the arguments that are passed to the JVM that the driver is starting. The location of the JVM must be specified on your PATH. See [JVM Arguments](#).

**JVM Class Path:** Specifies the CLASSPATH for the JVM used by the driver. See [JVM Classpath](#).

**Proxy Host:** Specifies the Hostname and possibly the Domain of the Proxy Server. See [Proxy Host](#).

**Proxy Port:** Specifies the port needed to connect to the Proxy Server. See [Proxy Port](#).

**Proxy User:** Specifies the user name needed to connect to the Proxy Server. See [Proxy User](#).

**Proxy Password:** Specifies the password needed to connect to the Proxy Server. See [Proxy Password](#).

**Server Port Number:** Specifies a valid port on which the SQL engine listens for requests from the driver. By default, the server listens on port 19928. See [Server Port Number](#) for more information.



**Java Path:** Specifies fully qualified path to the J2SE 5 or higher JVM executable that you want to use to run the SQL Engine Server. The path must not contain double quotation marks.

**Server DB Directory:** Specifies the path of the working directory for the SQL engine service to use to store the newly created database files or locate the existing database files. If the [Database](#) connection option contains a file name prefix, the user's local database is created at the path specified by Server DB Directory. However, if the Database connection option contains a fully qualified path, the user's local database is created using that path; the path specified by Server DB Directory is ignored.

**Services:** Shows the Salesforce ODBC SQL engine service that runs as a separate process instead of being loaded within the process of an ODBC application.

**Start (Stop):** Starts or stops the Salesforce service. A message window is displayed, confirming that the Salesforce service was started or stopped.

**Apply:** Applies the changes.

- 3 When you complete your changes, click **Apply**.
- 4 Click **OK** to save the changes and return to the SQL Engine tab or click **Cancel**.

## Starting the SQL Engine Server

In server mode, you must start the SQL engine server before using the driver. Before starting the SQL engine server, choose a directory to store the local database files. Make sure that you have the correct permissions to write to this directory.

By default, the JVM Classpath is set to the sfce.jar file in the installation directory.

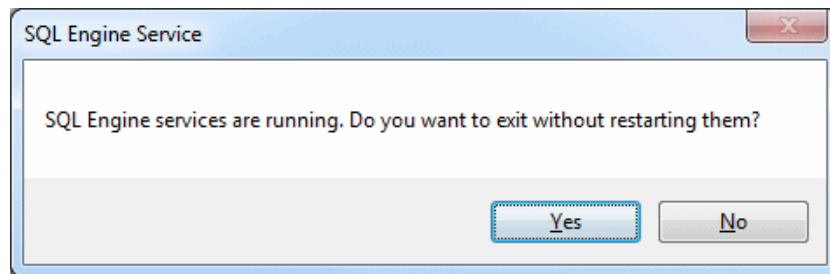
### To start the SQL engine server:

- 1 Start the ODBC Administrator by selecting its icon from the DataDirect Connect program group.
- 2 Select a tab:
  - **User DSN:** If you are configuring an existing user data source, select the data source name and click **Configure** to display the driver Setup dialog box.  
 If you are configuring a new user data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.
  - **System DSN:** If you are configuring an existing system data source, select the data source name and click **Configure** to display the driver Setup dialog box.  
 If you are configuring a new system data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.
  - **File DSN:** If you are configuring an existing file data source, select the data source file and click **Configure** to display the driver Setup dialog box.

If you are configuring a new file data source, click **Add** to display a list of installed drivers; then, select a driver. Click **Advanced** if you want to specify attributes; otherwise, click **Next** to proceed. Specify a name for the data source and click **Next**. Verify the data source information; then, click **Finish** to display the driver Setup dialog box.

- 3 On the ODBC Salesforce Driver Setup dialog box, select the **SQL Engine** tab; then, select **1- Server** from the SQL Engine Mode drop-down list.
- 4 Click **Edit Server Settings**.
- 5 When you complete your changes, click **Apply**.
- 6 Verify that Progress DataDirect Salesforce SQL Engine is selected in the Services drop-down list, and then, click **Start** to start the service. A message window appears to confirm that the service is running. Click **OK**.
- 7 Click **OK** to close the ODBC Salesforce SQL Engine Service Setup dialog box.

NOTE: If you made changes after starting the service, a message window is displayed:



If you want the service to run with the new settings, click **No**. Then, click **Stop** to stop the service, and then click **Start** to restart the service. Then, click **OK** to close the ODBC Salesforce SQL Engine Service Setup dialog box.

## Stopping the SQL Engine Server

- 1 To stop the SQL engine server, open the ODBC Salesforce Driver Setup dialog box and select the SQL Engine tab.
- 2 Select **1- Server** from the SQL Engine Mode drop-down list. Then, click **Edit Server Settings**.
- 3 Click **Stop** to stop the service. A message window appears to confirm that the service is stopped. Click **OK**.
- 4 Click **OK** to close the ODBC Salesforce SQL Engine Service Setup dialog box.

## Configuring Java Logging for the SQL Engine Server

Java logging can be configured by placing a logging configuration file named `ddlog.properties` in the Server DB directory (see ["Configuring Server Mode" on page 759](#) for information on configuring Server DB Directory). The simple way to create one of these is to make a copy of the `ddlog.properties` file, which is located in your driver installation directory, in the `install_dir/Sample/Example` subdirectory. For more information on logging in

Salesforce, refer to ["Logging \(Salesforce Driver\)"](#) in [Chapter 1](#) of the *DataDirect Connect Series for ODBC Troubleshooting Guide*.

---

## Unicode Support

The Salesforce driver is fully Unicode enabled. On UNIX and Linux platforms, the driver supports both UTF-8 and UTF-16. On Windows platforms, the Salesforce driver supports UCS-2/UTF-16 only.

The driver supports the Unicode ODBC W (Wide) function calls, such as SQLConnectW. This allows the Driver Manager to transmit these calls directly to the driver. Otherwise, the Driver Manager would incur the additional overhead of converting the W calls to ANSI function calls, and vice versa.

See ["UTF-16 Applications on UNIX and Linux" on page 125](#) for related details. Also, refer to [Chapter 4 "Internationalization, Localization, and Unicode"](#) in the *DataDirect Connect Series for ODBC Reference* for a more detailed explanation of Unicode.

---

## Advanced Features

The driver supports the following advanced features:

- Failover
- Security
- Connection Pooling

### Failover

The Salesforce driver supports reconnection after a session timeout. Salesforce, like most web-based connections has a session timeout associated with it. The Salesforce driver will reconnect to Salesforce if it receives an error from Salesforce indicating the session has timed out. No configuration is needed.

You can configure the Salesforce driver to retry web service fetch operations if the web service operation timed out; Insert, Update and Delete operations are not retried. the [WSRetry Count](#) connection option specifies whether the Salesforce driver retries fetch operations and the number of times it retries. The [WSTimeout](#) connection option specifies Web Service timeout period.

### Security

No configuration is required to use SSL. By default, all communication using the driver is SSL-encrypted. SSL secures the integrity of your data by encrypting information and providing authentication. See ["Data Encryption Across the Network" on page 78](#) for an overview.

Depending on how the Salesforce instance is configured, a security token may need to be included with the user id and password. The [Security Token](#) connection option specifies the token.

## Connection Pooling

The driver supports connection pooling and its related connection options. Connection pooling connection options are located on the [Pooling tab](#) of the driver Setup dialog box. See ["Using DataDirect Connection Pooling" on page 82](#) for a general description of connection pooling and its implementation.

---

## Parameter Metadata Support

The Salesforce driver supports returning parameter metadata as described in this section.

### Insert and Update Statements

The Salesforce driver supports returning parameter metadata for the following forms of Insert and Update statements:

- `INSERT INTO foo VALUES(?, ?, ?)`
- `INSERT INTO foo (col1, col2, col3) VALUES(?, ?, ?)`
- `UPDATE foo SET col1=?, col2=?, col3=? WHERE col1 operator ? [{AND | OR} col2 operator ?]`

where *operator* is any of the following SQL operators: =, <, >, <=, >=, and <>.

### Select Statements

The Salesforce driver supports returning parameter metadata for Select statements that contain parameters in ANSI SQL 92 entry-level predicates, for example, such as COMPARISON, BETWEEN, IN, LIKE, and EXISTS predicate constructs. Refer to the ANSI SQL reference for detailed syntax.

Parameter metadata can be returned for a Select statement if one of the following conditions is true:

- The statement contains a predicate value expression that can be targeted against the source tables in the associated FROM clause. For example:

```
SELECT * FROM foo WHERE bar > ?
```

In this case, the value expression "bar" can be targeted against the table "foo" to determine the appropriate metadata for the parameter.

- The statement contains a predicate value expression part that is a nested query. The nested query's metadata must describe a single column. For example:

```
SELECT * FROM foo WHERE (SELECT x FROM y WHERE z = 1) < ?
```

The following Select statements show further examples for which parameter metadata can be returned:

```
SELECT col1, col2 FROM foo WHERE col1 = ? and col2 > ?
SELECT ... WHERE colname = (SELECT col2 FROM t2 WHERE col3 = ?)
SELECT ... WHERE colname LIKE ?
SELECT ... WHERE colname BETWEEN ? and ?
SELECT ... WHERE colname IN (?, ?, ?)
SELECT ... WHERE EXISTS(SELECT ... FROM T2 WHERE col1 < ?)
```

ANSI SQL 92 entry-level predicates in a WHERE clause containing GROUP BY, HAVING, or ORDER BY statements are supported. For example:

```
SELECT * FROM t1 WHERE col = ? ORDER BY 1
```

Joins are supported. For example:

```
SELECT * FROM t1,t2 WHERE t1.col1 = ?
```

Fully qualified names and aliases are supported. For example:

```
SELECT a, b, c, d FROM T1 AS A, T2 AS B WHERE A.a = ? and B.b = ?"
```

---

## Using DataDirect Bulk Load With the Salesforce Driver

The driver supports DataDirect bulk load. Bulk load connection options are located on the [Bulk tab](#) of the driver Setup dialog box. The driver sends data to a Salesforce instance using the Salesforce Bulk API instead of the Web Service API. Using the Bulk API significantly reduces the number of Web service calls the driver uses to transfer data and may improve performance.

See ["Using DataDirect Bulk Load" on page 85](#) for a general description of DataDirect bulk load and its implementation.

### Bulk Operation Support

The Salesforce driver supports additional bulk operations besides bulk Insert when loading a data from an input file.

Features include:

- Bulk Delete - The input file contains only a list of values for a Salesforce primary key column. The driver uses the bulk protocol to delete the rows corresponding to the key values.
- Bulk Update - The input file contains a list of values for a Salesforce primary key column (required) and columns for which the data should be updated. The driver uses the bulk protocol to update the columns based on the rows that match the primary key values.

- Bulk Upsert - The input file contains a list of values for a Salesforce External Id column (required) and column values for which the data should be updated if the external id value already exists or inserted to the table if the external id value does not exist.

Refer to [Chapter 9 “DataDirect Bulk Load”](#) in the *DataDirect Connect Series for ODBC Reference* for details.

## Using Column Mapping

The Salesforce driver can use the DataDirect functions LoadTableFromFile (for an ANSI application) and LoadTableFromFileW (for a Unicode application) to support mapping of the column names defined in the input file to columns defined in the Salesforce table.

To take advantage of this feature, create a list of columns that are *not* auto-generated by Salesforce and specify a string argument of the format:

```
TableName(ColumnName[,ColumnName[,ColumnName]...])
```

for the TableName parameter of the LoadTableFromFile(W) function.

By default, the driver requires a one-to-one mapping of the columns defined in the input file to the columns defined in the Salesforce table. Because Salesforce tables contain several auto-generated key columns, it is not possible to achieve the required one-to-one mapping. Use only columns that were *not* auto-generated in the column list for which the data file defines the data that you want to load.

Refer to [Chapter 9 “DataDirect Bulk Load”](#) in the *DataDirect Connect Series for ODBC Reference* for details.

## Using Bulk Load for Single Inserts/Updates/Deletes

You can configure the driver to use the bulk load protocol when it encounters a request to execute a single Insert, Update, or Delete statement that affects many rows. When the EnableBulkLoad connection option is set to true, the Salesforce driver uses the Salesforce Bulk API for single Insert, Update, and Delete statements if the number of rows affected by the operation exceeds the threshold set by the BulkLoadThreshold connection option.

For example, if you set the EnableBulkLoad connection option to true and the BulkLoadThreshold connection option to 2000, executing the following statement would use the Bulk API if the number of rows returned by `SELECT rowid, sys_name FROM account` is more than 2000 rows.

```
INSERT INTO tmpAccounts(accountId, accountName)
SELECT rowid, sys_name FROM account
```

## Error Handling

The Salesforce driver reports errors to the application by returning SQL\_ERROR to the failing ODBC API call. The application can then call SQLGetDiagRec to obtain the error details which consist of the following information:

- Description of the probable cause of the error, prefixed by the component that generated the error
- Vendor error code (if applicable)
- String containing the XOPEN SQLState

### Driver Errors

An error generated by the driver has the following format:

```
[DataDirect][ODBC Salesforce Driver]message
```

For example:

```
[DataDirect][ODBC Salesforce Driver]Timeout expired.
```

You may need to check the last ODBC call your application made and refer to the ODBC specification for the recommended action.

**NOTE TO DATABASE.COM USERS:** The error text reads [ODBC Salesforce Driver] even if the driver is connected to Database.com.

### Data Source Errors

An error generated by the remote or local data source has the following format:

```
[DataDirect][ODBC Salesforce Driver][Salesforce] message
```

For example:

```
[DataDirect][ODBC Salesforce Driver][Salesforce]
Invalid Object Name.
```

Refer to your Salesforce documentation for details on the returned message.

## Isolation and Lock Levels Supported

Salesforce supports isolation level 0 (read uncommitted).

Refer to [Chapter 7 “Locking and Isolation Levels”](#) in the *DataDirect Connect Series for ODBC Reference* for details.

## SQL Support

Refer to [Chapter 10 “SQL Statements and Extensions for the Salesforce Driver”](#) in the *DataDirect Connect Series for ODBC Reference* for information about the SQL statements and extensions supported by the Salesforce driver.

## ODBC Conformance Level

Refer to [Chapter 2 “ODBC API and Scalar Functions”](#) in the *DataDirect Connect Series for ODBC Reference* for a list of the Core and Level 1 functions supported by the driver for Force.com Web Services API.

The Salesforce driver extends the standard results returned by the `SQLColumns` ODBC function to include the `IS_EXTERNAL_ID` column, as shown in the following table.

**Table 21-9. Extended Functionality for the `SQLColumns` Function**

Column	Data Type	Description
<code>IS_EXTERNAL_ID</code>	<code>VARCHAR (3), NOT NULL</code>	<p>Provides an indication of whether the column can be used as an External ID. External ID columns can be used as the lookup column for insert and upsert operations and foreign-key relationship values.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> <li>■ <b>YES:</b> The column can be used as an external ID.</li> <li>■ <b>NO:</b> The column cannot be used as an external ID.</li> </ul> <p>The standard catalog table <code>SYSTEM_COLUMNS</code> is also extended to include the <code>IS_EXTERNAL_ID</code> column.</p>

The Salesforce driver supports only the following Level 2 functions:

- `SQLColumnPrivileges`
- `SQLDescribeParam`
- `SQLForeignKeys`
- `SQLPrimaryKeys`
- `SQLProcedures`
- `SQLTablePrivileges`

## Number of Connections and Statements Supported

The driver supports multiple connections and multiple statements per connection to the Force.com Web Services API.



## 22 The Sybase IQ Wire Protocol Driver

The DataDirect Connect XE *for* ODBC and DataDirect Connect64 XE *for* ODBC Sybase IQ Wire Protocol driver (the Sybase IQ Wire Protocol driver) support the following Sybase IQ database servers:

- Sybase IQ Server 15.0, 15.1, 15.2, and 15.3

The Sybase IQ Wire Protocol driver is supported in the Windows, UNIX, and Linux environments. See ["Environment-Specific Information" on page 45](#) for detailed information about the environments supported by this driver.

Refer to the readme file shipped with your DataDirect product for the file name of the Sybase IQ Wire Protocol driver.

---

### Driver Requirements

The driver has no client requirements.

---

### Configuring and Connecting to Data Sources

After you install the driver, you configure data sources to connect to the database. See [Chapter 1 "Quick Start Connect" on page 35](#) for an explanation of different types of data sources. The data source contains connection options that allow you to tune the driver for specific performance. If you want to use a data source but need to change some of its values, you can either modify the data source or override its values at connection time through a connection string.

If you choose to use a connection string, you must use specific connection string attributes. See ["Using a Connection String" on page 778](#) and [Table 22-1](#) for an alphabetical list of driver connection string attributes and their initial default values.

#### Data Source Configuration in the UNIX/Linux odbc.ini File

On UNIX and Linux, you must set up the proper ODBC environment before configuring data sources. See ["Environment-Specific Information" on page 45](#) for basic setup information and ["Environment Variables" on page 97](#) for more detail about this procedure.

You can configure and modify data sources by editing the odbc.ini file and storing default connection values there. See ["Configuration Through the System Information \(odbc.ini\) File" on page 103](#) for detailed information about the specific steps necessary to configure a data source.

Table 22-1 lists driver connection string attributes that must be used in the `odbc.ini` file to set the value of the attributes. Note that only the long name of the attribute can be used in the file. The default listed in the table is the initial default value when the driver is installed.

## Data Source Configuration through a GUI

On Windows, data sources are stored in the Windows Registry. You can configure and modify data sources through the ODBC Administrator using a driver Setup dialog box, as described in this section.

When the driver is first installed, the values of its connection options are set by default. These values appear on the driver Setup dialog box tabs when you create a new data source. You can change these default values by modifying the data source. In the following procedure, the description of each tab is followed by a table that lists the connection options for that tab and their initial default values. This table links you to a complete description of the options and their connection string attribute equivalents. The connection string attributes are used to override the default values of the data source if you want to change these values at connection time.

### To configure a Sybase IQ data source:

- 1 Start the ODBC Administrator by selecting its icon from the DataDirect Connect program group.



- 2 Select a tab:

- **User DSN:** If you are configuring an existing user data source, select the data source name and click **Configure** to display the driver Setup dialog box.

If you are configuring a new user data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.



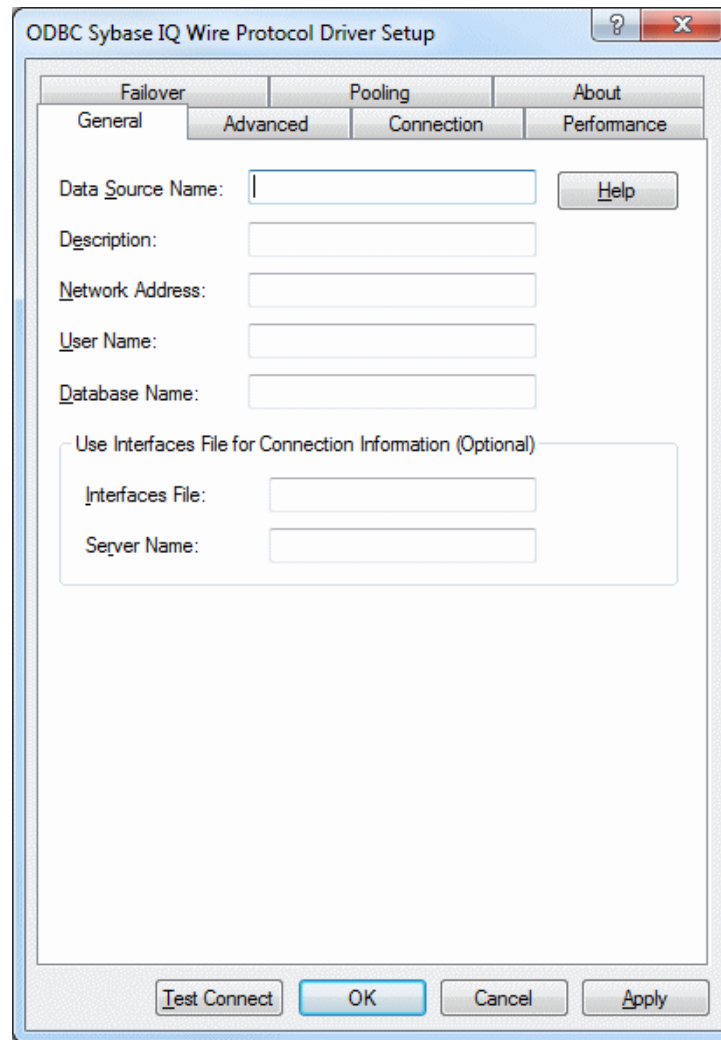
- **System DSN:** If you are configuring an existing system data source, select the data source name and click **Configure** to display the driver Setup dialog box.

If you are configuring a new system data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.

- **File DSN:** If you are configuring an existing file data source, select the data source file and click **Configure** to display the driver Setup dialog box.

If you are configuring a new file data source, click **Add** to display a list of installed drivers; then, select a driver. Click **Advanced** if you want to specify attributes; otherwise, click **Next** to proceed. Specify a name for the data source and click **Next**. Verify the data source information; then, click **Finish** to display the driver Setup dialog box.

The General tab of the Setup dialog box appears by default.

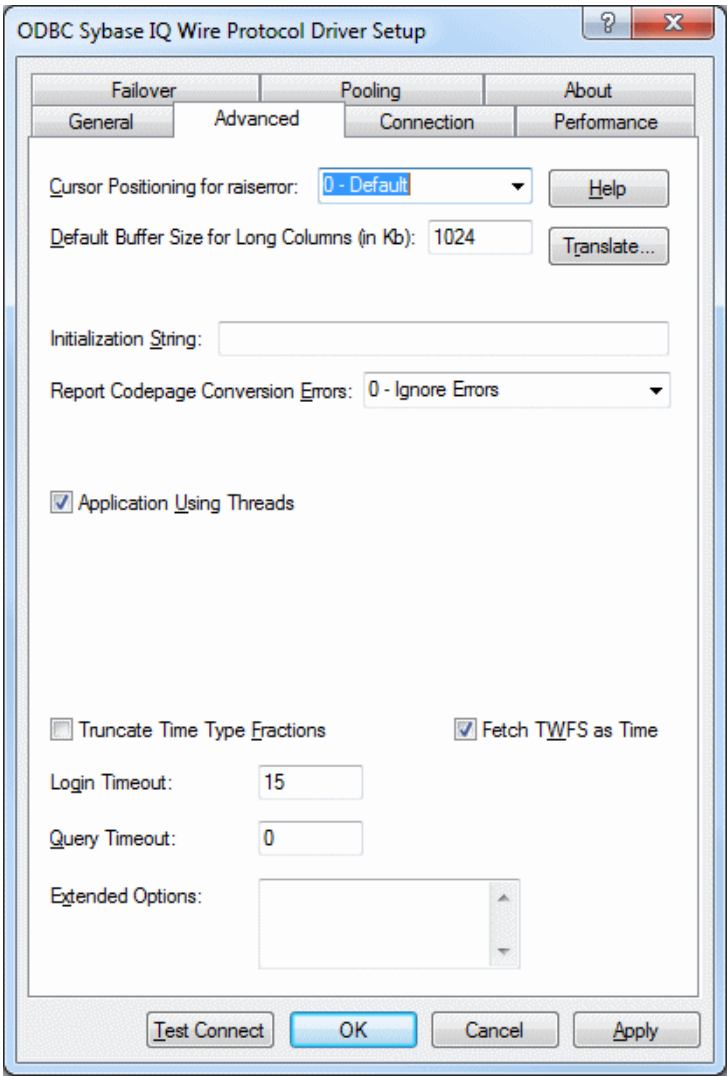


NOTE: The General tab displays the only fields that are required for creating a data source. The fields on all other tabs are optional, unless noted otherwise.

- 3 On this tab, provide values for the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: General	Default
<a href="#">"Data Source Name" on page 785</a>	None
<a href="#">Description (see page 786)</a>	None
<a href="#">Network Address (see page 793)</a>	None
<a href="#">User Name (see page 796)</a>	None
<a href="#">Database Name (see page 786)</a>	None
<a href="#">Interfaces File (see page 790)</a>	None
<a href="#">Server Name (see page 796)</a>	None

- 4 Optionally, click the **Advanced** tab to specify additional data source settings.



On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Advanced	Default
<a href="#">Cursor Positioning for Raiserror (see page 785)</a>	0 - Default
<a href="#">Default Buffer Size for Long Columns (in Kb) (see page 786)</a>	1024
<a href="#">Initialization String (see page 790)</a>	None
<a href="#">Report Codepage Conversion Errors (see page 795)</a>	0 - Ignore Errors
<a href="#">Application Using Threads (see page 782)</a>	Enabled
<a href="#">Fetch TWFS as Time (see page 788)</a>	Enabled
<a href="#">Truncate Time Type Fractions (see page 796)</a>	Disabled
<a href="#">Login Timeout (see page 792)</a>	15
<a href="#">Query Timeout (see page 794)</a>	0

**Extended Options:** Type a semi-colon separated list of connection options and their values. Use this configuration option to set the value of undocumented connection options that are provided by Progress DataDirect customer support. You can include any valid connection option in the Extended Options string, for example:

```
Database=Server1;UndocumentedOption1=value[;UndocumentedOption2=value;]
```

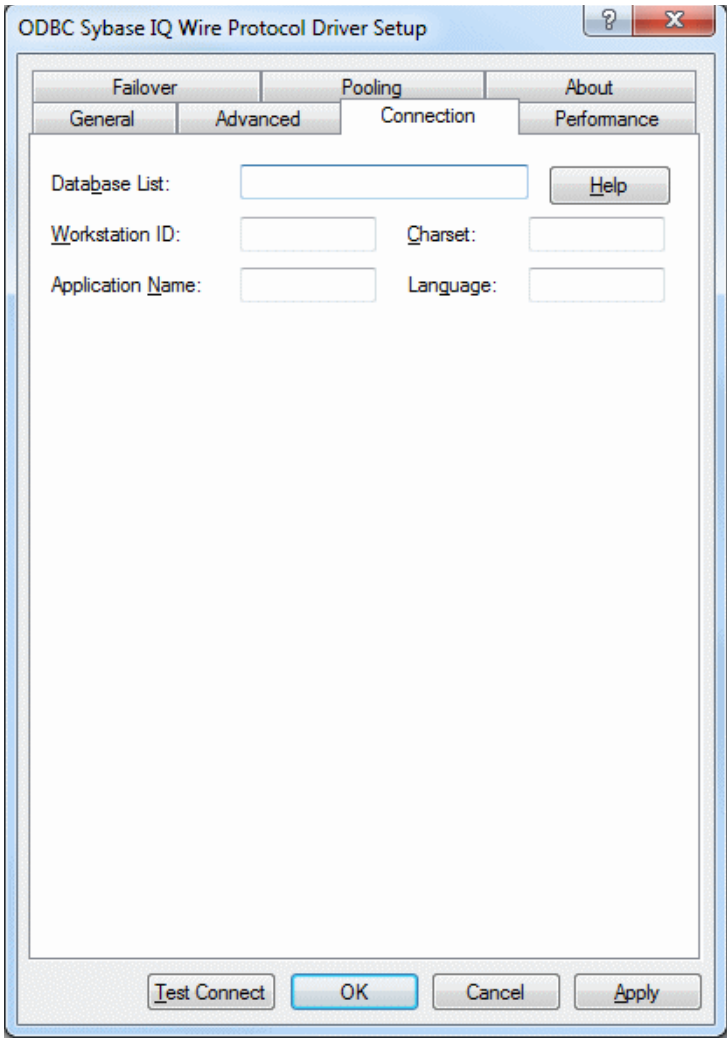
If the Extended Options string contains option values that are also set in the setup dialog or data source, the values of the options specified in the Extended Options string take precedence. However, connection options that are specified on a connection string override any option value specified in the Extended Options string.

**NOTE:** Do not specify the Extended Options configuration option in a connection string, or the driver will return an error. Instead, applications should specify the individual undocumented connection options in the connection string.

**Translate:** Click **Translate** to display the Select Translator dialog box, which lists the translators specified in the ODBC Translators section of the Registry. Progress 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.

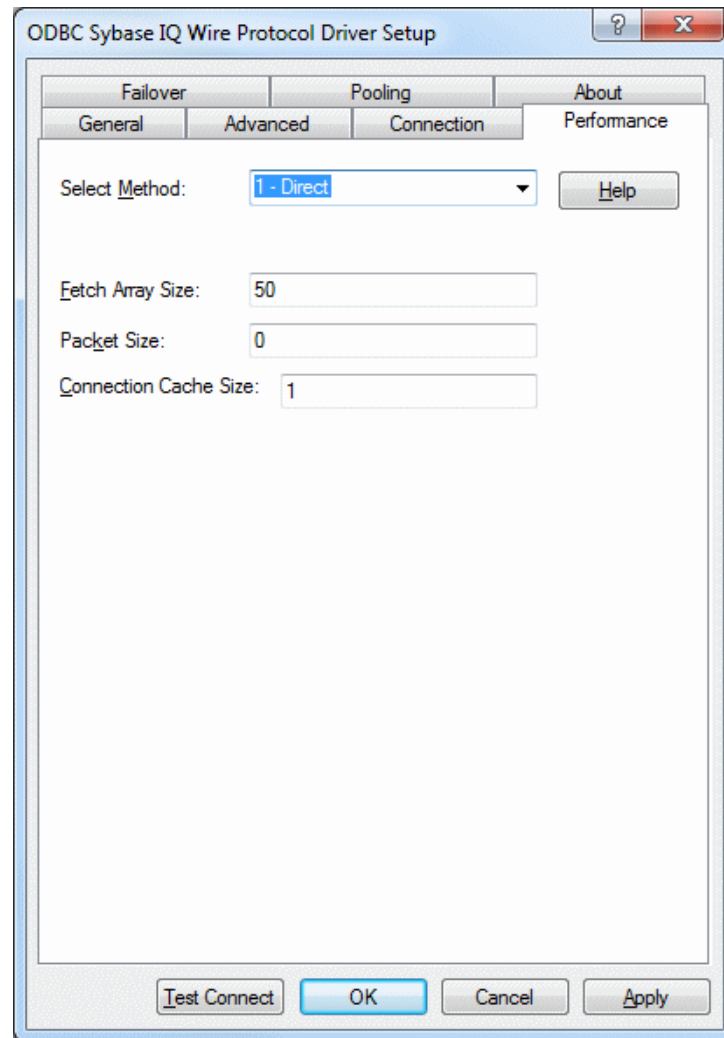
- 5 Optionally, click the **Connection** tab to specify data source settings.



On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Connection	Default
<a href="#">Database List (see page 785)</a>	None
<a href="#">Workstation ID (see page 797)</a>	None
<a href="#">Charset (see page 782)</a>	None
<a href="#">Application Name (see page 782)</a>	None
<a href="#">Language (see page 791)</a>	None

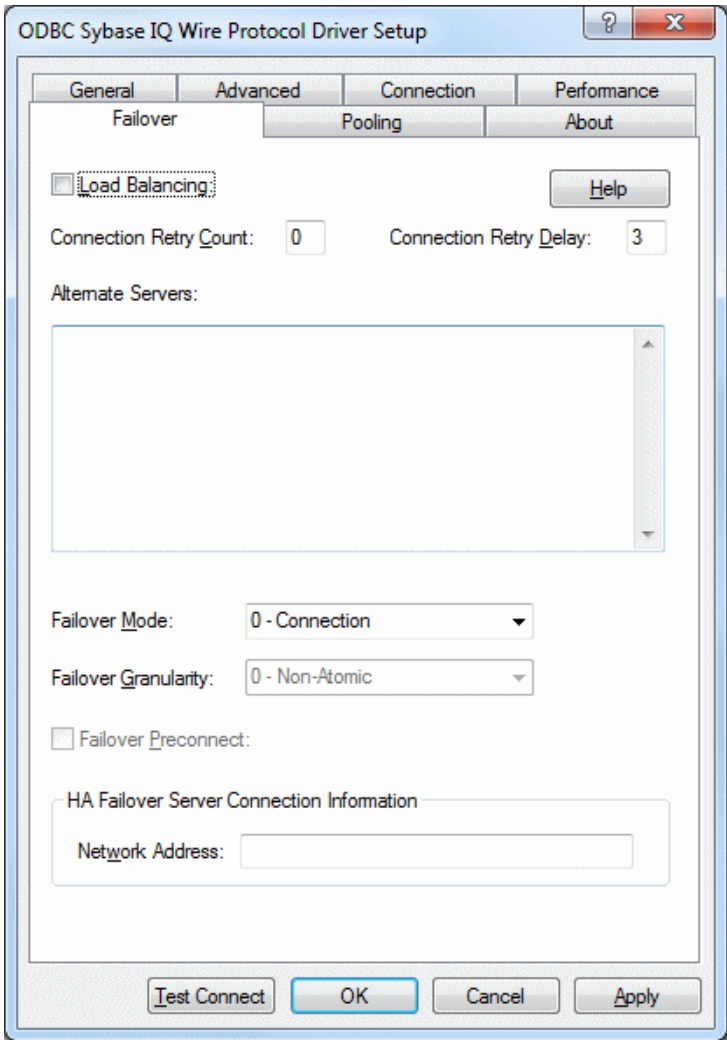
- 6 Optionally, click the **Performance** tab to specify performance data source settings.



On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Performance	Default
<a href="#">Select Method (see page 795)</a>	1 - Direct
<a href="#">Fetch Array Size (see page 788)</a>	50
<a href="#">Packet Size (see page 793)</a>	0
<a href="#">Connection Cache Size (see page 783)</a>	1

- 7 Optionally, click the **Failover** tab to specify failover data source settings.



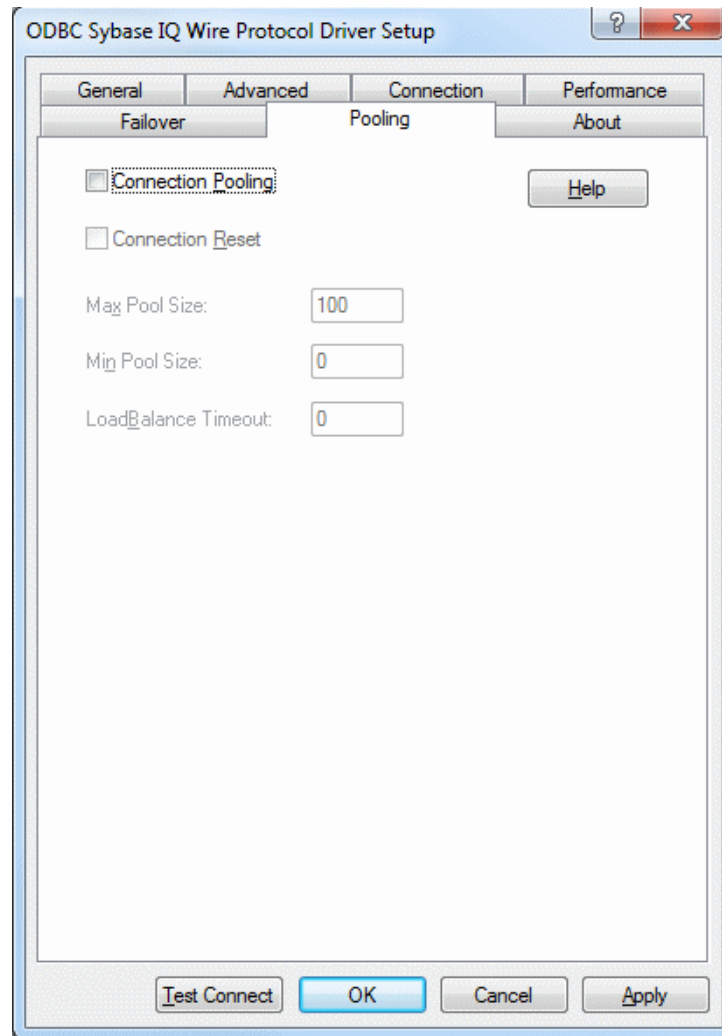
See ["Using Failover" on page 65](#) for a general description of failover and its related connection options.

On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Failover	Default
<a href="#">Load Balancing (see page 791)</a>	Disabled
<a href="#">Connection Retry Count (see page 784)</a>	0
<a href="#">Connection Retry Delay (see page 784)</a>	3
<a href="#">Alternate Servers (see page 781)</a>	None
<a href="#">Failover Mode (see page 787)</a>	0 - Connection
<a href="#">Failover Granularity (see page 787)</a>	0 - Non-Atomic
<a href="#">Failover Preconnect (see page 788)</a>	Disabled
<a href="#">HA Failover Server Connection Information/Network Address (see page 789)</a>	None



- 8 Optionally, click the **Pooling** tab to specify connection pooling data source settings.



See ["Using DataDirect Connection Pooling" on page 82](#) for a general description of connection pooling.

On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Pooling	Default
<a href="#">Connection Pooling (see page 783)</a>	Disabled
<a href="#">Connection Reset (see page 784)</a>	Disabled
<a href="#">Max Pool Size (see page 792)</a>	100
<a href="#">Min Pool Size (see page 793)</a>	0
<a href="#">Load Balance Timeout (see page 791)</a>	0

- 9 At any point during the configuration process, you can click **Test Connect** to attempt to connect to the data source using the connection options specified in the driver Setup dialog box. A logon dialog box appears (see ["Using a Logon Dialog Box" on page 779](#) for details). 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 incorrect environment or connection value, it displays an appropriate error message. Click **OK**.

NOTE: If you are configuring alternate servers for use with the connection failover feature, be aware that the Test Connect button tests only the primary server, not the alternate servers.
- 10 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.

## Using a Connection String

If you want to use a connection string for connecting to a database, or if your application requires it, you must specify either a DSN (data source name), a File DSN, or a DSN-less connection in the string. The difference is whether you use the `DSN=`, `FILEDSN=`, or the `DRIVER=` keyword in the connection string, as described in the ODBC specification. A DSN or FILEDSN connection string tells the driver where to find the default connection information. Optionally, you may specify *attribute=value* pairs in the connection string to override the default values stored in the data source.

The DSN connection string has the form:

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

The FILEDSN connection string has the form:

```
FILEDSN=filename.dsn[;attribute=value[;attribute=value]...]
```

The DSN-less connection string specifies a driver instead of a data source. All connection information must be entered in the connection string because the information is not stored in a data source.

The DSN-less connection string has the form:

```
DRIVER=[{ }driver_name[ } ][;attribute=value[;attribute=value]...]
```

[Table 22-1](#) lists the long and short names for each attribute, as well as the initial default value when the driver is first installed. You can specify either long or short names in the connection string.

An example of a DSN connection string with overriding attribute values for Sybase IQ is:

```
DSN=SYBIQTABES;DB=PAYROLL;UID=JOHN;PWD=XYZZY
```

A FILEDSN connection string is similar except for the initial keyword:

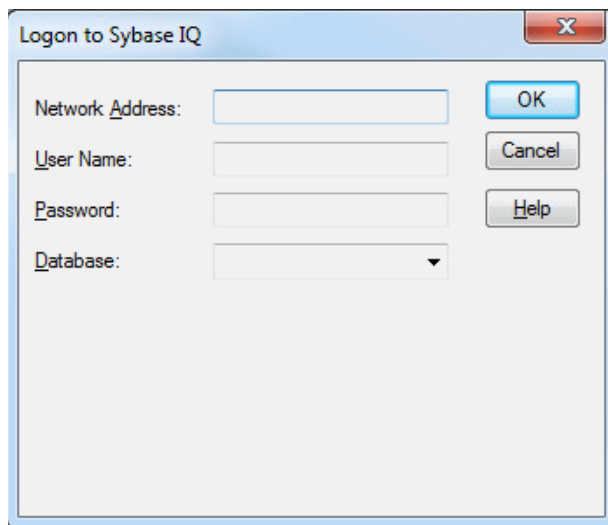
```
FILEDSN=SYBIQ.dsn;DB=PAYROLL;UID=JOHN;PWD=XYZZY
```

A DSN-less connection string must provide all necessary connection information:

```
DRIVER=DataDirect 7.1 Sybase IQ Wire Protocol;NA=123.456.78.90, 2638;  
DB=SYBIQACCT;UID=JOHN;PWD=XYZZY
```

## 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 the Logon dialog box, provide the following information:

- 1 In the Network Address field, specify an IP address for the Sybase IQ server as follows:  
*IP address,port\_number*. For example, you might enter 199.226.224.34,2638. If your network supports named servers, you can specify an address as:  
*servername,port\_number*. For example, you might enter SybIQserver,2638.

The IP address can be specified in either IPv4 or IPv6 format, or a combination of the two. See ["Using IP Addresses" on page 54](#) for details concerning these formats.

- 2 If required, type your case-sensitive login ID.
- 3 If required, type your case-sensitive password for the system.
- 4 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 that you specified on the Connection tab of the ODBC Sybase IQ Wire Protocol driver Setup dialog box.

NOTE: If you are connecting through the **Test Connect** button of the Setup dialog box, only the default database specified on the General tab of the Setup dialog box is available in the Database drop-down list. The database names specified on the Connection tab are not available.

- 5 Click **OK** to complete the logon and to update the values in the Registry.

## Connection Option Descriptions

The following connection option descriptions are listed alphabetically by the GUI name that appears on the driver Setup dialog box. The connection string attribute name, along with its short name, is listed immediately underneath the GUI name. For example:

### Application Using Threads

Attribute    ApplicationUsingThreads (AUT)

In most cases, the GUI name and the attribute name are the same; however, some exceptions exist. If you need to look up an option by its connection string attribute name, please refer to the alphabetical table of connection string attribute names.

Also, a few connection string attributes, for example, Password, do not have equivalent options that appear on the GUI. They are in the list of descriptions alphabetically by their attribute names.

[Table 22-1](#) lists the connection string attributes supported by the Sybase IQ Wire Protocol driver.

---

**Table 22-1. Sybase IQ Wire Protocol Attribute Names**

---

Attribute (Short Name)	Default
<a href="#">AlternateServers (ASRV)</a>	None
<a href="#">ApplicationName (APP)</a>	None
<a href="#">ApplicationUsingThreads (AUT)</a>	1 (Enabled)
<a href="#">ArraySize (AS)</a>	50
<a href="#">Charset (CS)</a>	None
<a href="#">ConnectionReset (CR)</a>	0 (Disabled)
<a href="#">ConnectionRetryCount (CRC)</a>	0
<a href="#">ConnectionRetryDelay (CRD)</a>	3
<a href="#">ConnectionCacheSize (CCS)</a>	1
<a href="#">Database (DB)</a>	None
<a href="#">Database List</a>	None
<a href="#">DataSourceName (DSN)</a>	None
<a href="#">DefaultLongDataBuffLen (DLDBL)</a>	1024
<a href="#">Description (n/a)</a>	None
<a href="#">FailoverGranularity (FG)</a>	0 (Non-Atomic)
<a href="#">FailoverMode (FM)</a>	0 (Connection)
<a href="#">FailoverPreconnect (FP)</a>	0 (Disabled)
<a href="#">FetchTWFSasTime (FTWFSAT)</a>	1 (Enabled)
<a href="#">IANAAppCodePage (IACP)</a>	4 (ISO 8559-1 Latin-1)
UNIX ONLY	
<a href="#">InitializationString (IS)</a>	None
<a href="#">InterfacesFile (IF)</a>	None

**Table 22-1. Sybase IQ Wire Protocol Attribute Names** (cont.)

Attribute (Short Name)	Default
<a href="#">InterfacesFileName (IFSN)</a>	None
<a href="#">Language (LANG)</a>	None
<a href="#">LoadBalanceTimeout (LBT)</a>	0
<a href="#">LoadBalancing (LB)</a>	0 (Disabled)
<a href="#">LoginTimeout (LT)</a>	15
<a href="#">LogonID (UID)</a>	None
<a href="#">MaxPoolSize (MXPS)</a>	100
<a href="#">MinPoolSize (MNPS)</a>	0
<a href="#">NetworkAddress (NA)</a>	None
<a href="#">PacketSize (PS)</a>	0
<a href="#">Password (PWD)</a>	None
<a href="#">Pooling (POOL)</a>	0 (Disabled)
<a href="#">QueryTimeout (QT)</a>	0
<a href="#">ReportCodepageConversionErrors (RCCE)</a>	0 (Ignore Errors)
<a href="#">SelectMethod (SM)</a>	1 (Direct)
<a href="#">TruncateTimeTypeFractions (TTTF)</a>	0 (Disabled)
<a href="#">WorkstationID (WKID)</a>	None

## Alternate Servers

Attribute	AlternateServers (ASRV)
Description	A list of alternate database servers to which the driver tries to connect if the primary database server is unavailable. Specifying a value for this option enables connection failover for the driver. The value you specify must be in the form of a string that defines the physical location of each alternate server. All of the other required connection information for each alternate server is the same as what is defined for the primary server connection.
Valid Values	<p><code>{{NetworkAddress=addressvalue   InterfacesFileName=sectionvalue}[ , ...]}</code></p> <p>NetworkAddress and InterfacesFileName can be used in the same string.</p> <p>NOTE: An alternate server address in IPv6 format must be enclosed in double quotation marks.</p> <p>You must specify the network address of each alternate database server or the section in the Interfaces file that contains the network connection information for the Sybase IQ database server you want to access (InterfacesFileName).</p> <p>NOTE: The Alternate Servers option and the HA Failover Server Connection Information option are mutually exclusive.</p>

**Example** The following example Alternate Servers values define three alternate database servers for connection failover:

```
(InterfacesFileName=Accounting, NetworkAddress="255.125.1.11, 4200",
NetworkAddress="SybaseIQ2, 4200")
```

In this example, the network address of the last two alternates contain commas. In this case, enclose the network address with double quotation marks as shown.

**Default** None

**GUI Tab** [Failover tab](#) on page 776

## Application Name

**Attribute** ApplicationName (APP)

**Description** The name used by Sybase IQ to identify your application.

**Valid Values** *string*

where *string* is a valid application name.

**Default** None

**GUI Tab** [Connection tab](#) on page 774

## Application Using Threads

**Attribute** ApplicationUsingThreads (AUT)

**Description** Determines whether the driver works with applications using multiple ODBC threads.

This connection option can affect performance. See ["Performance Considerations" on page 797](#) for details.

**Valid Values** 0 | 1

If set to 1 (Enabled), the driver works with single-threaded and multi-threaded applications.

If set to 0 (Disabled), the driver does not work with multi-threaded applications. If using the driver with single-threaded applications, this value avoids additional processing required for ODBC thread-safety standards.

**Default** 1 (Enabled)

**GUI Tab** [Advanced tab](#) on page 772

## Charset

**Attribute** Charset (CS)

**Description** The name of a character set installed on the Sybase IQ server to be used by the driver.

This option is not a substitute for the IANAAppCodePage option. See [IANAAppCodePage](#) for details.

Valid Values *charset*

where *charset* is the name of a character set installed on the Sybase IQ server.

If unspecified, the character set setting on the Sybase IQ server is used.

For the driver to return Unicode SQL types for connections to Sybase IQ 15.0 and higher, use a value of UTF-8. Refer to the Sybase IQ server documentation for a list of valid character sets.

Example If your client needs to receive data in iso-8859-1 from a non-Unicode Sybase IQ server, you would specify a value of iso\_1.

Default None

GUI Tab [Connection tab](#) on page 774

## Connection Cache Size

Attribute ConnectionCacheSize (CCS)

Description The number of connections that the connection cache can hold.

Valid Values *x*

where *x* is a positive integer representing the number of connections that the connection cache can hold.

To enable the connection cache, you must set the Select Method option to 1 (Direct). Increasing the connection cache may increase performance of some applications but requires additional database resources.

Default 1

GUI Tab [Performance tab](#) on page 775

## Connection Pooling

Attribute Pooling (POOL)

Description Specifies whether to use the driver's connection pooling.

NOTE: The application must be thread-enabled to use connection pooling.

This connection option can affect performance. See ["Performance Considerations" on page 797](#) for details.

Valid Values 0 | 1

If set to 1 (Enabled), the driver uses connection pooling.

If set to 0 (Disabled), the driver does not use connection pooling.

Default 0 (Disabled)

GUI Tab [Pooling tab](#) on page 777

## Connection Reset

Attribute	ConnectionReset (CR)
Description	<p>Determines whether the state of connections that are removed from the connection pool for reuse by the application is reset to the initial configuration of the connection.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 797</a> for details.</p>
Valid Values	<p>0   1</p> <p>If set to 1 (Enabled), the state of connections removed from the connection pool for reuse by an application is reset to the initial configuration of the connection. Resetting the state can negatively impact performance because additional commands must be sent over the network to the server to reset the state of the connection.</p> <p>If set to 0 (Disabled), the state of connections is not reset.</p>
Default	0 (Disabled)
GUI Tab	<a href="#">Pooling tab</a> on page 777

## Connection Retry Count

Attribute	ConnectionRetryCount (CRC)
Description	<p>The number of times the driver retries connection attempts to the primary database server, and if specified, alternate servers until a successful connection is established.</p> <p>This option and the Connection Retry Delay connection option, which specifies the wait interval between attempts, can be used in conjunction with failover.</p>
Valid Values	<p>0   <i>x</i></p> <p>where <i>x</i> is a positive integer from 1 to 65535.</p> <p>If set to 0, the driver does not try to connect after the initial unsuccessful attempt.</p> <p>If set to <i>x</i>, the driver retries connection attempts the specified number of times. If a connection is not established during the retry attempts, the driver returns an error that is generated by the last server to which it tried to connect.</p>
Default	0
GUI Tab	<a href="#">Failover tab</a> on page 776

## Connection Retry Delay

Attribute	ConnectionRetryDelay (CRD)
Description	<p>The number of seconds the driver waits between connection retry attempts when Connection Retry Count is set to a positive integer.</p> <p>This option and the Connection Retry Count connection option can be used in conjunction with failover.</p>



Valid Values	0   $x$  where $x$ is a positive integer from 1 to 65535.  If set to 0, there is no delay between retries.  If set to $x$ , the driver waits the specified number of seconds between connection retry attempts.
Default	3
GUI Tab	<a href="#">Failover tab</a> on page 776

## Cursor Positioning for Raiserror

Attribute	RaiseErrorPositionBehavior (REPB)
Description	Determines whether the driver returns raiserrors when the next statement is executed or handles them separately.
Valid Values	0   1  If set to 0 (Default), raiserrors are handled separately from surrounding statements. The error is returned when a raiserror is processed (for example, resulting from SQLExecute, SQLExecDirect, or SQLMoreResults). The result set is empty.  If set to 1 (Microsoft compatible), raiserrors are returned when the next statement is processed, and the cursor is positioned on the first row of the subsequent result set. This could result in multiple raiserrors being returned on a single execute.
Default	0 (Default)
GUI Tab	<a href="#">Advanced tab</a> on page 785

## Data Source Name

Attribute	DataSourceName (DSN)
Description	The name of a data source in your Windows Registry or odbc.ini file.
Valid Values	<i>string</i>  where <i>string</i> is the name of a data source.
Default	None
GUI Tab	<a href="#">General tab</a> on page 785

## Database List

Attribute	n/a
Description	A list of database names that will appear in the drop-down list of the logon dialog box (see <a href="#">"Using a Logon Dialog Box" on page 779</a> for a description).
Valid Values	<i>database_list</i>  where <i>database_list</i> is a comma-separated list of database names that will appear in the drop-down list of the logon dialog box.

Default None  
 GUI Tab [Connection tab](#) on page 774

## Database Name

Attribute Database (DB)  
 Description The name of the database to which you want to connect.  
 Valid Values *database\_name*

where *database\_name* is the name of a valid database. If you do not specify a value, the default is the database defined by the system administrator for each user.

Default None  
 GUI Tab [General tab](#) on page 771

## Description

Attribute Description (n/a)  
 Description An optional long description of a data source. This description is not used as a runtime connection attribute, but does appear in the ODBC.INI section of the Registry and in the odbcc.ini file.

Valid Values *string*

where *string* is a description of a data source.

Default None  
 GUI Tab [General tab](#) on page 771

## Default Buffer Size for Long Columns (in Kb)

Attribute DefaultLongDataBuffLen (DLDBL)  
 Description The maximum length of data (in KB) the driver can fetch from long columns in a single round trip and the maximum length of data that the driver can send using the SQL\_DATA\_AT\_EXEC parameter.

This option also applies to binding long parameters in chunks. The driver truncates any data passed in a Long/LOB SQL\_DATA\_AT\_EXEC parameter to the size specified.

This connection option can affect performance. See ["Performance Considerations" on page 797](#) for details.

Valid Values An integer in multiples of 1024

The value must be in multiples of 1024 (for example, 1024, 2048). You need to increase the default value if the total size of any Long data exceeds 1 MB. This value is multiplied by 1024 to determine the total maximum length of fetched data. For example, if you enter a value of 2048, the maximum length of data would be 1024 x 2048, or 2097152 (2 MB).

Default 1024  
 GUI Tab [Advanced tab](#) on page 772

## Failover Granularity

Attribute	FailoverGranularity (FG)
Description	<p>Determines whether the driver fails the entire failover process or continues with the process if errors occur while trying to reestablish a lost connection.</p> <p>This option applies only when Failover Mode is set to 1 (Extended Connection) or 2 (Select).</p> <p>The Alternate Servers option specifies one or multiple alternate servers for failover and is required for all failover methods.</p>
Valid Values	<p>0   1   2   3</p> <p>If set to 0 (Non-Atomic), the driver continues with the failover process and posts any errors on the statement on which they occur.</p> <p>If set to 1 (Atomic) the driver fails the entire failover process if an error is generated as the result of anything other than executing and repositioning a Select statement. If an error is generated as a result of repositioning a result set to the last row position, the driver continues with the failover process, but generates a warning that the Select statement must be reissued.</p> <p>If set to 2 (Atomic Including Repositioning), the driver fails the entire failover process if any error is generated as the result of restoring the state of the connection or the state of work in progress.</p> <p>If set to 3 (Disable Integrity Check), the driver does not verify that the rows that were restored during the failover process match the original rows. This value applies only when Failover Mode is set to 2 (Select).</p>
Default	0 (Non-Atomic)
GUI Tab	<a href="#">Failover tab</a> on page 776

## Failover Mode

Attribute	FailoverMode (FM)
Description	<p>Specifies the type of failover method the driver uses.</p> <p>The Alternate Servers option specifies one or multiple alternate servers for failover and is required for all failover methods.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 797</a> for details.</p>
Valid Values	<p>0   1   2</p> <p>If set to 0 (Connection), the driver provides failover protection for new connections only.</p> <p>If set to 1 (Extended Connection), the driver provides failover protection for new and lost connections, but not any work in progress.</p> <p>If set to 2 (Select), the driver provides failover protection for new and lost connections. In addition, it preserves the state of work performed by the last Select statement executed.</p>

Default 0 (Connection)  
 GUI Tab [Failover tab](#) on page 776

## Failover Preconnect

Attribute FailoverPreconnect (FP)

Description Specifies whether the driver tries to connect to the primary and an alternate server at the same time.

This attribute applies only when Failover Mode is set to 1 (Extended Connection) or 2 (Select) and at least one alternate server is specified.

The Alternate Servers option specifies one or multiple alternate servers for failover and is required for all failover methods.

Valid Values 0 | 1

If set to 0 (Disabled), the driver tries to connect to an alternate server only when failover is caused by an unsuccessful connection attempt or a lost connection. This value provides the best performance, but your application typically experiences a short wait while the failover connection is attempted.

If set to 1 (Enabled), the driver tries to connect to the primary and an alternate server at the same time. This can be useful if your application is time-sensitive and cannot absorb the wait for the failover connection to succeed.

Default 0 (Disabled)

GUI Tab [Failover tab](#) on page 776

## Fetch Array Size

Attribute ArraySize (AS)

Description The number of rows the driver retrieves from the server for a fetch. This is not the number of rows given to the user. You should use Fetch Array Size in conjunction with [Select Method](#).

This connection option can affect performance. See "[Performance Considerations](#)" on [page 797](#) for details.

Valid Values  $x$

where  $x$  is a positive integer specifying the number of rows.

Default 50

GUI Tab [Performance tab](#) on page 775

## Fetch TWFS as Time

Attribute FetchTWFSasTime (FTWFSAT)

Description Determines whether the driver returns column values with the time data type as the ODBC data type SQL\_TYPE\_TIME or SQL\_TYPE\_TIMESTAMP.

Valid Values 0 | 1

If set to 1 (Enabled), the driver returns column values with the time data type as the ODBC data type SQL\_TYPE\_TIME. The fractional seconds portion of the value is truncated.

If set to 0 (Disabled), the driver returns column values with the time data type as the ODBC data type SQL\_TYPE\_TIMESTAMP. The fractional seconds portion of the value is preserved. Time columns are not searchable when they are described and fetched as timestamp

NOTE: When returning time with fractional seconds data as SQL\_TYPE\_TIMESTAMP, the Year, Month and Day parts of the timestamp must be set to zero.

Default 1 (Enabled)

GUI Tab [Advanced tab](#) on page 772

## HA Failover Server Connection Information/Network Address

Attribute FailoverNetworkAddress (FNA)

Description The network 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 specified HA Failover server. This option is valid only for Sybase IQ servers that have the High Availability Failover feature enabled.

Valid Values *IP\_address, port\_number | server\_name, port\_number*

where

*IP\_address* is the IP address that uniquely identifies the HA Failover server.

*port\_number* is the port number assigned to the listener process on the HA Failover server.

*server\_name* is a name that uniquely identifies the HA Failover server. You can use this format if your environment supports named servers.

NOTE: The HA Failover Server Connection Information option and the Alternate Servers option are mutually exclusive.

Examples 199.226.224.34, 2638  
Sybaseiqserver, 2638

Default None

GUI Tab [Failover tab](#) on page 776



## IANAAppCodePage

Attribute IANAAppCodePage (IACP)

Description An Internet Assigned Numbers Authority (IANA) value. You must specify a value for this option if your application is not Unicode-enabled or if your database character set is not Unicode. Refer to [Chapter 4 “Internationalization, Localization, and Unicode”](#) in the *DataDirect Connect Series for ODBC Reference* for details.

The driver uses the specified IANA code page to convert "W" (wide) functions to ANSI.

The driver and Driver Manager both check for the value of IANAAppCodePage in the following order:

- In the connection string
- In the Data Source section of the system information file (odbc.ini)
- In the ODBC section of the system information file (odbc.ini)

If the driver does not find an IANAAppCodePage value, the driver uses the default value of 4 (ISO 8859-1 Latin-1).

Valid Values *IANA\_code\_page*

where *IANA\_code\_page* is one of the valid values listed in [Chapter 1 “Values for the Attribute IANAAppCodePage”](#) in the *DataDirect Connect Series for ODBC Reference*. The value must match the database character encoding and the system locale.

Default 4 (ISO 8559-1 Latin-1)

GUI Tab N/A

## Initialization String

Attribute InitializationString (IS)

Description A SQL command that is issued immediately after connecting to the database to manage session settings.

NOTE: If the statement fails to execute, the connection fails and the driver reports the error returned from the server.

Valid Values *SQL\_command*

where *SQL\_command* is a valid SQL command that is supported by the database.

Example To allow delimited identifiers, specify:

```
Initialization String=set QUOTED_IDENTIFIER on
```

Default None

GUI Tab [Advanced tab](#) on page 772



## Interfaces File

Attribute InterfacesFile (IF)

Description The directory to the Interfaces file.

NOTE: This option and the Network Address option are mutually exclusive.

Valid Values *file\_dir*

where *file\_dir* is the directory to the Interfaces file.

If unspecified and a value is specified for the Server Name option, the driver looks for the path name of the Interfaces file in the Registry under HKEY\_LOCAL\_MACHINE\SOFTWARE\DataDirect\InterfacesFile. If this Registry value

is empty, the driver will try to open the SQL.INI file found in the same directory where the driver is located and use it as the Interfaces file.

Default None  
 GUI Tab [General tab](#) on page 771

## Language

Attribute Language (LANG)  
 Description The national character set installed on the Sybase IQ server.  
 Valid Values *charset*  
 where *charset* is the national character set installed on the Sybase IQ server.  
 Default None (English)  
 GUI Tab [Connection tab](#) on page 774

## Load Balance Timeout

Attribute LoadBalanceTimeout (LBT)  
 Description The number of seconds to keep inactive connections open in a connection pool. An inactive connection is a database session that is not associated with an ODBC connection handle, that is, a connection in the pool that is not in use by an application.  
 NOTE: The Min Pool Size option may cause some connections to ignore this value.  
 This connection option can affect performance. See "[Performance Considerations](#)" on [page 797](#) for details.  
 Valid Values 0 | *x*  
 where *x* is a positive integer that specifies a number of seconds.  
 If set to 0, inactive connections are kept open.  
 If set to *x*, inactive connections are closed after the specified number of seconds passes.  
 Default 0 (Disabled)  
 GUI Tab [Pooling tab](#) on page 777

## Load Balancing

Attribute LoadBalancing (LB)  
 Description Determines whether the driver uses client load balancing in its attempts to connect to the database servers (primary and alternate). You can specify one or multiple alternate servers by setting the Alternate Servers option.  
 Valid Values 0 | 1  
 If set to 1 (Enabled), the driver uses client load balancing and attempts to connect to the database servers (primary and alternate servers) in random order.

If set to 0 (Disabled), the driver does not use client load balancing and connects to each server based on their sequential order (primary server first, then, alternate servers in the order they are specified).

NOTE: This option has no effect unless alternate servers are defined for the Alternate Servers connection option.

Default 0 (Disabled)

GUI Tab [Failover tab](#) on page 776

## Login Timeout

Attribute LoginTimeout (LT)

Description The number of seconds the driver waits for a connection to be established before returning control to the application and generating a timeout error. To override the value that is set by this connection option for an individual connection, set a different value in the SQL\_ATTR\_LOGIN\_TIMEOUT connection attribute using the SQLSetConnectAttr() function.

Valid Values -1 | 0 |  $x$

where  $x$  is a positive integer that specifies a number of seconds.

If set to -1, the connection request does not time out. The driver silently ignores the SQL\_ATTR\_LOGIN\_TIMEOUT attribute.

If set to 0, the connection request does not time out, but the driver responds to the SQL\_ATTR\_LOGIN\_TIMEOUT attribute.

If set to  $x$ , the connection request times out after the specified number of seconds unless the application overrides this setting with the SQL\_ATTR\_LOGIN\_TIMEOUT attribute.

Default 15

GUI Tab [Advanced tab](#) on page 772

## Max Pool Size

Attribute MaxPoolSize (MXPS)

Description The maximum number of connections allowed within a single connection pool. When the maximum number of connections is reached, no additional connections can be created in the connection pool.

This connection option can affect performance. See ["Performance Considerations" on page 797](#) for details.

Valid Values An integer from 1 to 65535

For example, if set to 20, the maximum number of connections allowed in the pool is 20.

Default 100

GUI Tab [Pooling tab](#) on page 777



## Min Pool Size

Attribute	MinPoolSize (MNPS)
Description	<p>The minimum number of connections that are opened and placed in a connection pool, in addition to the active connection, when the pool is created. The connection pool retains this number of connections, even when some connections exceed their Load Balance Timeout value.</p> <p>This connection option can affect performance. See <a href="#">"Performance Considerations" on page 797</a> for details.</p>
Valid Values	<p>0   <i>x</i></p> <p>where <i>x</i> is an integer from 1 to 65535.</p> <p>For example, if set to 5, the start-up number of connections in the pool is 5 in addition to the current existing connection.</p> <p>If set to 0, no connections are opened in addition to the current existing connection.</p>
Default	0
GUI Tab	<a href="#">Pooling tab</a> on page 777

## Network Address

Attribute	NetworkAddress (NA)
Description	<p>A unique identifier assigned to the Sybase IQ server machine.</p> <p>NOTE: This option is mutually exclusive with the Interfaces File and the Server Name option.</p>
Valid Values	<p><i>server_name</i>   <i>IP_address</i></p> <p>where</p> <p><i>server_name</i> is the Sybase IQ server name specified as: <i>named_server</i>, <i>port_number</i>. For example, you can enter <i>SyIQserver</i>, 2638.</p> <p><i>IP_address</i> is the Sybase IQ server address specified as: <i>IP_address</i>, <i>port_number</i>. For example, you can enter 199.226.224.34, 2638. The IP address can be specified in either IPv4 or IPv6 format, or a combination of the two. See <a href="#">"Using IP Addresses" on page 54</a> for details about these formats.</p>
Default	None
GUI Tab	<a href="#">General tab</a> on page 771

## Packet Size

Attribute	PacketSize (PS)
Description	<p>Determines the number of bytes for each database protocol packet that is transferred from the database server to the client machine. Adjusting the packet size can improve performance. The optimal value depends on the typical size of data that is inserted, updated, or returned by the application and the environment in which it is running. Typically,</p>

larger packet sizes work better for large amounts of data. For example, if an application regularly returns character values that are 10,000 characters in length, using a value of 32 (16 KB) typically results in improved performance.

NOTE: The ODBC connection attribute `SQL_ATTR_PACKET_SIZE` provides the same functionality as the Packet Size option; however, `SQL_ATTR_PACKET_SIZE` and the Packet Size option are mutually exclusive. If Packet Size is specified, the driver returns the message `Driver Not Capable` if an application attempts to call `SQLSetConnectAttr()` for `SQL_ATTR_PACKET_SIZE`. If you do not set the Packet Size option, application calls to `SQLSetConnectAttr()` for `SQL_ATTR_PACKET_SIZE` are accepted by the driver.

This connection option can affect performance. See ["Performance Considerations" on page 797](#) for details.

Valid Values	-1   0   <i>x</i>
	If set to -1, the driver uses the maximum packet size that is set by the database server.
	If set to 0, the driver uses the default packet size that is used by the database server.
	If set to <i>x</i> , an integer from 1 to 127, the driver uses a packet size that is a multiple of 512 bytes. For example, <code>PacketSize=8</code> means to set the packet size to 8 * 512 bytes (4096 bytes).
Default	0
GUI Tab	<a href="#">Performance tab</a> on page 775

**Password**

Attribute	Password (PWD)
Description	The password that the application uses to connect to your database. The Password option cannot be specified through the driver Setup dialog box and should not be stored in a data source. It is specified through the Logon dialog box or a connection string.
Valid Values	<i>pwd</i>  where <i>pwd</i> is a valid password.
Default	None
GUI Tab	n/a

**Query Timeout**

Attribute	QueryTimeout (QT)
Description	The number of seconds for the default query timeout for all statements that are created by a connection. To override the value set by this connection option for an individual statement, set a different value in the <code>SQL_ATTR_QUERY_TIMEOUT</code> statement attribute on the <code>SQLSetStmtAttr()</code> function.
Valid Values	-1   0   <i>x</i>  where <i>x</i> is a positive integer that specifies a number of seconds.

If set to -1, the query does not time out. The driver silently ignores the SQL\_ATTR\_QUERY\_TIMEOUT attribute.

If set to 0, the query does not time out, but the driver responds to the SQL\_ATTR\_QUERY\_TIMEOUT attribute.

If set to *x*, all queries time out after the specified number of seconds unless the application overrides this value by setting the SQL\_ATTR\_QUERY\_TIMEOUT attribute.

Default 0

GUI Tab [Advanced tab](#) on page 772

## Report Codepage Conversion Errors

Attribute ReportCodepageConversionErrors (RCCE)

Description Specifies how the driver handles code page conversion errors that occur when a character cannot be converted from one character set to another.

An error message or warning can occur if an ODBC call causes a conversion error, or if an error occurs during code page conversions to and from the database or to and from the application. The error or warning generated is `Code page conversion error encountered`. In the case of parameter data conversion errors, the driver adds the following sentence: `Error in parameter x`, where *x* is the parameter number. The standard rules for returning specific row and column errors for bulk operations apply.

Valid Values 0 | 1 | 2

If set to 0 (Ignore Errors), the driver substitutes 0x1A for each character that cannot be converted and does not return a warning or error.

If set to 1 (Return Error), the driver returns an error instead of substituting 0x1A for unconverted characters.

If set to 2 (Return Warning), the driver substitutes 0x1A for each character that cannot be converted and returns a warning.

Default 0 (Ignore Errors)

GUI Tab [Advanced tab](#) on page 772

## Select Method

Attribute SelectMethod (SM)

Description Determines whether database cursors are used for Select statements.

This connection option can affect performance. See "[Performance Considerations](#)" on [page 797](#) for details.

Valid Values 0 | 1

If set to 0 (Cursor), database cursors are used. In some cases performance degradation can occur when performing large numbers of sequential Select statements because of the amount of overhead associated with creating database cursors.

If set to 1 (Direct), Select statements are run directly without using database cursors, and the data source is limited to one active statement.

Default 1 (Direct)

GUI Tab [Performance tab](#) on page 775

### Server Name

Attribute InterfacesFileServerName (IFSN)

Description The name of the section in the Interfaces file containing the network connection information for the Sybase IQ server. Typically, the section name is the host name of the Sybase IQ server.

NOTE: The Network Address option and the Server Name option are mutually exclusive.

Valid Values *section\_name*

where *section\_name* is a section in the Interfaces file containing the network connection information for the Sybase IQ server.

Default None

GUI Tab [General tab](#) on page 771

### Truncate Time Type Fractions

Attribute TruncateTimeTypeFractions (TTTF)

Description Determines whether the driver sets fractional seconds to zero (0) when converting data from the TIME data type to TIMESTAMP, CHAR, or WCHAR data types.

Valid Values 0 | 1

If set to 1 (Enabled), the driver converts fractional seconds to zero when converting the TIME data type.

If set to 0 (Disabled), the driver does not set fractional seconds to zero when converting the TIME data type.

Default 0 (Disabled)

GUI Tab [Advanced tab](#) on page 772

### User Name

Attribute LogonID (UID)

Description The default user ID that is used to connect to your database. Your ODBC application may override this value or you may override it in the logon dialog box or connection string.

Valid Values *userid*

where *userid* is a valid user ID with permissions to access the database.

Default None

GUI Tab [General tab](#) on page 771

## Workstation ID

Attribute	WorkstationID (WKID)
Description	An identifier for the client machine.
Valid Values	<i>ID</i>  where <i>ID</i> is workstation ID use by the client machine.
Default	None
GUI Tab	<a href="#">Connection tab</a> on page 774

---

## Performance Considerations

The following connection options can enhance driver performance. You can also enhance performance through efficient application design. Refer to Chapter 5 “Designing ODBC Applications for Performance Optimization” in the *DataDirect Connect Series for ODBC Reference* for details.

**Application Using Threads (ApplicationUsingThreads):** The driver coordinates concurrent database operations (operations from different threads) by acquiring locks. Although locking prevents errors in the driver, it also decreases performance. If your application does not make ODBC calls from different threads, the driver has no reason to coordinate operations. In this case, the ApplicationUsingThreads attribute should be disabled (set to 0).

NOTE: If you are using a multi-threaded application, you must enable the Application Using Threads option.

**Connection Pooling (ConnectionPooling):** If you enable the driver to use connection pooling, you can set additional options that affect performance:

- **Load Balance Timeout:** You can define how long to keep connections in the pool. The time that a connection was last used is compared to the current time and, if the timespan exceeds the value of the Load Balance Timeout option, the connection is destroyed. The Min Pool Size option can cause some connections to ignore this value.
- **Connection Reset:** Resetting a re-used connection to the initial configuration settings impacts performance negatively because the connection must issue additional commands to the server.
- **Max Pool Size:** Setting the maximum number of connections that the pool can contain too low might cause delays while waiting for a connection to become available. Setting the number too high wastes resources.
- **Min Pool Size:** A connection pool is created when the first connection with a unique connection string connects to the database. The pool is populated with connections up to the minimum pool size, if one has been specified. The connection pool retains this number of connections, even when some connections exceed their Load Balance Timeout value.

**Default Buffer Size for Long/LOB Columns (DefaultLongDataBuffLen):** To improve performance when your application fetches images, pictures, or long text or binary data, a buffer size can be set to accommodate the maximum size of the data. The buffer size should only be large enough to accommodate the maximum amount of data retrieved; otherwise, performance is reduced by transferring large amounts of data into an oversized buffer. If your application retrieves more than 1 MB of data, the buffer size should be increased accordingly.

**Failover Mode (FailoverMode):** Although high availability that replays queries after a failure provides increased levels of protection, it can adversely affect performance because of increased overhead.

**Fetch Array Size (ArraySize):** If the Select Method connection option is set to 0 and your application fetches more than 50 rows at a time, you should set Fetch Array Size to the approximate number of rows being fetched. This reduces the number of round trips on the network, thereby increasing performance. For example, if your application normally fetches 200 rows, it is more efficient for the driver to fetch 200 rows at one time over the network than to fetch 50 rows at a time during four round trips over the network. You should use Fetch Array Size in conjunction with Select Method.

NOTE: The ideal setting for your application will vary. To calculate the ideal setting for this option, you must know the size in bytes of the rows that you are fetching and the size in bytes of your Network Packet. Then, you must calculate the number of rows that will fit in your Network Packet, leaving space for packet overhead. For example, suppose your Network Packet size is 1024 bytes and the row size is 8 bytes. Dividing 1024 by 8 equals 128; however, the ideal setting for Fetch Array Size is 127, not 128, because the number of rows times the row size must be slightly smaller than the Network Packet size.

**Packet Size (PacketSize):** Typically, it is optimal for the client to use the maximum packet size that the database server allows. This reduces the total number of round trips required to return data to the client, thus improving performance. Therefore, performance can be improved if the PacketSize attribute is set to the maximum packet size of the Sybase IQ server.

**Select Method (SelectMethod):** If your application often executes a SQL statement before processing or closing the previous result set, then it uses multiple active statements per connection. An active statement is defined as a statement where all the result rows or result sets have not been fetched. Using multiple active statements can cause high overhead on the server. The default setting (1) of this option causes the driver to execute statements directly without the use of database cursors and limits the application to one active statement per connection. If your application requires multiple active statements, then set Select Method to 0 (Cursor). Keep in mind that you may see a negative impact in performance. If this option is set to 0, it should be used in conjunction with Fetch Array Size (ArraySize). If this option is set to 1, Fetch Array Size (ArraySize) has no effect.

## Data Types

Table 22-2 shows how the Sybase IQ data types are mapped to the standard ODBC data types. "Unicode Support" on page 800 lists Sybase IQ to Unicode data type mappings.

**Table 22-2. Sybase IQ Data Type Mapping**

Sybase IQ Data Type...	Maps to ODBC Data Type
BIGINT	SQL_BIGINT
BINARY	SQL_BINARY
BIT	SQL_BIT
CHAR	SQL_CHAR
DATE	SQL_TYPE_DATE
DATETIME	SQL_TYPE_TIMESTAMP
DECIMAL	SQL_DECIMAL
DOUBLE	SQL_DOUBLE
IMAGE	SQL_LONGVARBINARY
INT	SQL_INTEGER
LONG BINARY	SQL_LONGVARBINARY
LONG VARCHAR	SQL_LONGVARCHAR
MONEY	SQL_DECIMAL
NUMERIC	SQL_NUMERIC
REAL	SQL_REAL
SMALLDATETIME	SQL_TYPE_TIMESTAMP
SMALLINT	SQL_SMALLINT
SMALLMONEY	SQL_DECIMAL
TEXT	SQL_LONGVARCHAR
TIME	SQL_TYPE_TIME
TIMESTAMP	SQL_TYPE_TIMESTAMP
TINYINT	SQL_TINYINT
UNIQUEIDENTIFIER	SQL_BINARY
UNIQUEIDENTIFIERSTR	SQL_CHAR
UNSIGNED BIGINT	SQL_BIGINT
UNSIGNED INT	SQL_INTEGER
VARBINARY	SQL_VARBINARY
VARCHAR	SQL_VARCHAR

NOTE: The Sybase IQ Wire Protocol driver supports extended new limits (XNL) for character and binary columns—columns with lengths greater than 255.

See ["Retrieving Data Type Information" on page 59](#) for information about retrieving data types.

# Unicode Support

When connected to a Unicode database, the Sybase IQ Wire Protocol driver supports the Unicode data types listed in the following table, in addition to standard ODBC data types listed in [Table 22-2 on page 799](#).

**Table 22-3. Mapping Sybase IQ Data Types to Unicode Data Types**

Sybase IQ Data Type. . .	Maps to Unicode Data Type. . .
CHAR <sup>1</sup>	SQL_WCHAR
LONG VARCHAR	SQL_WLONGVARCHAR
TEXT <sup>1</sup>	SQL_WLONGVARCHAR
UNIQUEIDENTIFIERSTR	SQL_WCHAR
VARCHAR <sup>1</sup>	SQL_WVARCHAR

1. This data type is available only if the data source is configured to use the UTF-8 character set.

For data types that require the UTF-8 character set, set the Charset connection string attribute. See ["Charset" on page 782](#) for information about using this connection string attribute.

The driver supports the Unicode ODBC W (Wide) function calls, such as SQLConnectW. This allows the Driver Manager to transmit these calls directly to the driver. Otherwise, the Driver Manager would incur the additional overhead of converting the W calls to ANSI function calls, and vice versa.

See ["UTF-16 Applications on UNIX and Linux" on page 125](#) for related details. Also, refer to [Chapter 4 "Internationalization, Localization, and Unicode"](#) in the *DataDirect Connect Series for ODBC Reference* for a more detailed explanation of Unicode.



---

## Advanced Features

The driver supports the following advanced features:

- Failover
- Connection Pooling

### Failover

The driver supports failover and its related connection options. Failover connection options are located on the [Failover tab](#) of the driver Setup dialog box. See ["Using Failover" on page 65](#) for a general description of failover and its implementation.

### Connection Pooling

The driver supports connection pooling and its related connection options. Connection pooling connection options are located on the [Pooling tab](#) of the driver Setup dialog box. See ["Using DataDirect Connection Pooling" on page 82](#) for a general description of connection pooling and its implementation.

---

## Unexpected Characters

Users are sometimes surprised when they insert a character into a database, only to have a different character displayed when they fetch it from the database. There are many reasons this can happen, but it most often involves code page issues, not driver errors.

Client and server machines in a database system each use code pages, which can be identified by a name or a number, such as Shift\_JIS (Japanese) or cp1252 (Windows English). A code page is a mapping that associates a sequence of bits, called a code point, with a specific character. Code pages include the characters and symbols of one or more languages. Regardless of geographical location, a machine can be configured to use a specific code page. Most of the time, a client and database server would use similar, if not identical, code pages. For example, a client and server might use two different Japanese code pages, such as Shift\_JIS and EUC\_JP, but they would still share many Japanese characters in common. These characters might, however, be represented by different code points in each code page. This introduces the need to convert between code pages to maintain data integrity. In some cases, no one-to-one character correspondence exists between the two code points. This causes a substitution character to be used, which can result in displaying an unexpected character on a fetch.

When the driver on the client machine opens a connection with the database server, the driver determines the code pages being used on the client and the server. This is determined from the Active Code Page on a Windows-based machine.

If the client and server code pages are compatible, the driver transmits data in the code page of the server. Even though the pages are compatible, a one-to-one correspondence for every character may not exist. If the client and server code pages are completely

dissimilar, for example, Russian and Japanese, then many substitutions occur because very few, if any, of the characters are mapped between the two code pages.

The following is a specific example of an unexpected character:

- The Windows client machine is running code page cp1252.
- The Sybase IQ server is running code page cp850.
- You insert decimal literals for character data. You think you are inserting LATIN SMALL LETTER I WITH ACUTE (i) and BOX DRAWINGS DOUBLE VERTICAL (||) in the database. When you fetch the data, you see INVERTED EXCLAMATION MARK (¡) and MASCULINE ORDINAL INDICATOR (º) displayed on the client instead.

This occurs because the code points do not correspond in the two code pages. An example of syntax you would use to insert the decimal literals is:

```
CREATE table cp850chars(val text)
INSERT INTO cp850chars values(CHAR(161)+CHAR(186))
```

This effectively inserts the hexadecimal bytes for the numbers 161 (0xA1) and 186 (0xBA) into the text column. Each of these hexadecimal bytes is treated as the single byte code point for the character it represents. The problem is that the character representation for these two particular hexadecimal values is different from code page cp850 to code page cp1252. On cp850, these hexadecimal values represent í (0xA1) and || (0xBA), which is what you thought you were inserting by using the previously described syntax. When you fetch these hexadecimal values, however, the characters displayed on your client machine are ¡ (0xA1) and º (0xBA), because that is what the hexadecimal values represent in code page cp1252. This is not a matter of data corruption or substitution; these hexadecimal values simply represent different values in the two different code pages.

This is not a driver error. It occurs because the code points map differently and because some characters do not exist in a code page. The best way to avoid these problems is to use the same code page on both the client and server machines.

---

## NULL Values

When the Sybase IQ Wire Protocol driver establishes a connection, the driver sets the Sybase database option ansinull to on. Setting ansinull to on ensures that the driver is compliant with the ANSI SQL standard, which makes developing cross-database applications easier.

By default, Sybase IQ does not evaluate NULL values in SQL equality (=), inequity (<>), or aggregate function comparisons in an ANSI SQL-compliant manner. For example, the ANSI SQL specification defines that `col1=NULL` always evaluates to false:

```
SELECT * FROM table WHERE col1 = NULL
```

Using the default database setting (ansinull=off), the same comparison evaluates to true instead of false.

Setting `ansinull` to on changes the default database behavior so that SQL statements must use `IS NULL` instead of `=NULL`. For example, using the Sybase IQ Wire Protocol driver, if the value of `col1` in the following statement is NULL, the comparison evaluates to true:

```
SELECT * FROM table WHERE col1 IS NULL
```

In your application, you can restore the default Sybase IQ behavior for a connection in the following ways:

- Use the Initialization String option to specify the SQL command `set ANSINULL off`. For example, the following connection string ensures that the handling of NULL values is restored to the Sybase IQ default for the current connection:

```
DSN=SYB TABLES;DB=PAYROLL;IS=set ANSINULL off
```

- Explicitly execute the following statement after the connection is established:

```
SET ANSINULL OFF
```

---

## Persisting a Result Set as an XML Data File

The driver allows you to persist a result as an XML data file with embedded schema. See ["Persisting a Result Set as an XML Data File" on page 60](#) for details about implementation.

---

## Isolation and Lock Levels Supported

The Sybase IQ database system supports isolation levels 0 (read uncommitted), 1 (read committed, the default), 2 (repeatable read), and 3 (serializable). It supports page-level locking.

Refer to Chapter 7 "Locking and Isolation Levels" in the *DataDirect Connect Series for ODBC Reference* for details.

---

## SQL Support

The driver supports the minimum SQL grammar.

---

## ODBC Conformance Level

The driver is Level 1 compliant, that is, it supports all ODBC Core and Level 1 functions. In addition, the driver supports the following Level 2 functions:

- SQLColumnPrivileges
- SQLForeignKeys
- SQLTablePrivileges

Refer to Chapter 2 “ODBC API and Scalar Functions” in the *DataDirect Connect Series for ODBC Reference* for a list of the API functions supported by the Sybase IQ Wire Protocol driver.

---

## Number of Connections and Statements Supported

The Sybase IQ database system supports multiple connections and multiple statements per connection. If the Select Method option on the Performance tab or the connection string attribute SelectMethod is set to 1 (Direct), Sybase IQ data sources are limited to one active statement in manual commit mode.

---

## Using Arrays of Parameters

When designing an application, using parameter arrays for bulk inserts or updates, for example, can improve performance. Refer to Chapter 5 “Designing ODBC Applications for Performance Optimization” in the *DataDirect Connect Series for ODBC Reference* for more information about using arrays of parameters to improve performance.

Because Sybase IQ databases do not support parameter arrays natively, the Sybase IQ Wire Protocol driver emulates them by sending T-SQL batches of Insert or Update statements to the database, which will improve performance.

## 23 The Driver for Apache Hive™

The DataDirect Connect XE *for* ODBC and DataDirect Connect64 XE *for* ODBC for Apache Hive™ Wire Protocol driver each support the following Apache Hive versions and distributions:

**Table 23-1. Supported Apache Hive Versions and Distribution Versions**

	Distribution Version	Apache Hive Version
Amazon Elastic MapReduce (Amazon EMR)	N/A	<ul style="list-style-type: none"> <li>■ Hive 0.8.x</li> <li>■ Hive 0.9.x (pending)</li> </ul>
Apache Hadoop Hive	N/A	<ul style="list-style-type: none"> <li>■ Hive 0.8.x</li> <li>■ Hive 0.9.x</li> </ul>
Cloudera's Distribution Including Apache Hadoop (CDH)	CDH 4.0.x	■ Hive 0.8.x
	CDH3 update 4	■ Hive 0.7.1
MapR Distribution for Apache Hadoop	MapR 1.2	■ Hive 0.7.1
	MapR 2.0	■ Hive 0.9.x

The driver is supported in the Windows, UNIX, and Linux environments. See ["Environment-Specific Information" on page 45](#) for detailed information about the environments supported by this driver.

Refer to the readme file shipped with your DataDirect Connect driver for the file name of the driver.

### Driver Requirements

The driver has no client requirements.

### Configuring and Connecting to Data Sources

After you install the driver, you configure data sources to connect to the database. See [Chapter 1 "Quick Start Connect" on page 35](#) for an explanation of different types of data sources. The data source contains connection options that allow you to tune the driver for specific performance. If you want to use a data source but need to change some of its values, you can either modify the data source or override its values at connection time through a connection string.

If you choose to use a connection string, you must use specific connection string attributes. See ["Using a Connection String" on page 810](#) and [Table 23-2](#) for an alphabetical list of driver connection string attributes and their initial default values.



## Data Source Configuration in the UNIX odbc.ini File

On UNIX and Linux, you must set up the proper ODBC environment before configuring data sources. See ["Environment Configuration" on page 37](#) for basic setup information and ["Environment Variables" on page 97](#) for more detail about this procedure.

Data sources for UNIX and Linux are stored in the system information file (by default, odbc.ini). You can configure and modify data sources directly by editing the odbc.ini file and storing default connection values there. See ["Configuration Through the System Information \(odbc.ini\) File" on page 103](#) for detailed information about the specific steps necessary to configure a data source.

[Table 23-2](#) lists driver connection string attributes that must be used in the odbc.ini file to set the value of the attributes. Note that only the long name of the attribute can be used in the file. The default listed in the table is the initial default value when the driver is installed.

## Data Source Configuration through a GUI



On Windows, data sources are stored in the Windows Registry. You can configure and modify data sources through the ODBC Administrator using a driver Setup dialog box, as described in this section.



On UNIX and Linux, data sources are stored in the odbc.ini file.

When the driver is first installed, the values of its connection options are set by default. These values appear on the driver Setup dialog box tabs when you create a new data source. You can change these default values by modifying the data source. In the following procedure, the description of each tab is followed by a table that lists the connection options for that tab and their initial default values. This table links you to a complete description of the options and their connection string attribute equivalents. The connection string attributes are used to override the default values of the data source if you want to change these values at connection time.



### To configure a data source for Apache Hive:

- 1 Start the ODBC Administrator:
  - On Windows, start the ODBC Administrator by selecting its icon from the DataDirect Connect program group.
- 2 Select a tab:
  - **User DSN:** If you are configuring an existing user data source, select the data source name and click **Configure** to display the driver Setup dialog box.

If you are configuring a new user data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.

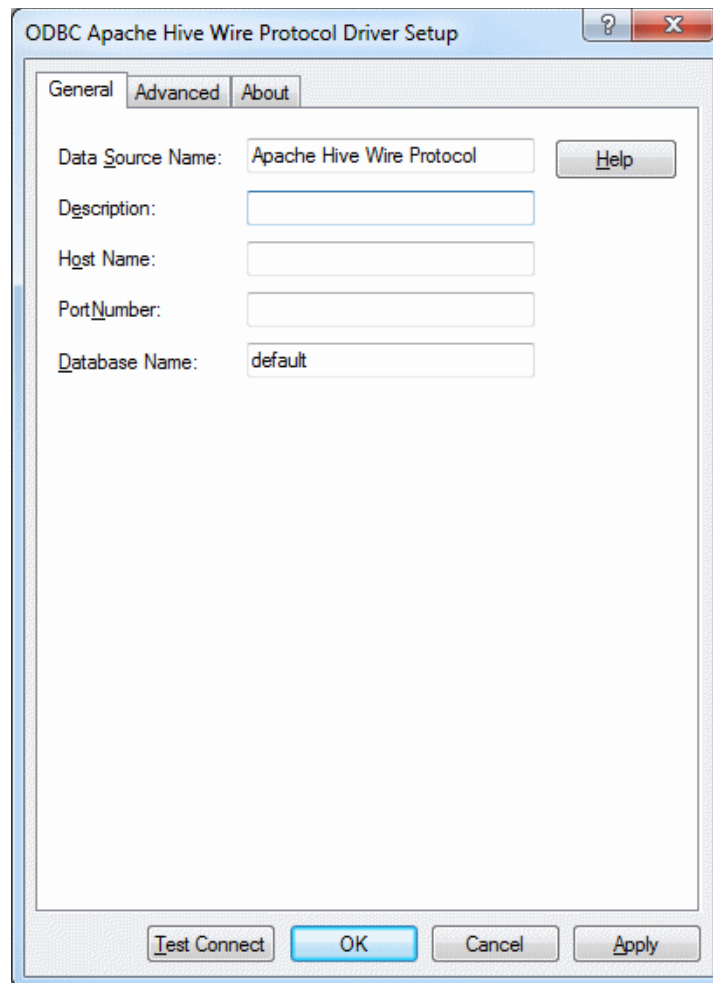
- **System DSN:** If you are configuring an existing system data source, select the data source name and click **Configure** to display the driver Setup dialog box.

If you are configuring a new system data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.

- **File DSN:** If you are configuring an existing file data source, select the data source file and click **Configure** to display the driver Setup dialog box.

If you are configuring a new file data source, click **Add** to display a list of installed drivers; then, select a driver. Click **Advanced** if you want to specify attributes; otherwise, click **Next** to proceed. Specify a name for the data source and click **Next**. Verify the data source information; then, click **Finish** to display the driver Setup dialog box.

The General tab of the Setup dialog box appears by default.

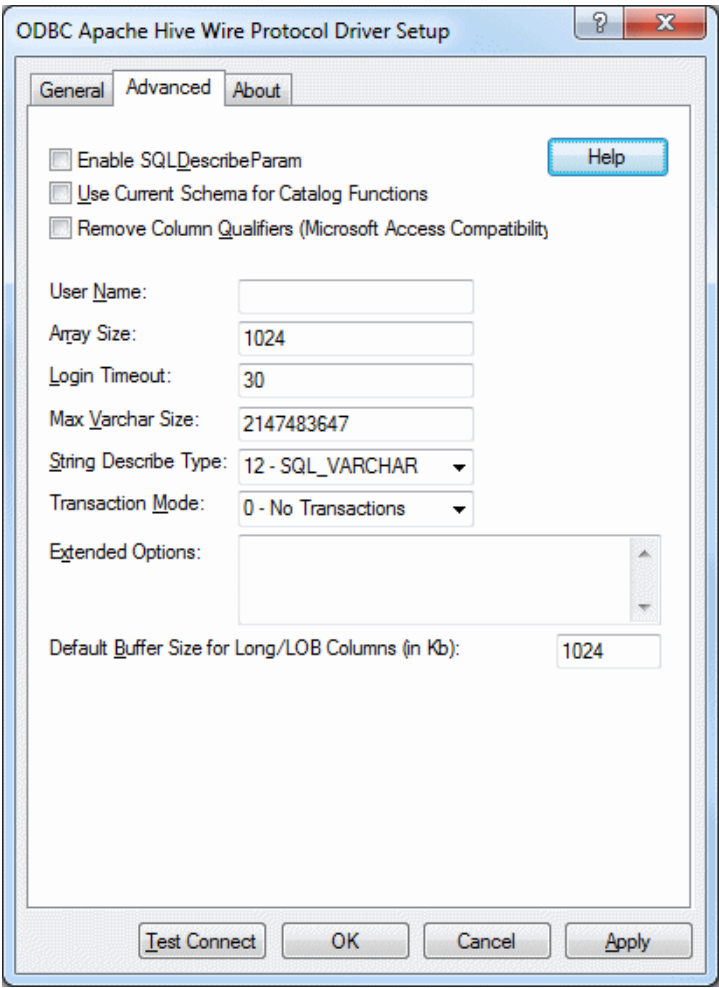


NOTE: The General tab displays only fields that are required for creating a data source. The fields on all other tabs are optional, unless noted otherwise.

- 3 On this tab, provide values for the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: General	Default
<a href="#">Data Source Name (see page 812)</a>	None
<a href="#">Description (see page 813)</a>	None
<a href="#">Host Name (see page 814)</a>	None
<a href="#">Port Number (see page 815)</a>	None
<a href="#">Database (see page 813)</a>	default

- 4 Optionally, click the **Advanced** tab to specify additional data source settings.



On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Advanced	Default
<a href="#">Enable SQLDescribeParam (see page 814)</a>	Disabled
<a href="#">Use Current Schema for Catalog Functions (see page 816)</a>	Disabled
<a href="#">Remove Column Qualifiers (see page 815)</a>	Disabled



**Connection Options: Advanced****Default**

User Name (see page 817)	None
Array Size (see page 812)	1024
Login Timeout (see page 814)	30
Max Varchar Size (see page 815)	2147483647
String Describe Type (see page 816)	12 - SQ_VARCHAR
Transaction Mode (see page 816)	0 - No Transactions
Default Buffer Size for Long/LOB Columns (in Kb) (see page 813)	1024



**Extended Options:** Type a semi-colon separated list of connection options and their values. Use this configuration option to set the value of undocumented connection options that are provided by Progress DataDirect customer support. You can include any valid connection option in the Extended Options string, for example:

```
Database=Server1;UndocumentedOption1=value[;UndocumentedOption2=value;]
```

If the Extended Options string contains option values that are also set in the setup dialog or data source, the values of the options specified in the Extended Options string take precedence. However, connection options that are specified on a connection string override any option value specified in the Extended Options string.

**NOTE:** Do not specify the Extended Options configuration option in a connection string, or the driver will return an error. Instead, applications should specify the individual undocumented connection options in the connection string.

- 5 At any point during the configuration process, you can click **Test Connect** to attempt to connect to the data source using the connection options specified in the driver Setup dialog box. A logon dialog box appears (see ["Using a Logon Dialog Box" on page 811](#) for details). 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 incorrect environment or connection value, it displays an appropriate error message. Click **OK**.

**NOTE:** If you are configuring alternate servers for use with the connection failover feature, be aware that the Test Connect button tests only the primary server, not the alternate servers.

- 6 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.

## Using a Connection String

If you want to use a connection string for connecting to a database, or if your application requires it, you must specify either a DSN (data source name), a File DSN, or a DSN-less connection in the string. The difference is whether you use the DSN=, FILEDSN=, or the DRIVER= keyword in the connection string, as described in the ODBC specification. A DSN or FILEDSN connection string tells the driver where to find the default connection information. Optionally, you may specify *attribute=value* pairs in the connection string to override the default values stored in the data source.

The DSN connection string has the form:

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

The FILEDSN connection string has the form:

```
FILEDSN=filename.dsn[;attribute=value[;attribute=value]...]
```

The DSN-less connection string specifies a driver instead of a data source. All connection information must be entered in the connection string because the information is not stored in a data source.

The DSN-less connection string has the form:

```
DRIVER=[{ } driver_name[ } ] [;attribute=value[;attribute=value]...]
```

[Table 23-2](#) lists the long and short names for each attribute, as well as the initial default value when the driver is first installed. You can specify either long or short names in the connection string.

An example of a DSN connection string with overriding attribute values for Hive is:

```
DSN=Accounting;UID=JOHN;PWD=XYZZY
```

A FILEDSN connection string is similar except for the initial keyword:

```
FILEDSN=Hive.dsn;UID=JOHN;PWD=XYZZY
```

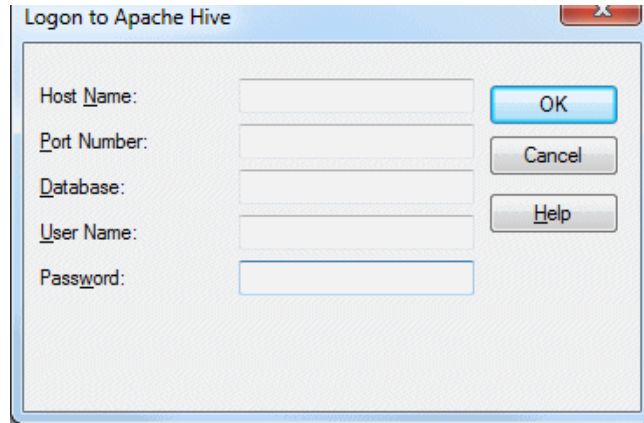
A DSN-less connection string must provide all necessary connection information:

```
DRIVER=DataDirect 7.1 Hive;HOST=server1;PORT=10000;UID=JOHN;PWD=XYZZY;
```

## 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.

NOTE: A user name and password are not required to connect to Hive.



In this dialog box, provide the following information:

NOTE: To configure a standard connection, complete the first two fields and skip to Step 4.

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

The IP address can be specified in either IPv4 or IPv6 format, or a combination of the two. See ["Using IP Addresses" on page 54](#) for details concerning these formats.

- 2 In the Port Number field, type the port number that your Hive server is listening on. Check with your Hive administrator for the correct number.
- 3 Click **OK** to log on to the Greenplum database installed on the server you specified and to update the values in the Registry.

NOTE: The User Name and Password fields are not used at this time to connect to the Apache Hive server.

---

## Connection Option Descriptions

The following connection option descriptions are listed alphabetically by the GUI name that appears on the driver Setup dialog box. The connection string attribute name, along with its short name, is listed immediately underneath the GUI name. For example:

### Array Size

Attribute    ArraySize (AS)

In most cases, the GUI name and the attribute name are the same; however, some exceptions exist. If you need to look up an option by its connection string attribute name, please refer to the alphabetical table of connection string attribute names.

Table 23-2 lists the connection string attributes supported by the driver.

Table 23-2. Attribute Names for the Driver for Apache Hive	
Attribute (Short Name)	Default
ArraySize (AS)	1024
Database (DB)	default
DataSourceName (DSN)	None
Description (n/a)	None
DefaultLongDataBuffLen (DLDBL)	1024
EnableDescribeParam (EDP)	0 (Disabled)
HostName (HOST)	None
LoginTimeout (LT)	30
LogonID (UID)	None
MaxVarcharSize (MVS)	2147483647
PortNumber (PORT)	None
RemoveColumnQualifiers (RCQ)	0 (Disabled)
StringDescribeType (SDT)	12 - SQL_VARCHAR
TransactionMode (TM)	0 (No Transactions)
UseCurrentSchema (UCS)	0 (Disabled)

Array Size

Attribute	ArraySize (AS)
Description	<p>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 connection option can affect performance.</p> <p>This connection option can affect performance. See "Performance Considerations" on page 817 for details.</p>
Valid Values	<p><i>x</i></p> <p>where <i>x</i> is a positive integer specifying the number of rows.</p>
Default	1024
GUI Tab	Advanced tab on page 808

Data Source Name

Attribute	DataSourceName (DSN)
Description	The name of a data source in your Windows Registry or odbc.ini file.
Valid Values	<p><i>string</i></p> <p>where <i>string</i> is the name of a data source.</p>

Default	None
GUI Tab	<a href="#">General tab</a> on page 808

## Database

Attribute	Database (DB)
Description	Specifies the name of the Hive database. The database must exist, or the connection attempt will fail.
Valid Values	<i>database_name</i>  where:  <i>database_name</i> is the name of the Hive database.
Default	default
GUI Tab	<a href="#">General tab</a> on page 806

## Default Buffer Size for Long/LOB Columns (in Kb)

Attribute	DefaultLongDataBuffLen (DLDBL)
Description	The maximum length of data (in KB) the driver can fetch from long columns in a single round trip and the maximum length of data that the driver can send using the SQL_DATA_AT_EXEC parameter.  This connection option can affect performance. See <a href="#">"Performance Considerations" on page 817</a> for details.
Valid Values	An integer in multiples of 1024  The value must be in multiples of 1024 (for example, 1024, 2048). You need to increase the default value if the total size of any Long data exceeds 1 MB. This value is multiplied by 1024 to determine the total maximum length of fetched data. For example, if you enter a value of 2048, the maximum length of data would be 1024 x 2048, or 2097152 (2 MB).
Default	1024
GUI tab	<a href="#">Advanced tab</a> on page 808

## Description

Attribute	Description (n/a)
Description	An optional long description of a data source. This description is not used as a runtime connection attribute, but does appear in the ODBC.INI section of the Registry and in the odbcc.ini file.
Valid Values	<i>string</i>  where <i>string</i> is a description of a data source.
Default	None
GUI Tab	<a href="#">General tab</a> on page 808

Enable SQLDescribeParam

Attribute	EnableDescribeParam (EDP)
Description	Determines whether the driver uses the SQLDescribeParam function, which describes parameters as a data type of SQL_VARCHAR with a length of 255 for statements.
Valid Values	0   1  If set to 1 (enabled), the SQLDescribeParam function describes parameters as a data type of SQL_VARCHAR with a length of 255 for statements.  If set to 0 (disabled), the SQLDescribeParam function returns the standard ODBC error IM001.
Default	0 (Disabled)
GUI tab	<a href="#">Advanced tab</a> on page 808

Host Name

Attribute	HostName (HOST)
Description	The name or the IP address of the server to which you want to connect.
Valid Values	<i>host_name</i>   <i>IP_address</i>  where:  <i>hostname</i> is the name of the Apache Hive server to which you want to connect  <i>IP_address</i> is the IP address of the server to which you want to connect.
Default	None
GUI Tab	<a href="#">General tab</a> on page 808

Login Timeout

Attribute	LoginTimeout (LT)
Description	The number of seconds the driver waits for a connection to be established before returning control to the application and generating a timeout error. To override the value that is set by this connection option for an individual connection, set a different value in the SQL_ATTR_LOGIN_TIMEOUT connection attribute using the SQLSetConnectAttr() function.
Valid Values	-1   0   <i>x</i>  where <i>x</i> is a positive integer that specifies a number of seconds.  If set to -1, the connection request does not time out. The driver silently ignores the SQL_ATTR_LOGIN_TIMEOUT attribute.  If set to 0, the connection request does not time out, but the driver responds to the SQL_ATTR_LOGIN_TIMEOUT attribute.  If set to <i>x</i> , the connection request times out after the specified number of seconds unless the application overrides this setting with the SQL_ATTR_LOGIN_TIMEOUT attribute.

Default 30  
 GUI Tab [Advanced tab](#) on page 808

## Max Varchar Size

Attribute MaxVarcharSize (MVS)  
 Description Specifies the maximum size of columns of type SQL\_VARCHAR that the driver describes through result set descriptions and catalog functions.  
 Valid Values A positive integer from 255 to *x*  
 where *x* is maximum size of the SQL\_VARCHAR data type.  
 Default 2147483647  
 GUI Tab [Advanced tab](#) on page 808

## Port Number

Attribute PortNumber (PORT)  
 Description The port number of the server listener.  
 Valid Values *port\_number*  
 where the *port\_number* is the port number of the server listener. Check with your database administrator for the correct number.  
 NOTE: The default port number for the Apache Hive server is 10000. Because of reported concurrency issues, you might want to use a different port number.  
 Default None  
 GUI Tab [General tab](#) on page 808

## Remove Column Qualifiers

Attribute RemoveColumnQualifiers (RCQ)  
 Description Specifies whether the driver removes 3-part column qualifiers and replaces them with alias.column qualifiers. Access execute a Select statement using this syntax when an index is specified on a linked table.  
 NOTE: When using the driver with Microsoft Access in creating a linked table, it is highly recommended that you do not specify an index. Specifying an index causes Access to execute a Select statement for each row, which results in very slow performance.  
 Valid Values 0 | 1  
 If set to 1 (enabled) the driver removes 3-part column qualifiers and replaces them with alias.column qualifiers. Microsoft Access Compatibility  
 If set to 0, the driver does not modify the request.  
 Default 0 (Disabled)  
 GUI Tab [Advanced tab](#) on page 808

## String Describe Type

Attribute	StringDescribeType (SDT)
Description	Specifies whether all string columns are described as SQL_VARCHAR. This connection option affects SQL_Columns, SQLDescribeCol, SQLColAttributes, etc. It does not affect SQLGetTypeInfo.
Valid Values	-1   12  If set to -1 (SQL_LONGVARCHAR), all string columns are described as SQL_LONGVARCHAR.  If set to 12 - SQL_VARCHAR, all string columns are described as SQL_VARCHAR.
Default	12 - SQL_VARCHAR
GUI Tab	<a href="#">Advanced tab</a> on page 808

## Transaction Mode

Attribute	TransactionMode (TM)
Description	Specifies how the driver handles manual transactions.
Valid Values	0   1  If set to 1 - Ignore, the data source does not support transactions and the driver always operates in auto-commit mode. Calls to set the driver to manual commit mode and to commit transactions are ignored. Calls to rollback a transaction cause the driver to return an error indicating that no transaction is started. Metadata indicates that the driver supports transactions and the ReadUncommitted transaction isolation level.  If set to 0 (No Transactions), the data source and the driver do not support transactions. Metadata indicates that the driver does not support transactions.
Default	0 (No Transactions)
GUI Tab	<a href="#">Advanced tab</a> on page 808

## Use Current Schema for Catalog Functions

Attribute	UseCurrentSchema (UCS)
Description	Specifies whether results are restricted to the tables and views in the current schema if a catalog function call is made without specifying a schema or if the schema is specified as the wildcard character %. Restricting results to the tables and views in the current schema improves performance of catalog calls that do not specify a schema.
Valid Values	0   1  If set to 1 (Enabled), results of catalog function calls are restricted to the tables and views in the current schema.  If set to 0 (Disabled), results of catalog function calls are not restricted.
Default	0 (Disabled)
GUI Tab	<a href="#">Advanced tab</a> on page 808



## User Name

Attribute	LogonID (UID)
Description	The default user ID that is used to connect to your database.
	NOTE: Not used to log on to Apache Hive at this time.
Valid Values	N/A
GUI Tab	<a href="#">Advanced tab</a> on page 808

---

## Performance Considerations

The following connection options can enhance driver performance. You can also enhance performance through efficient application design. Refer to [Chapter 5 “Designing ODBC Applications for Performance Optimization”](#) in the *DataDirect Connect Series for ODBC Reference* for details.

**Array Size (ArraySize):** If this connection string attribute is set appropriately, the driver can improve performance of your application by reducing the number of round trips on the network. For example, if your application normally retrieves 200 rows, it is more efficient for the driver to retrieve 200 rows at one time over the network than to retrieve 50 rows at a time during four round trips over the network.

---

## Data Types

[Table 23-3](#) shows how the Apache Hive data types are mapped to the standard ODBC data types.

---

**Table 23-3. Apache Hive Data Types**

---

Apache Hive	ODBC
Bigint	SQL_BIGINT
Binary <sup>1</sup>	SQL_VARBINARY
Boolean	SQL_BIT
Double	SQL_DOUBLE
Float	SQL_REAL
Int	SQL_INTEGER
Smallint	SQL_SMALLINT
String	SQL_VARCHAR <sup>2</sup>

**Table 23-3. Apache Hive Data Types** (cont.)

Apache Hive	ODBC
Timestamp <sup>3</sup>	SQL_TYPE_TIMESTAMP
Tinyint	SQL_TINYINT

- 1. Supported only with Hive 0.8.1 and higher.
- 2. Maximum of 2 GB
- 3. Supported only with Hive 0.8.0 and higher.

## Materialized Views

Apache Hive supports views but purely as logical objects with no associated storage. As such, there is no support for materialized views in Hive; therefore, the driver does not support materialized views.

## Stored Procedures

Apache Hive has no concept of stored procedures. Therefore, they are not supported in the driver.

## Unicode Support

The driver is fully Unicode enabled. On UNIX and Linux platforms, the driver supports both UTF-8 and UTF-16. On Windows platforms, the Hive driver supports UCS-2/UTF-16 only.

The driver supports the Unicode ODBC W (Wide) function calls, such as SQLConnectW. This allows the Driver Manager to transmit these calls directly to the driver. Otherwise, the Driver Manager would incur the additional overhead of converting the W calls to ANSI function calls, and vice versa.

See ["UTF-16 Applications on UNIX and Linux" on page 125](#) for related details. Also, refer to Chapter 4 "Internationalization, Localization, and Unicode" in the *DataDirect Connect Series for ODBC Reference* for a more detailed explanation of Unicode.

---

## Isolation and Lock Levels Supported

Apache Hive supports isolation level 1 (read committed) and isolation level 3 (serializable). Hive supports table and partition locks.

Refer to [Chapter 7 “Locking and Isolation Levels”](#) in the *DataDirect Connect Series for ODBC Reference* for details.

---

## SQL Support

The driver supports the core SQL grammar.

Refer to the [Hive Language Manual](#) for information about using HiveQL.

Also, refer to [Chapter 11 “SQL Functionality for the Driver for Apache Hive™”](#) in the *DataDirect Connect Series for ODBC Reference* for a more detailed information.

---

## ODBC Conformance Level

The driver supports ODBC API Conformance Level 1.

NOTES:

- SQLCancel and SQLTransact execute successfully but perform no functions.
- SQLStatistics always returns an empty result set.

---

## Using Arrays of Parameters

The driver supports parameter arrays, processing the arrays as a series of executions, one execution for each row in the array. Refer to [Chapter 5 “Designing ODBC Applications for Performance Optimization”](#) in the *DataDirect Connect Series for ODBC Reference* for more information about using arrays of parameters to improve performance.

---

## Limitations on Apache Hive™ Functionality

The following restrictions are based on using Apache Hive 0.9:

- No support for user-level authentication
- No support for canceling a running query
- No support for row-level inserts, updates, or deletes
- No support for multiple simultaneous connections per port
- No difference between "NULL" and null values

For a more complete listing of Apache Hive known issues and limitations for your version of Hive, refer to the Apache Hive user documentation:

<https://cwiki.apache.org/confluence/display/Hive/Home>

Note that Apache Hive is not designed for OLTP workloads and does not offer real-time queries or row-level updates. Instead, Hive is designed for batch type jobs over large data sets with high latency. This means that queries such as "SELECT \* FROM mytable" return quickly. However, other SELECT statements are much slower.

---

## Best Practices

Consider the following suggestions when setting up your Apache Hive environment:

- Create an Apache Hive database for each user, with a database name that corresponds to the user name, using the Database connection option. For example, user odbc01 uses database odbc01, user odbc02 uses database odbc02, and so on.
- Because Apache Hive servers do not currently handle multiple connections well, consider using one Apache Hive server for each connection.

## 24 The Driver for the Teradata Database

The DataDirect Connect XE *for* ODBC and DataDirect Connect64 XE *for* ODBC driver for the Teradata database each support the following Teradata database servers when using the appropriate client software:

- Teradata 14.0
- Teradata 13.0, 13.1
- Teradata 12.0
- V2R6.0, V2R6.1, V2R6.2

The driver is supported in the Windows, UNIX, and Linux environments. See ["Environment-Specific Information" on page 45](#) for detailed information about the environments supported by this driver.

Refer to the readme file shipped with your DataDirect Connect product for the file name of the driver.

---

### Driver Requirements

The driver requires Teradata Tools and Utilities (TTU) 8.2 or higher, which includes CLIV2, TGSS, and ICU client software, on all platforms. It requires TTU 12.0 to support 12.0 functionality.

NOTE: TTU 12.0 is not available for the Itanium II platform. You can use TTU 8.2 on an Itanium II client to connect to a Teradata 12.0 database, but functionality is limited to that of TTU 8.2.

---

### Configuring and Connecting to Data Sources

After you install the driver, you configure data sources to connect to the database. See [Chapter 1 "Quick Start Connect" on page 35](#) for an explanation of different types of data sources. The data source contains connection options that allow you to tune the driver for specific performance. If you want to use a data source but need to change some of its values, you can either modify the data source or override its values at connection time through a connection string.

If you choose to use a connection string, you must use specific connection string attributes. See ["Using a Connection String" on page 827](#) and [Table 24-1 on page 830](#) for an alphabetical list of driver connection string attributes and their initial default values.

## Data Source Configuration in the UNIX/Linux odbc.ini File

On UNIX and Linux, you must set up the proper ODBC environment before configuring data sources. See ["Environment Configuration" on page 37](#) for basic setup information and ["Environment Variables" on page 97](#) for more detail about this procedure.

Data sources for UNIX and Linux are stored in the system information file (by default, odbc.ini). If you have a Motif GUI environment on Linux, you can configure and modify data sources through the DataDirect ODBC Data Source Administrator for Linux (the Linux ODBC Administrator) using a driver Setup dialog box. (See ["Configuration Through the Administrator" on page 100](#) for a detailed explanation of the Administrator.)

If you do not have a GUI environment, you can configure and modify data sources directly by editing the odbc.ini file and storing default connection values there. See ["Configuration Through the System Information \(odbc.ini\) File" on page 103](#) for detailed information about the specific steps necessary to configure a data source.

[Table 24-1 on page 830](#) lists driver connection string attributes that must be used in the odbc.ini file to set the value of the attributes. Note that only the long name of the attribute can be used in the file. The default listed in the table is the initial default value when the driver is installed.

## Data Source Configuration through a GUI

On Windows, data sources are stored in the Windows Registry. You can configure and modify data sources through the ODBC Administrator using a driver Setup dialog box, as described in this section.

On UNIX and Linux, data sources are stored in the odbc.ini file. On Linux, you can configure and modify data sources through the Linux ODBC Administrator using a driver Setup dialog box, as described in this section.

NOTE: This book shows dialog box images that are specific to Windows. If you are using the drivers in the Linux environment, the dialog box that you see may differ slightly from the Windows version. Windows-only and UNIX-only connection options are specifically noted by icons in the Setup dialog box descriptions.

When the driver is first installed, the values of its connection options are set by default. These values appear on the driver Setup dialog box tabs when you create a new data source. You can change these default values by modifying the data source. In the following procedure, the description of each tab is followed by a table that lists the connection options for that tab and their initial default values. This table links you to a complete description of the options and their connection string attribute equivalents. The connection string attributes are used to override the default values of the data source if you want to change these values at connection time.

**To configure a Teradata data source:****1** Start the ODBC Administrator:

- On Windows, start the ODBC Administrator by selecting its icon from the DataDirect Connect program group.
- On Linux, change to the *install\_dir/tools* directory and, at a command prompt, enter:

```
odbcadmin
```

where *install\_dir* is the path to the product installation directory.

**2** Select a tab:

- **User DSN:** If you are configuring an existing user data source, select the data source name and click **Configure** to display the driver Setup dialog box.

If you are configuring a new user data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.

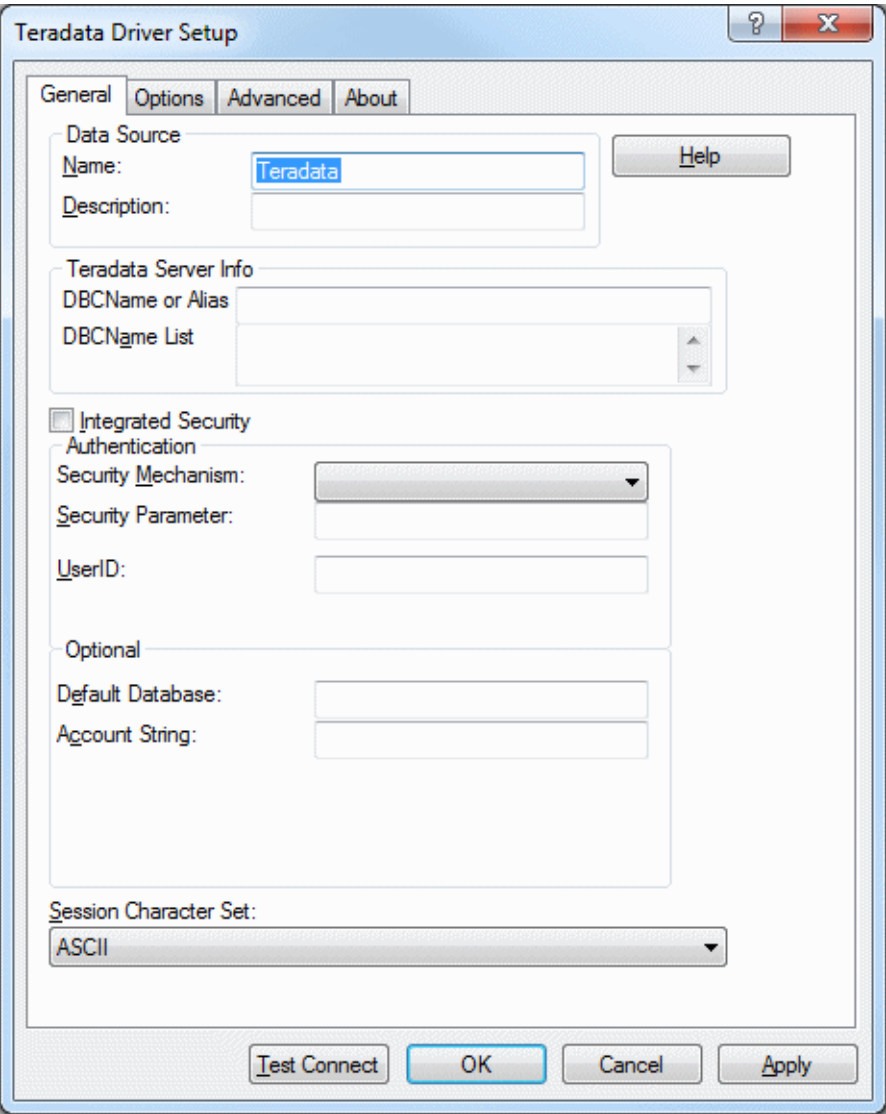
- **System DSN:** If you are configuring an existing system data source, select the data source name and click **Configure** to display the driver Setup dialog box.

If you are configuring a new system data source, click **Add** to display a list of installed drivers. Select the driver and click **Finish** to display the driver Setup dialog box.

- **File DSN:** If you are configuring an existing file data source, select the data source file and click **Configure** to display the driver Setup dialog box.

If you are configuring a new file data source, click **Add** to display a list of installed drivers; then, select a driver. Click **Advanced** if you want to specify attributes; otherwise, click **Next** to proceed. Specify a name for the data source and click **Next**. Verify the data source information; then, click **Finish** to display the driver Setup dialog box.

The General tab of the Setup dialog box appears by default.



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.

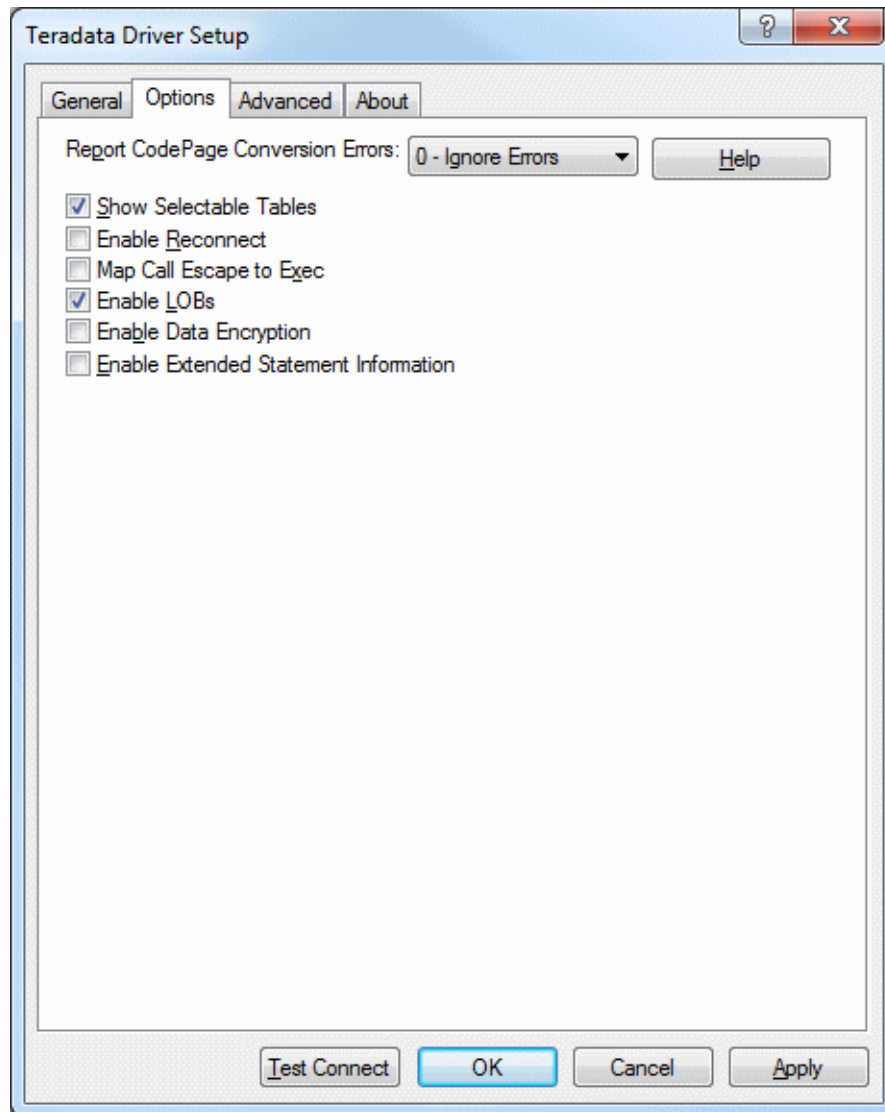
- 3 On this tab, provide values for the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: General	Default
<a href="#">Name (see page 837)</a>	None
<a href="#">Description (see page 833)</a>	None
<a href="#">DBCName or Alias (see page 832)</a>	None
<a href="#">DBCName List (see page 831)</a>	None
<a href="#">Integrated Security (see page 835)</a>	Disabled
<a href="#">Security Mechanism (see page 839)</a>	None
<a href="#">Security Parameter (see page 840)</a>	None
<a href="#">UserID (see page 841)</a>	None



Connection Options: General	Default
<a href="#">Default Database (see page 832)</a>	None
<a href="#">Account String (see page 831)</a>	None
<a href="#">Session Character Set (see page 840)</a>	ASCII

- 4 Click the **Options** tab to specify additional configuration options.

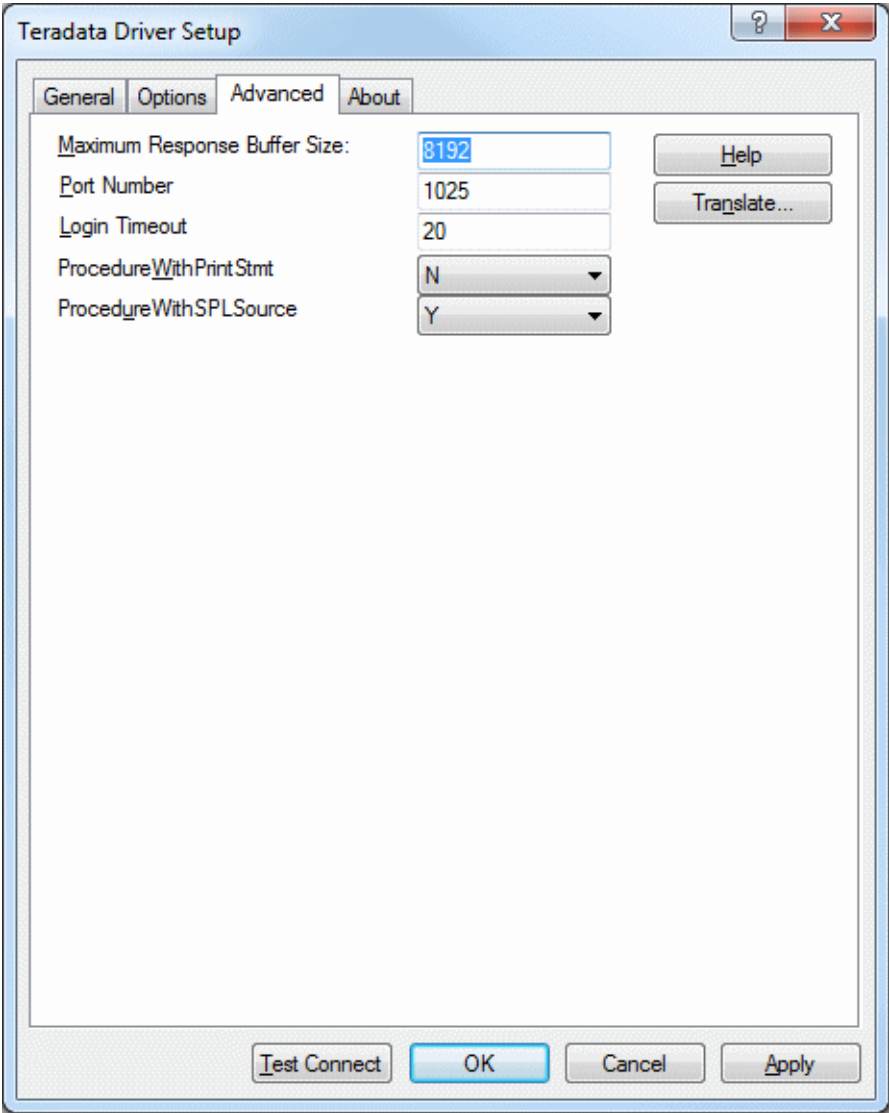


On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Options	Default
<a href="#">Report Codepage Conversion Errors (see page 838)</a>	0 - Ignore Errors
<a href="#">Show Selectable Tables (see page 841)</a>	Enabled
<a href="#">Enable Reconnect (see page 834)</a>	Disabled
<a href="#">Map Call Escape to Exec (see page 836)</a>	Disabled
<a href="#">Enable LOBs (see page 834)</a>	Enabled

Connection Options: Options	Default
<a href="#">Enable Data Encryption (see page 833)</a>	Disabled
<a href="#">Enable Extended Statement Information (see page 834)</a>	Disabled

- 5 Optionally, click the **Advanced** tab to specify additional data source settings.



On this tab, provide values for any of the options in the following table; then, click **Apply**. The table provides links to descriptions of the connection options.

Connection Options: Advanced	Default
<a href="#">Maximum Response Buffer Size (see page 836)</a>	8192
<a href="#">Port Number (see page 837)</a>	1025
<a href="#">Login Timeout (see page 836)</a>	20
<a href="#">ProcedureWithPrintStmt (see page 837)</a>	N (No)

Connection Options: Advanced (cont.)	Default
<a href="#">ProcedureWithSPLSource</a> (see page 838)	Y (Yes)
<a href="#">IANAAppCodePage</a> (see page 835)	4 (ISO 8559-1 Latin-1)
UNIX ONLY	

**Translate:** Click **Translate** to display the Select Translator dialog box, which lists the translators specified in the ODBC Translators section of the Registry. Progress 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.

- 6 At any point during the configuration process, you can click **Test Connect** to attempt to connect to the data source using the connection options specified in the driver Setup dialog box. A logon dialog box appears (see ["Using a Logon Dialog Box" on page 828](#) for details). 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 incorrect environment or connection value, it displays an appropriate error message.

Verify that all required client software is properly installed. If it is not, you will see the message:

Specified driver could not be loaded due to system error [xxx].

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.

## Using a Connection String

If you want to use a connection string for connecting to a database, or if your application requires it, you must specify either a DSN (data source name), a File DSN, or a DSN-less connection in the string. The difference is whether you use the `DSN=`, `FILEDSN=`, or the `DRIVER=` keyword in the connection string, as described in the ODBC specification. A DSN or FILEDSN connection string tells the driver where to find the default connection information. Optionally, you may specify *attribute=value* pairs in the connection string to override the default values stored in the data source.

The DSN connection string has the form:

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

The FILEDSN connection string has the form:

```
FILEDSN=filename.dsn[;attribute=value[;attribute=value]...]
```

The DSN-less connection string specifies a driver instead of a data source. All connection information must be entered in the connection string because the information is not stored in a data source.

The DSN-less connection string has the form:

```
DRIVER=[{driver_name}] [;attribute=value[;attribute=value]...]
```

Table 24-1 lists the long and short names for each attribute, as well as the initial default value when the driver is first installed. You can specify either long or short names in the connection string.

An example of a DSN connection string with overriding attribute values for Teradata is:

```
DSN=Teradata Tables;AS=User2;EnableDataEncryption=Yes
```

A FILEDSN connection string is similar except for the initial keyword:

```
FILEDSN=Teradata.dsn;AS=User2;EnableDataEncryption=Yes
```

A DSN-less connection string must provide all necessary connection information:

```
DRIVER=DataDirect 7.1 Teradata;DBCSN=123.456.78.90;UIS=YES
```

## 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.

Teradata RDBMS Connect...

DBC Name or Alias: [dropdown menu]

Authentication:

☐ Integrated Security

Security Mechanism: [dropdown menu]

Security Parameter: [text field]

UserID: [text field]

Password: [text field]

Optional:

Default Database: [text field]

Account String: [text field]

Buttons: OK, Cancel, Help

In this dialog box, provide the following information:

- 1 Select an alias name or IP address of a Teradata server from the DBC Name or Alias drop-down list. The choices for this list are determined by the entries in DBC Name or Alias and DBCName List on the General tab of the driver Setup dialog box.
- 2 Select the Integrated Security check box to enable the user to connect to the database through Single Sign On (SSO) using one of the authentication mechanisms that support SSO. In this case, User Name, Password, and Domain are not required and are not available fields.
- 3 If you do not use Integrated Security, select a value from the Security Mechanism drop-down list to specify the authentication mechanism used for connections to the data source.
- 4 Type a string of characters in the Security Parameter field that is to be regarded as a parameter to the authentication mechanism. The string is ignored by the ODBC driver and is passed on to the TeraSSO function that is called to set the authentication mechanism.

The characters [] {} () , ; ? \* = ! @ must be enclosed in curly braces.

- 5 Other options that are displayed on the Logon Dialog box depend on the authentication mechanism selected. See the descriptions of these options under [Security Mechanism](#).
- 6 Type the domain name for Third Party Sign On along with the username and password. If a domain name is not provided, then the local domain is assumed.
- 7 Type a default Teradata database (optional).
- 8 Type an account string to be used during the creation of a user account in the Teradata Database instead of providing account information during configuration of ODBC (optional).
- 9 Click **OK** to complete the logon and to update these values in the Registry.

---

## Connection Option Descriptions

The following connection option descriptions are listed alphabetically by the GUI name that appears on the driver Setup dialog box. The connection string attribute name, along with its short name, is listed immediately underneath the GUI name. For example:

### Application Using Threads

Attribute    ApplicationUsingThreads (AUT)

In most cases, the GUI name and the attribute name are the same; however, some exceptions exist. If you need to look up an option by its connection string attribute name, please refer to the alphabetical table of connection string attribute names.

Also, a few connection string attributes, for example, Password, do not have equivalent options that appear on the GUI. They are in the list of descriptions alphabetically by their attribute names.

Table 24-1 lists the connection string attributes supported by the driver.

**Table 24-1. Teradata Attribute Names**

Attribute (Short Name)	Default
AccountString (AS)	None
AuthenticationDomain (AD)	None
AuthenticationPassword (AP)	None
AuthenticationUserId (AUI)	None
CharacterSet (CS)	ASCII
Database (DB)	None
DataSourceName (DSN)	None
DBCName (DBCN)	None
DBCName List	None
Description (n/a)	None
EnableDataEncryption (EDE)	No (Disabled)
EnableExtendedStmtInfo (EESI)	No (Disabled)
EnableLOBs (EL)	Yes (Enabled)
EnableReconnect (ER)	No (Disabled)
IANAAppCodePage (IACP)	4 (ISO 8559-1 Latin-1)
UNIX ONLY	
IntegratedSecurity (IS)	No (Disabled)
LoginTimeout (LTO)	20
MapCallEscapeToExec (MCETE)	No (Disabled)
MaxRespSize (MRS)	8192
Password (PWD)	None
PortNumber (PORT)	1025
PrintOption (PO)	N (No)
ProcedureWithSPLSource (PWSS)	Y (Yes)
ReportCodepageConversionErrors (RCCE)	0 (Ignore Errors)
SecurityMechanism (SECM)	None
SecurityParameter (SP)	None
ShowSelectableTables (SST)	Yes (Enabled)
TDProfile (TDP)	None
TDRole (TDR)	None
TDUserName (TDUN)	None
UserID (UID)	None

## Account String

Attribute	AccountString (AS)
Description	An account string. For a complete description of account strings, refer to the <i>Teradata Database Administration Guide</i> .
Valid Values	<i>string</i>  where <i>string</i> is an account string.
Default	None
GUI Tab	<a href="#">General tab</a> on page 824

## Authentication Password

Attribute	AuthenticationPassword (AP)
Description	The password for the Kerberos, LDAP, NTLM, and TD authentication mechanisms. The Authentication Password option cannot be specified through the driver Setup dialog box and should not be stored in a data source. It is specified through the Logon dialog box or a connection string.
Valid Values	<i>pwd</i>  where <i>pwd</i> is a valid password.
Default	None
GUI Tab	n/a

## Authentication UserID

Attribute	AuthenticationUserId (AUI)
Description	The user ID for the Kerberos, LDAP, NTLM, and TD authentication mechanisms.
Valid Values	<i>userid</i>  where <i>userid</i> is a valid user ID.
Default	None
GUI Tab	<a href="#">General tab</a> on page 824

## DBCName List

Attribute	n/a
Description	A list of IP addresses or aliases to appear in the drop-down list of the Logon dialog box (see <a href="#">"Using a Logon Dialog Box" on page 828</a> for a description). The DBCName List option is not used as a runtime connection attribute.
Valid Values	<i>ip_address</i>   <i>alias</i> [, <i>ip_address</i>   <i>alias</i> ][...]  where  <i>ip_address</i> is an IP address to appear in the drop-down list of the Logon dialog box.

*alias* is an alias to appear in the drop-down list of the Logon dialog box.

Separate multiple IP addresses or aliases with commas. The same restrictions apply as described for the DBCName or Alias option.

Default     None

GUI Tab    [General tab](#) on page 824

**DBCName or Alias**

Attribute   DBCName (DBCN)

Description   The IP address or alias of the Teradata server.

Valid Values   *IP\_address* | *alias*

where

*IP\_address* is the IP address of the Teradata server.

*alias* is the alias of the Teradata server.

If set to *IP\_address*, the time the driver waits for connections to be established is faster. The disadvantage is that if the server designated by that IP address is unavailable, the connection fails and the driver does not attempt to fail over to another IP address.

If set to *alias*, the time the driver waits for connections to be established is slower because the driver must search a local hosts file to resolve the alias to an IP address. The advantage is that the driver fails over the connection to an alternate IP address if the first address fails.

To use aliases, a local hosts file that maps aliases to IP addresses is required. Aliases cannot be more than eight characters. In the hosts file, you must specify the aliases and map each of them to an IP address in the order that you want the driver to attempt the connections. For example:

```
167.56.78.1 (NCR5100COP1)
167.56.78.2 (NCR5100COP2)
167.56.78.3 (NCR5100COP3)
```

where NCR5100 is an alias and COP<sub>*n*</sub> (where *n* = 1, 2, 3, ..., 128) is a suffix that sets the order of failover connection attempts. The eight-character limit on the alias does not include the suffix. You can enter a maximum of 128 COP (communications processor) entries per host.

NOTE: Although you must add a COP suffix to the alias in the hosts file, do *not* specify the suffix when entering the alias in the DBCName or Alias field of the Setup dialog box. Only specify the alias.

Default     None

GUI Tab    [General tab](#) on page 824

**Default Database**

Attribute   Database (DB)

Description   The name of the database to which you want to connect.



Valid Values *database\_directory*

where *database\_directory* is the full path name of the directory in which the data files are stored. If no directory is specified, the current working directory is used.

Default None

GUI Tab [General tab](#) on page 824

## Default Role

Attribute TDRole (TDR)

Description Specifies the Teradata role for the LDAP authentication mechanism.

Valid Values *string*

where *string* is a valid role.

Default None

GUI Tab [General tab](#) on page 824

## Description

Attribute Description (n/a)

Description An optional long description of a data source. This description is not used as a runtime connection attribute, but does appear in the ODBC.INI section of the Registry and in the odbc.ini file.

Valid Values *string*

where *string* is a description of a data source.

Default None

GUI Tab [General tab](#) on page 824

## Enable Data Encryption

Attribute EnableDataEncryption (EDE)

Description Determines whether the driver uses data encryption.

Valid Values 1 | 0

If set to 1 (Enabled), the driver encrypts data and communicates with the Teradata gateway using encryption.

NOTE: Before you use this value, verify that the server is encryption capable. Data encryption may adversely affect performance because of the additional overhead (mainly CPU usage) required to encrypt and decrypt data.

If set to 0 (Disabled), the driver does not encrypt data except for logon information.

Default 0 (Disabled)

GUI Tab [Options tab](#) on page 825

## Enable Extended Statement Information

Attribute	EnableExtendedStmtInfo (EESI)
Description	Determines whether the driver supports extended statement information.
Valid Values	1   0
	<p>If set to 1 (Enabled), the driver queries the server to see if it supports the Statement Information parcel. If the server supports the Statement Information parcel, the driver requests the Statement Information parcel and enables auto-generated key retrieval and SQLDescribeParam support. Use this value if you want to enable the Return Generated Keys option.</p> <p>If set to 0 (Disabled), the driver does not attempt to expose auto-generated key retrieval or SQLDescribeParam.</p>
Default	0 (Disabled)
GUI Tab	<a href="#">Options tab</a> on page 825

## Enable LOBs

Attribute	EnableLOBs (EL)
Description	Determines whether the driver enforces native LOB data type mapping.
Valid Values	1   0
	<p>If set to 1 (Enabled), the driver enforces native LOB data type mapping as described:</p> <ul style="list-style-type: none"> <li>■ ODBC data type SQL_LONGVARBINARY is mapped to the Teradata BLOB feature.</li> <li>■ ODBC data type SQL_LONGVARCHAR is mapped to the Teradata CLOB feature.</li> </ul> <p>If set to 0 (Disabled), the driver provides backward compatibility for applications without LOB support that are using a version of Teradata Database prior to V2R5.1. The mappings are:</p> <ul style="list-style-type: none"> <li>■ ODBC data type SQL_LONGVARBINARY is mapped to the Teradata VARBYTE(32000) feature.</li> <li>■ ODBC data type SQL_LONGVARCHAR is mapped to the Teradata LONG VARCHAR feature.</li> </ul> <p>This value can improve performance if your application does not send data to, or retrieve it from, LOB columns. You may receive an error if you disable this option and try to retrieve data from a LOB column.</p>
Default	1 (Enabled)
GUI Tab	<a href="#">Options tab</a> on page 825

## Enable Reconnect

Attribute	EnableReconnect (ER)
Description	Determines whether the driver will reconnect after a system crash or reset is detected.

Valid Values 1 | 0

If set to 1 (Enabled), the driver attempts to reconnect to the saved sessions; however, sessions cannot be reconnected until the Teradata system is available. After a session has been reconnected, applications can expect to receive error messages describing why the ODBC function failed, as well as a status report describing the post-recovery state.

If set to 0 (Disabled), the driver does not attempt to reconnect to the saved sessions.

Default 0 (Disabled)

GUI Tab [Options tab](#) on page 825

## IANAAppCodePage

Attribute IANAAppCodePage (IACP)

Description An Internet Assigned Numbers Authority (IANA) value. You must specify a value for this option if your application is not Unicode-enabled or if your database character set is not Unicode. Refer to [Chapter 4 “Internationalization, Localization, and Unicode”](#) in the *DataDirect Connect Series for ODBC Reference* for details.

The driver uses the specified IANA code page to convert "W" (wide) functions to ANSI.

The driver and Driver Manager both check for the value of IANAAppCodePage in the following order:

- In the connection string
- In the Data Source section of the system information file (odbc.ini)
- In the ODBC section of the system information file (odbc.ini)

If the driver does not find an IANAAppCodePage value, the driver uses the default value of 4 (ISO 8859-1 Latin-1).

Valid Values *IANA\_code\_page*

where *IANA\_code\_page* is one of the valid values listed in [Chapter 1 “Values for the Attribute IANAAppCodePage”](#) in the *DataDirect Connect Series for ODBC Reference*. The value must match the database character encoding and the system locale.

Default 4 (ISO 8559-1 Latin-1)

GUI Tab [Advanced tab](#) on page 826

## Integrated Security

Attribute IntegratedSecurity (IS)

Description Determines whether the driver allows the user to connect to the database using Single Sign On (SSO) through an authentication mechanism that supports SSO.

Valid Values Yes | No

If set to Yes (Enabled), SSO is allowed. The driver uses the operating system user ID and password.

If set to No (Disabled), you must specify a value for the UserID option.

Default No (Disabled)  
 GUI Tab [General tab](#) on page 824

## Login Timeout

Attribute LoginTimeout (LTO)  
 Description The number of seconds to wait when establishing a virtual circuit with Teradata for login.  
 Valid Values  $x$

where  $x$  is a positive integer.

If set to  $x$ , the driver waits the specified number of seconds.

Default 20  
 GUI Tab [Advanced tab](#) on page 826

## Map Call Escape to Exec

Attribute MapCallEscapeToExec (MCETE)  
 Description Determines whether the driver converts the `{CALL <name> (...)}` statement to `EXEC name(...)`.  
 Valid Values Yes | No

If set to Yes (Enabled), the driver considers the `{CALL <name> (...)}` statement as the SQL for MACRO execution and converts it to `EXEC name(...)`.

If set to No (Disabled), the driver does not convert `{CALL name(...)}` statements to `EXEC name(...)`, and considers them as CALL statements for Stored Procedure Execution.

Default No (Disabled)  
 GUI Tab [Options tab](#) on page 825

## Maximum Response Buffer Size

Attribute MaxRespSize (MRS)  
 Description The size of the Teradata response buffer used for SQL requests. This value may be adjusted dynamically if Teradata cannot send a result within the defined size.  
 Valid Values A positive integer from 1 to 65477

If using a slow TCP/IP interface, such as PPP or SLIP, enter a smaller value. If you expect to retrieve large result sets in a LAN environment, set a larger value.

Default 8192  
 GUI Tab [Advanced tab](#) on page 826

## Name

Attribute	DataSourceName (DSN)
Description	The name of a data source in your Windows Registry or odbc.ini file.
Valid Values	<i>string</i>  where <i>string</i> is the name of a data source.
Default	None
GUI Tab	<a href="#">General tab</a> on page 824

## Password

Attribute	Password (PWD)
Description	The password that the application uses to connect to your database. The Password option cannot be specified through the driver Setup dialog box and should not be stored in a data source. It is specified through the Logon dialog box or a connection string.
Valid Values	<i>pwd</i>  where <i>pwd</i> is a valid password.
Default	None
GUI Tab	n/a

## Port Number

Attribute	PortNumber (PORT)
Description	The port number of the server listener.
Valid Values	<i>port_name</i>  where the <i>port_name</i> is the port number of the server listener. Check with your database administrator for the correct number.
Default	1025
GUI Tab	<a href="#">Advanced tab</a> on page 826

## ProcedureWithPrintStmt

Attribute	PrintOption (PO)
Description	Determines whether the driver activates the print option when creating stored procedures.
Valid Values	P   N  If set to P (Print), the driver activates the print option.  If set to N (No), the driver does not activate the print option.
Default	N (No)
GUI Tab	<a href="#">Advanced tab</a> on page 826

## ProcedureWithSPLSource

Attribute	ProcedureWithSPLSource (PWSS)
Description	Determines whether the driver specifies SPL text when creating stored procedures.
Valid Values	Y   N  If set to Y (Yes), the driver specifies SPL text.  If set to N (No), the driver does not specify SPL text.
Default	Y (Yes)
GUI Tab	<a href="#">Advanced tab</a> on page 826

## Profile

Attribute	TDProfile (TDP)
Description	Specifies the Teradata profile for the LDAP authentication mechanism.
Valid Values	<i>string</i>  where <i>string</i> is a valid profile.
Default	None
GUI Tab	<a href="#">General tab</a> on page 824

## Realm

Attribute	AuthenticationDomain (AD)
Description	Specifies the domain appropriate to the selected authentication mechanism
Valid Values	<i>string</i>  where <i>string</i> is a valid domain.
Default	None
GUI Tab	None

## Report Codepage Conversion Errors

Attribute	ReportCodepageConversionErrors (RCCE)
Description	Specifies how the driver handles code page conversion errors that occur when a character cannot be converted from one character set to another.  An error message or warning can occur if an ODBC call causes a conversion error, or if an error occurs during code page conversions to and from the database or to and from the application. The error or warning generated is <code>Code page conversion error encountered</code> . In the case of parameter data conversion errors, the driver adds the following sentence: <code>Error in parameter x</code> , where <i>x</i> is the parameter number. The standard rules for returning specific row and column errors for bulk operations apply.

Valid Values 0 | 1 | 2

If set to 0 (Ignore Errors), the driver substitutes 0x1A for each character that cannot be converted and does not return a warning or error.

If set to 1 (Return Error), the driver returns an error instead of substituting 0x1A for unconverted characters.

If set to 2 (Return Warning), the driver substitutes 0x1A for each character that cannot be converted and returns a warning.

Default 0 (Ignore Errors)

GUI Tab [Options tab](#) on page 825

## Security Mechanism

Attribute SecurityMechanism (SECM)

Description The authentication method to be used by the driver for connections to the database.

Valid Values TD1 | TD2 | Idap | KRB5 | KRB5C | NTLM | NTLMC

If set to TD1, the driver uses Teradata 1.

If set to TD2, the driver uses Teradata 2.

If set to Idap, the driver uses LDAP.

If set to KRB5, the driver uses Kerberos on Windows clients working with Windows servers if the server is V2R6.0.

If set to KRB5C, the driver uses Kerberos Compatibility on Windows clients working with Windows servers if the server is pre-V2R6.0.

If set to NTLM, the driver uses NTLM on Windows clients working with Windows servers if the server is V2R6.0.

If set to NTLMC, the driver uses NTLM Compatibility on Windows clients working with Windows servers if the server is pre-V2R6.0.

NOTE: Kerberos and NTLM are enabled through the Teradata client. See your Teradata documentation for requirements.

The following options may appear, based on the selected method:

### ■ *No mechanism selected*

- **User name:** A user name for the default Teradata database. If TeraSSO allows fully qualified user names, the user name may contain a domain or realm, for example, {judy@linedata}. Values containing a character like @ must be enclosed in braces.

### ■ *KRB5 and KRB5C*

- **Authentication UserID:** The Kerberos user ID.
- **Realm:** The Kerberos domain. (The equivalent connection string attribute is AuthenticationDomain.)

- *LDAP*
  - **Authentication UserID:** The LDAP user ID.
  - **Realm:** The LDAP domain. (The equivalent connection string attribute is AuthenticationDomain.)
  - **TD User name:** The Teradata user name.
  - **Profile:** The Teradata Profile. (The equivalent connection string attribute is TDProfile.)
  - **Default Role:** The Teradata Role. (The equivalent connection string attribute is TDRole.)
- *NTLM and NTLMC*
  - **Authentication UserID:** The NTLM user ID.
  - **Realm:** The NTLM domain. (The equivalent connection string attribute is AuthenticationDomain.)
- *TD1 and TD2*
  - **Authentication UserID:** The TD1 or TD2 user ID.

Other parameters for the authentication mechanism can be entered in the Security Parameter field.

Default    None  
GUI Tab    [General tab](#) on page 824

Security Parameter

Attribute    SecurityParameter (SP)  
Description    A string that is passed as a parameter to the authentication method. The string is ignored by the ODBC driver and is passed to the TeraSSO function that is called to set the authentication method.  
Valid Values    *string*  
  
                  where *string* is a string of characters. The characters [ ] { } ( ) , ; ? \* = ! @ must be enclosed in curly braces.  
  
Default    None  
GUI Tab    [General tab](#) on page 824

Session Character Set

Attribute    CharSet (CS)  
Description    A character set used to override the Teradata character set.  
Valid Values    ASCII | UTF16 (valid only for V2R6.x servers) | LATIN1252\_0A | LATIN9\_0A | LATIN1\_0A | Shift-JIS | EUC | BIG5 | GB | NetworkKorean  
  
                  The specified character set must be installed on the database.



Default ASCII  
 GUI Tab [General tab](#) on page 824

## Show Selectable Tables

Attribute ShowSelectableTables (SST)  
 Description Determines whether the driver supports X views.  
 Valid Values Yes | No

If set to Yes (Enabled), SQLTables() and SQLProcedures() use dbc.tablesX and dbc.databasesX instead of dbc.tables and dbc.databases. Also, SQLColumns() and SQLProcedureColumns() use dbc.columnsX instead of dbc.columns. SqlStatistics() uses dbc.tablesSizeX instead of dbc.tablesSize. The X tables only contain information that the user has permission to access. These tables are optional for Teradata, so verify that they exist.

If set to No (Disabled), SQLTables() and SQLProcedures() use dbc.tables and dbc.databases. Also, SQLColumns() and SQLProcedureColumns() use dbc.columns. SqlStatistics() uses dbc.tablesSize.

Default Yes (Enabled)  
 GUI Tab [Options tab](#) on page 825

## TDUserName

Attribute TDUserName (TDUN)  
 Description Specifies the Teradata user name for the LDAP authentication mechanism.  
 Valid Values *user\_name*

where *user\_name* is a valid user name.

Default None  
 GUI Tab [General tab](#) on page 824

## UserID

Attribute UserID (UID)  
 Description The default user ID that is used to connect to your database. Your ODBC application may override this value or you may override it in the logon dialog box or connection string.  
 Valid Values *userid*

where *userid* is a valid user ID with permissions to access the database.

The user name is interpreted in the context of the authentication mechanism. If, for example, the authentication mechanism is NTLM, the user name is assumed to be a Windows user name.

If TeraSSO allows fully qualified user names, the user name may contain a domain or realm, for example, {judy@linedata}. Values containing a character such as @ must be enclosed in curly braces.

SSO is indicated by the absence of a UserID.

Default None

GUI Tab [General tab](#) on page 824

## Data Types

[Table 24-2](#) shows how the Teradata data types map to the standard ODBC data types.

**Table 24-2. Teradata Data Types**

Teradata	ODBC
Blob <sup>1</sup>	SQL_LONGVARBINARY
Bigint <sup>2</sup>	SQL_BIGINT
Byte	SQL_BIT
Byteint	SQL_TINYINT
Char	SQL_CHAR
Clob <sup>3</sup>	SQL_LONGVARCHAR
Date	SQL_TYPE_DATE
Decimal <sup>4</sup>	SQL_DECIMAL
Double	SQL_DOUBLE
Float	SQL_FLOAT
Integer	SQL_INTEGER
Interval day	SQL_INTERVAL_DAY
Interval day to hour	SQL_INTERVAL_DAY_TO_HOUR
Interval day to minute	SQL_INTERVAL_DAY_TO_MINUTE
Interval day to second	SQL_INTERVAL_DAY_TO_SECOND
Interval hour	SQL_INTERVAL_HOUR
Interval hour to minute	SQL_INTERVAL_HOUR_TO_MINUTE
Interval hour to second	SQL_INTERVAL_HOUR_TO_SECOND
Interval minute <sup>5</sup>	SQL_INTERVAL_MINUTE
Interval minute to second	SQL_INTERVAL_MINUTE_TO_SECOND
Interval month <sup>5</sup>	SQL_INTERVAL_MONTH
Interval second	SQL_INTERVAL_SECOND
Interval year	SQL_INTERVAL_YEAR
Interval year to month	SQL_INTERVAL_YEAR_TO_MONTH
Numeric	SQL_NUMERIC

**Table 24-2. Teradata Data Types** (cont.)

Teradata	ODBC
Real	SQL_REAL
Smallint	SQL_SMALLINT
Time	SQL_TYPE_TIME
Timestamp	SQL_TYPE_TIMESTAMP
Varchar	SQL_VARCHAR

1. If no LOB support, VARBYTE(32000).
2. Supported only on Teradata 6.2 and higher.
3. If no LOB support, LONGVARCHAR.
4. Precision of 18 unless on a Teradata 6.2 or higher server that supports large decimal types.
5. Supported only on Teradata 6.2 and higher when EnableExtendedStmtInfo is enabled.

See ["Retrieving Data Type Information" on page 59](#) for information about retrieving data types.

## Unicode Support

The driver supports Unicode data types. [Table 24-3](#) shows how the Teradata data types map to the Unicode data types, but only when CharacterSet is set to UTF-16.

**Table 24-3. Teradata Unicode Data Types**

Teradata	Unicode
char () charset Unicode	SQL_WCHAR
clob charset Unicode	SQL_WLONGVARCHAR
varchar () charset Unicode	SQL_WVARCHAR

The driver supports the Unicode ODBC W (Wide) function calls, such as SQLConnectW. This allows the Driver Manager to transmit these calls directly to the driver. Otherwise, the Driver Manager would incur the additional overhead of converting the W calls to ANSI function calls, and vice versa.

See ["UTF-16 Applications on UNIX and Linux" on page 125](#) for related details. Also, refer to [Chapter 4 "Internationalization, Localization, and Unicode"](#) in the *DataDirect Connect Series for ODBC Reference* for a more detailed explanation of Unicode.

---

## Persisting a Result Set as an XML Data File

The driver allows you to persist a result as an XML data file with embedded schema. See ["Persisting a Result Set as an XML Data File" on page 60](#) for details about implementation.

---

## Isolation and Lock Levels Supported

Teradata supports isolation levels 0 (read uncommitted) and 3 (serializable).

Refer to [Chapter 7 "Locking and Isolation Levels"](#) in the *DataDirect Connect Series for ODBC Reference* for details.

---

## SQL Support

The driver supports the minimum SQL grammar.

---

## ODBC Conformance Level

The driver is Level 1 compliant, that is, it supports all ODBC Core and Level 1 functions.

Refer to [Chapter 2 "ODBC API and Scalar Functions"](#) in the *DataDirect Connect Series for ODBC Reference* for a list of the API functions supported by the driver for the Teradata database.

---

## Number of Connections and Statements Supported

The driver supports multiple connections and 16 statements per connection to the Teradata database system.

# Index

## Symbols

# symbol in regular identifiers 659

## A

about

- connection failover 66
- connection pooling 82
- DataDirect Bulk Load 85
- extended connection failover 67
- select failover 69

Account String, Teradata 831

Accounting Info

- DB2 Wire Protocol 152
- Oracle Wire Protocol 266

Action for Undefined Tables

- Btrieve 550
- Text 614

Action, Oracle Wire Protocol 267

Add to Create Table, DB2 Wire Protocol 153

adding connections to a connection pool 83

address, IP 54

Administrator, Data Source

- UNIX and Linux 39, 100
- Windows 36

ADO 2.5 persisted file format for XML driver 630

AIX

See UNIX and Linux

aliasing table references 658

Allow Update and Delete, Text driver 614

Alter Table statement

- Btrieve 558
- dBASE 580
- Text 627

alternate database servers

- about 66
- guidelines for specifying 70

Alternate ID, DB2 Wire Protocol 153

Alternate Servers

- DB2 Wire Protocol 153
- Greenplum Wire Protocol 679
- Informix Wire Protocol 198
- MySQL Wire Protocol 220
- Oracle 490
- Oracle Wire Protocol 267
- PostgreSQL Wire Protocol 326
- Progress OpenEdge Wire Protocol 360
- SQL Server Legacy Wire Protocol 524
- SQL Server Wire Protocol 392
- Sybase IQ Wire Protocol 781
- Sybase Wire Protocol 440

AnsiNPW

- SQL Server Legacy Wire Protocol 525
- SQL Server Wire Protocol 393

ansinull option

- Sybase IQ Wire Protocol 802
- Sybase Wire Protocol 473

Apache Hive database

- best practices 820
- limitations on functionality 820

Apache Hive, driver for

- connection string attributes 812
- creating a linked table with Microsoft Access 815
- data source
  - configuring 805
  - connecting via connection string 810
  - connecting via logon dialog box 811

data types 817

driver requirements 805

enabling manual transactions 816

Extended Options 808

isolation levels 819

locking levels 819

ODBC conformance 819

performance considerations 817

Remove Column Qualifiers 815

SQL grammar support 819

stored procedures 818

Unicode support 818

using arrays of parameters 819

Application Name

- DB2 Wire Protocol 154
- Oracle Wire Protocol 267
- SQL Server Legacy Wire Protocol 525
- SQL Server Wire Protocol 392, 393
- Sybase IQ Wire Protocol 782
- Sybase Wire Protocol 440

Application Using Threads

- Btrieve 550
- DB2 Wire Protocol 154
- dBASE 570
- Greenplum Wire Protocol 679
- Informix 596
- Informix Wire Protocol 199
- MySQL Wire Protocol 221
- Oracle 491
- Oracle Wire Protocol 268
- PostgreSQL Wire Protocol 327
- Salesforce 719
- SQL Server Wire Protocol 394
- Sybase IQ Wire Protocol 782
- Sybase Wire Protocol 441
- Text 615

array binding, use in bulk load operations 94

- Array Size
    - Apache Hive, driver for 812
    - Btrieve 550
    - Oracle 491
    - Oracle Wire Protocol 268
  - arrays of parameters
    - Apache Hive Wire Protocol 819
    - DB2 Wire Protocol 190
    - Greenplum Wire Protocol 698
    - Oracle 514
    - Oracle Wire Protocol 314
    - PostgreSQL Wire Protocol 350
    - SQL Server Legacy Wire Protocol 539
    - SQL Server Wire Protocol 420
    - Sybase IQ Wire Protocol 804
    - Sybase Wire Protocol 475
  - ArraySize
    - Apache Hive, driver for 812
    - Progress OpenEdge Wire Protocol 365
    - Sybase IQ Wire Protocol 788
    - Sybase Wire Protocol 451
  - AttachDBFileName, SQL Server Legacy Wire Protocol 525
  - authentication
    - SSL client 80
    - SSL server 80
    - user 77
  - Authentication Method
    - DB2 Wire Protocol 155
    - Oracle Wire Protocol 269
    - SQL Server Wire Protocol 394
    - Sybase Wire Protocol 441
  - AuthenticationDomain, Teradata 838
  - AuthenticationPassword, Teradata 831
  - AuthenticationUserID, Teradata 831
  - AutoTranslate, SQL Server Legacy Wire Protocol 526
- ## B
- batch inserts, using bulk load for 95
  - Batch Size
    - DB2 Wire Protocol 155
    - Oracle Wire Protocol 269
    - SQL Server Wire Protocol 394
    - Sybase Wire Protocol 442
  - BIGTIME data type, Sybase Wire Protocol 451
  - bind packages, DB2 Wire Protocol 130
  - binding
    - dynamic parameters 55
    - parameter markers 55
  - Btrieve driver
    - Alter Table statement 558
    - column names 557
    - connection string attributes 549
    - connections supported 559
    - Create Index statement 558
    - data dictionary file 544
    - data source
      - configuring 544
      - connecting via connection string 548
    - data types 555
    - driver requirements 543
    - Drop Index statement 558
    - indexes 556
    - isolation levels 559
    - locking levels 559
    - managing databases 544
    - ODBC conformance 559
    - Rowid pseudo-column 557
    - Scalable SQL 544
    - Select statement 557
    - SQL grammar support 559
    - statements supported 559
    - table structure 548
    - transactions 544
  - Bulk Binary Threshold
    - DB2 Wire Protocol 156
    - Oracle Wire Protocol 269
    - SQL Server Wire Protocol 395
    - Sybase Wire Protocol 442
  - Bulk Character Threshold
    - DB2 Wire Protocol 156
    - Oracle Wire Protocol 270
    - SQL Server Wire Protocol 395
    - Sybase Wire Protocol 442
  - bulk load
    - batch inserts 95
    - bulk data configuration file 92
    - BulkLoadAsync 720
    - BulkLoadConcurrencyMode 720
    - BulkLoadPollInterval 721
    - BulkLoadThreshold 721
    - BulkLoadTimeout 722
    - determining the protocol 94
    - EnableBulkLoad, Salesforce 728
    - external overflow file 95
    - field delimiter
      - DB2 Wire Protocol 167
      - Oracle Wire Protocol 284
      - Salesforce 728
      - SQL Server Wire Protocol 403
      - Sybase Wire Protocol 451
    - overview 85
    - record delimiter
      - DB2 Wire Protocol 176
      - Oracle Wire Protocol 293
      - Salesforce 735
      - SQL Server Wire Protocol 410
      - Sybase Wire Protocol 460
    - sample application 94
    - single inserts/updates/deletes (Salesforce) 96
    - transactions on Sybase 471
    - using for single inserts/updates/deletes 766
    - validating files 91
  - Bulk Load Asynchronous, Salesforce 720
  - Bulk Load Concurrency Mode, Salesforce 720

- bulk load data file
  - setting the maximum size of binary data exported
    - DB2 Wire Protocol 156
    - Oracle Wire Protocol 269
    - SQL Server Wire Protocol 395
    - Sybase Wire Protocol 442
  - setting the maximum size of character data exported
    - DB2 Wire Protocol 156
    - Oracle Wire Protocol 270
    - SQL Server Wire Protocol 395
    - Sybase Wire Protocol 442
- Bulk Options
  - Oracle Wire Protocol 270
  - SQL Server Wire Protocol 396
- BulkBinaryThreshold
  - DB2 Wire Protocol 156
  - Oracle Wire Protocol 269
  - SQL Server Wire Protocol 395
  - Sybase Wire Protocol 442
- BulkLoadBatchSize
  - DB2 Wire Protocol 155
  - Oracle Wire Protocol 269
  - Salesforce 720
  - SQL Server Wire Protocol 394
  - Sybase Wire Protocol 442
- BulkLoadPollInterval, Salesforce 721
- BulkLoadThreshold, Salesforce 721
- BulkLoadTimeout, Salesforce 722

## C

- Cache Size
  - dBASE 570
  - Text 615
- Cached Cursor Limit, Oracle Wire Protocol 271
- Cached Description Limit, Oracle Wire Protocol 271
- caches, client-side 745
- Cancel Detect Interval
  - Informix 596
  - Informix Wire Protocol 199
- Catalog Functions Include Synonyms
  - Oracle 491
  - Oracle Wire Protocol 271
- Catalog Options
  - Oracle 492
  - Oracle Wire Protocol 272
- Catalog Schema, DB2 Wire Protocol 157
- catalog tables, Salesforce driver 748
- Century Boundary, Text 615
- character set conversions 94
- Character Set for CCSID 65535, DB2 Wire Protocol 157
- character string literals 660
- characters, unexpected
  - DB2 Wire Protocol 187
  - Oracle 510
  - Oracle Wire Protocol 311
  - Sybase IQ Wire Protocol 801
  - Sybase Wire Protocol 471
- Charset
  - Sybase IQ Wire Protocol 782
  - Sybase Wire Protocol 443
- Client Host Name
  - DB2 Wire Protocol 158
  - Oracle Wire Protocol 272
- Client ID, Oracle Wire Protocol 273
- client information
  - about 75
  - how databases store 75
  - location used for storing 76
  - storing 76
- client load balancing, about 71
- Client User
  - DB2 Wire Protocol 158
  - Oracle Wire Protocol 273
- Client Version, Oracle 492
- client-side caches 745
- code pages, IBM DB2 181
- Collection, DB2 Wire Protocol 158
- Column Names in First Line, Text 616
- comma-separated value (CSV) format file
  - character set conversions 94
  - use in bulk load 92
- comments, SQL 665
- Concurrent Access Resolution 159
- Config Options, Salesforce 722
- configuration file
  - bulk data 92
  - database 755
- configuring
  - server mode (Salesforce) 759
- connecting
  - proxy server (Salesforce) 758
- Connection Cache Size
  - Sybase IQ Wire Protocol 783
  - Sybase Wire Protocol 443
- connection failover
  - about 65, 66
  - connection retry and 67
  - load balancing and 71
- connection pool
  - adding connections 83
  - creating 82
  - dead connections 84
  - removing connections 83
  - when a physical connection to a server is lost 84
- Connection Pooling
  - DB2 Wire Protocol 159
  - Greenplum Wire Protocol 680
  - MySQL Wire Protocol 221
  - Oracle 493
  - Oracle Wire Protocol 273
  - PostgreSQL Wire Protocol 327
  - Salesforce 725
  - SQL Server Wire Protocol 396
  - Sybase IQ Wire Protocol 783
  - Sybase Wire Protocol 444

- Connection Reset
  - DB2 Wire Protocol 160
  - Greenplum Wire Protocol 680
  - MySQL Wire Protocol 221
  - Oracle 493
  - Oracle Wire Protocol 273, 274
  - PostgreSQL Wire Protocol 327
  - Salesforce 725
  - SQL Server Wire Protocol 397
  - Sybase IQ Wire Protocol 784
  - Sybase Wire Protocol 444
- Connection Retry Count
  - DB2 Wire Protocol 160
  - Greenplum Wire Protocol 680
  - Informix Wire Protocol 200
  - MySQL Wire Protocol 222
  - Oracle 493
  - Oracle Wire Protocol 274
  - PostgreSQL Wire Protocol 328
  - Progress OpenEdge Wire Protocol 360
  - SQL Server Legacy Wire Protocol 526
  - SQL Server Wire Protocol 397
  - Sybase IQ Wire Protocol 784
  - Sybase Wire Protocol 444
- Connection Retry Delay
  - DB2 Wire Protocol 161
  - Greenplum Wire Protocol 681
  - Informix Wire Protocol 200
  - MySQL Wire Protocol 222
  - Oracle 494
  - Oracle Wire Protocol 275
  - PostgreSQL Wire Protocol 328
  - Progress OpenEdge Wire Protocol 361
  - SQL Server Legacy Wire Protocol 527
  - SQL Server Wire Protocol 398
  - Sybase IQ Wire Protocol 784
  - Sybase Wire Protocol 445
- connection retry, about 67, 72
- connection string attributes
  - Apache Hive, driver for 812
  - Btrieve 549
  - DB2 Wire Protocol 151
  - dBASE 569
  - Greenplum Wire Protocol 678
  - Informix 595
  - Informix Wire Protocol 198
  - MySQL Wire Protocol 219
  - Oracle 489
  - Oracle Wire Protocol 264
  - PostgreSQL Wire Protocol 325
  - Progress OpenEdge Wire Protocol 359
  - Salesforce 718
  - SQL Server Legacy Wire Protocol
    - UNIX and Linux 523
    - Windows 523
  - SQL Server Wire Protocol 390
  - Sybase IQ Wire Protocol 780
  - Sybase Wire Protocol 438
  - Teradata 830
  - Text 613
  - XML 642, 647
- ConnectionCacheSize (Sybase IQ Wire Protocol) 783
- connections supported
  - Btrieve 559
  - DB2 Wire Protocol 189
  - dBASE 585
  - Greenplum Wire Protocol 698
  - Informix 606
  - Informix Wire Protocol 207
  - MySQL Wire Protocol 239
  - Oracle 514
  - Oracle Wire Protocol 313
  - PostgreSQL Wire Protocol 350
  - Progress OpenEdge Wire Protocol 373
  - Salesforce 768
  - SQL Server Legacy Wire Protocol 539
  - SQL Server Wire Protocol 420
  - Sybase IQ Wire Protocol 804
  - Sybase Wire Protocol 475
  - Teradata 844
  - Text 628
  - XML 658
- contacting Customer Support 31
- conventions, typographical 28
- Create Index statement
  - Btrieve 558
  - dBASE 581
- Create Type
  - dBASE 571
  - FoxPro 571
- CSV
  - See comma-separated value
- Current Function Path, DB2 Wire Protocol 161
- cursor stability, DB2 186
- CursorBehavior, Informix 597
- CursorCacheSize, Sybase Wire Protocol 443
- Customer Support, contacting 31

## D

- data encryption 78
- Data File Extension
  - Btrieve 550
  - dBASE 571
  - Text 616
- data integrity (Oracle Wire Protocol driver) 308
- Data Integrity Level
  - Oracle Wire Protocol 275
- Data Integrity Types
  - Oracle Wire Protocol 276
- Data Island format for XML driver 630



## data source

## configuring 209, 351

- Apache Hive, driver for 805
- Btrieve 544
- DB2 Wire Protocol 132
- dBASE 561
- FoxPro DBC 566
- Greenplum Wire Protocol 669
- Informix 588
- Informix Wire Protocol 191
- Oracle 479
- Oracle Wire Protocol 241
- PostgreSQL Wire Protocol 315
- Salesforce 701
- SQL Server Legacy Wire Protocol 516
- SQL Server Wire Protocol 375
- Sybase IQ Wire Protocol 769
- Sybase Wire Protocol 421
- Teradata 821
- Text 608
- UNIX and Linux 37, 100
- Windows 35
- XML 635

## connecting via connection string

- Apache Hive, driver for 810
- Btrieve 548
- DB2 Wire Protocol 148
- dBASE 568, 612
- Greenplum Wire Protocol 676
- Informix 593
- Informix Wire Protocol 196
- MySQL Wire Protocol 217
- Oracle 487
- Oracle Wire Protocol 261
- PostgreSQL Wire Protocol 323
- Progress OpenEdge Wire Protocol 357
- Salesforce 716
- SQL Server Legacy Wire Protocol 520
- SQL Server Wire Protocol 388
- Sybase IQ Wire Protocol 778
- Sybase Wire Protocol 436
- Teradata 827
- XML 641

## connecting via logon dialog box

- Apache Hive, driver for 811
- DB2 Wire Protocol 149
- Greenplum Wire Protocol 677
- Informix 594
- Informix Wire Protocol 197
- MySQL Wire Protocol 218
- Oracle 488
- Oracle Wire Protocol 262
- PostgreSQL Wire Protocol 324
- Progress OpenEdge Wire Protocol 358
- Salesforce 717
- SQL Server Legacy Wire Protocol 521
- SQL Server Wire Protocol 389
- Sybase IQ Wire Protocol 779
- Sybase Wire Protocol 437

- Teradata 828

- XML 641

## Data Source Administrator

- UNIX and Linux 39, 100
- Windows 36

## Data Source Name

- Apache Hive, driver for 812
- Btrieve 551
- DB2 Wire Protocol 161
- dBASE 572
- Greenplum Wire Protocol 681
- Informix 598
- Informix Wire Protocol 201
- MySQL Wire Protocol 223
- Oracle 494
- Oracle Wire Protocol 276
- PostgreSQL Wire Protocol 328
- Progress OpenEdge Wire Protocol 361
- Salesforce 726
- SQL Server Legacy Wire Protocol 527
- SQL Server Wire Protocol 398
- Sybase IQ Wire Protocol 785
- Sybase Wire Protocol 446
- Text 616
- XML 643

## data types

- Apache Hive, driver for 817
- Btrieve 555
- DB2 Wire Protocol 181
- dBASE 578
- Greenplum Wire Protocol 694
- Informix 602
- Informix Wire Protocol 205
- MySQL Wire Protocol 236
- Oracle 506
- Oracle Wire Protocol 304
- PostgreSQL Wire Protocol 345
- Progress OpenEdge Wire Protocol 371
- Salesforce 742
- SQL Server Legacy Wire Protocol 536
- SQL Server Wire Protocol 415
- Sybase IQ Wire Protocol 799
- Sybase Wire Protocol 467
- Teradata 842
- Text 626
- XML 655

## Database (Salesforce driver) 727

## database configuration file 755

## Database Directory

- Btrieve 551
- dBASE 572
- Text 616

## Database List

- Informix 597
- Sybase IQ Wire Protocol 785
- Sybase Wire Protocol 446

- Database Name
  - Apache Hive, driver for 813
  - DB2 Wire Protocol 161
  - dBASE 572
  - Greenplum Wire Protocol 681
  - Informix 598
  - Informix Wire Protocol 200
  - MySQL Wire Protocol 223
  - PostgreSQL Wire Protocol 329
  - Progress OpenEdge Wire Protocol 361
  - SQL Server Legacy Wire Protocol 527
  - SQL Server Wire Protocol 398
  - Sybase IQ Wire Protocol 786
  - Sybase Wire Protocol 446
- DataDirect Bulk Load
  - character set conversions 94
  - external overflow file 95
  - overview 85
  - Sybase Wire Protocol, configuring 471
- DataDirect Connect for ODBC drivers
  - Btrieve 543
  - DB2 Wire Protocol 129
  - dBASE 561
  - Informix 587
  - Informix Wire Protocol 191
  - MySQL Wire Protocol 209
  - Oracle 477
  - Oracle Wire Protocol 241
  - PostgreSQL Wire Protocol 315
  - SQL Server Legacy Wire Protocol 515
  - SQL Server Wire Protocol 375
  - Sybase IQ Wire Protocol 769
  - Sybase Wire Protocol 421
  - Text 607
  - XML 629
- DataDirect Connect XE and Connect64 XE for ODBC drivers
  - Greenplum Wire Protocol 669
  - Salesforce 699
  - Sybase IQ Wire Protocol 769
  - Teradata 821
- DataDirect Connect64 for ODBC drivers
  - DB2 Wire Protocol 129
  - Informix Wire Protocol 191
  - MySQL Wire Protocol 209
  - Oracle 477
  - Oracle Wire Protocol 241
  - PostgreSQL Wire Protocol 315
  - Progress OpenEdge Wire Protocol 351
  - SQL Server Legacy Wire Protocol 515
  - SQL Server Wire Protocol 375
- DataDirect Connect64 XE for ODBC drivers
  - Apache Hive Wire Protocol 805
- date and time literals 660
- date masks, defining for the Text driver 625
- DB2
  - connection properties
    - for Bulk Load 96
    - for connection failover 72
    - for connection pooling 85
    - for security 81
- High Availability Disaster Recovery (HADR) 70
- IBM code page values 181
- pureScale 188
- DB2 Wire Protocol driver
  - authentication 137
  - bind packages 130
  - connection pool
    - creating 82
    - dead connections 84
  - connection string attributes 151
  - connections supported 189
  - data encryption 137
  - data source
    - configuring 132, 701
    - connecting via connection string 148
    - connecting via logon dialog box 149
  - data types 181
  - driver requirements 129
  - Extended Options 135
  - isolation levels 189
  - locking levels 189
  - ODBC conformance 189, 350
  - persisting result set as XML 189
  - SQL support 189
  - statements supported 189
  - stored procedure support 187
  - unexpected characters 187
  - Unicode support 183
  - Workload Manager (WLM) service class 180, 183
  - workloads
    - overview 183
    - performance 184
  - XML data type 182
  - XQuery expressions 187
- dBASE driver
  - Alter Table statement 580
  - column names 579
  - connection string attributes 569
  - connections supported 585
  - Create Index statement 581
  - data source
    - configuring 561
    - connecting via connection string 568
  - data types 578
  - defining index attributes 576
  - defining index attributes under UNIX 578
  - driver requirements 561
  - Drop Index statement 582
  - isolation levels 585
  - locking 584
  - locking levels 585
  - ODBC conformance 585
  - Pack statement 583
  - Rowid pseudo-column 580
  - Select statement 580
  - SQL grammar support 585
  - statements supported 585
- DBCName List, Teradata 831
- DBCName or Alias, Teradata 832
- ddtestlib tool 99

- dead connections in a connection pool 84
- Decimal Symbol, Text 617
- dedicated bulk protocol 94
- Default Buffer Size for Long/LOB Columns
  - Apache Hive, driver for 813
  - MySQL Wire Protocol 223
  - Oracle 494
  - Oracle Wire Protocol 276
  - Sybase Wire Protocol 447, 786
- Default Database, Teradata 832
- Default Isolation Level
  - DB2 Wire Protocol 162
  - Progress OpenEdge Wire Protocol 361
- Default Logon ID, Btrieve 551
- Default Role, Teradata 833
- Default Table Type, Text 617
- Delete Linked Schema, XML 648
- Delimiter Character, Text 617
- demo tool, XML persistence
  - DB2 Wire Protocol 189
  - Informix 605
  - Informix Wire Protocol 206
  - MySQL Wire Protocol 238
  - Oracle 513
  - Oracle Wire Protocol 312
  - PostgreSQL Wire Protocol 349
  - SQL Server Wire Protocol 419
  - Sybase IQ Wire Protocol 803
  - Sybase Wire Protocol 474
  - Teradata 844
  - XML 657
- demoodbc application 119
- Describe at Prepare
  - Oracle 495
  - Oracle Wire Protocol 277
- Distributed Transaction Coordinator
  - Informix Wire Protocol 206
  - Oracle 511
  - Oracle Wire Protocol 309
  - Sybase Wire Protocol 473
- Distributed Transaction Model
  - Sybase Wire Protocol 447
- distributed transactions
  - Informix 604
  - Informix Wire Protocol 206
  - Oracle 511
  - Oracle Wire Protocol 309
  - Sybase Wire Protocol 473
- documentation, about 29
- Driver Options, XML 643
- driver requirements
  - Apache Hive, driver for 805
  - Btrieve 543
  - DB2 Wire Protocol 129
  - dBASE 561
  - Greenplum Wire Protocol 669
  - Informix 587
  - Informix Wire Protocol 191
  - MySQL Wire Protocol 209
  - Oracle 477

- Oracle Wire Protocol 241
- PostgreSQL Wire Protocol 315
- Progress OpenEdge Wire Protocol 351
- Salesforce 701
- SQL Server Legacy Wire Protocol 515
- SQL Server Wire Protocol 375
- Sybase IQ Wire Protocol 769
- Sybase Wire Protocol 421
- Teradata 821
- Text 607
- XML 629
- drivers
  - about 44
  - version string information 57
- Drop Index statement
  - Btrieve 558
  - dBASE 582
- DTC
  - See distributed transactions
- dynamic parameters, binding 55
- Dynamic Sections 163

## E

- Edition Name, Oracle Wire Protocol 277
- Enable Bulk Load
  - DB2 Wire Protocol 164
  - Oracle Wire Protocol 278
  - SQL Server Wire Protocol 399
  - Sybase Wire Protocol 447
- Enable Bulk Load, Salesforce 728
- Enable Data Encryption, Teradata 833
- Enable Describe Parameter 448
- Enable Extended Statement Information, Teradata 834
- Enable Insert Cursors, Informix 599
- Enable Keyset Cursors
  - Greenplum Wire Protocol 682
  - PostgreSQL Wire Protocol 329
- Enable LOBs, Teradata 834
- Enable N-CHAR Support
  - Oracle 495
  - Oracle Wire Protocol 278
- Enable Quoted Identifiers
  - SQL Server Legacy Wire Protocol 528
  - SQL Server Wire Protocol 399
  - Sybase Wire Protocol 448
- Enable Reconnect, Teradata 834
- Enable Scrollable Cursors
  - Oracle 496
  - Oracle Wire Protocol 279
- Enable Server Result Cache, Oracle Wire Protocol 279
- Enable SQLDescribeParam
  - Apache Hive, driver for 814
  - Greenplum Wire Protocol 682
  - MySQL Wire Protocol 224
  - Oracle 496
  - Oracle Wire Protocol 279
  - PostgreSQL Wire Protocol 330

- Enable Static Cursors for Long Data
  - Oracle 497
  - Oracle Wire Protocol 280
- Enable Timestamp with Timezone
  - Oracle 497
  - Oracle Wire Protocol 280
  - Progress OpenEdge Wire Protocol 362
- encryption
  - data 78
  - SSL 79
- Encryption Method
  - DB2 Wire Protocol 164
  - MySQL Wire Protocol 224
  - Oracle Wire Protocol 281
  - PostgreSQL Wire Protocol 330
  - Progress OpenEdge Wire Protocol 363
  - SQL Server Wire Protocol 400
  - Sybase Wire Protocol 448
- environment-specific information 29, 45
- escape sequences 661
- example
  - bulk load sample application 94
  - character set conversions 94
  - creating a connection pool 82
- Extended Column Metadata
  - Greenplum Wire Protocol 683
  - PostgreSQL Wire Protocol 330
- extended connection failover 67
- Extended Options
  - Apache Hive, driver for 808
  - DB2 Wire Protocol 135
  - Greenplum Wire Protocol 673
  - MySQL Wire Protocol 213
  - Oracle 482
  - Oracle Wire Protocol 245
  - PostgreSQL Wire Protocol 318
  - Progress OpenEdge Wire Protocol 354
  - Salesforce 707
  - SQL Server Wire Protocol 378
  - Sybase IQ Wire Protocol 772
  - Sybase Wire Protocol 424
- Extension Case, dBASE 573
- Extension List, Text 618
- extensions to SQL standards 658
- external overflow file 95

## F

- Failover Granularity
  - DB2 Wire Protocol 165
  - Greenplum Wire Protocol 683
  - MySQL Wire Protocol 225
  - Oracle Wire Protocol 283
  - PostgreSQL Wire Protocol 331
  - Progress OpenEdge Wire Protocol 363
  - SQL Server Wire Protocol 400
  - Sybase IQ Wire Protocol 787
  - Sybase Wire Protocol 449

- Failover Mode
  - DB2 Wire Protocol 165
  - Greenplum Wire Protocol 684
  - MySQL Wire Protocol 225
  - Oracle Wire Protocol 283
  - PostgreSQL Wire Protocol 332
  - Progress OpenEdge Wire Protocol 364
  - SQL Server Wire Protocol 401
  - Sybase IQ Wire Protocol 787
  - Sybase Wire Protocol 450
- Failover Preconnect
  - DB2 Wire Protocol 166
  - Greenplum Wire Protocol 684
  - MySQL Wire Protocol 226
  - Oracle Wire Protocol 284
  - PostgreSQL Wire Protocol 332
  - Progress OpenEdge Wire Protocol 364
  - SQL Server Wire Protocol 402
  - Sybase IQ Wire Protocol 788
  - Sybase Wire Protocol 450
- FailoverNetworkAddress
  - Sybase IQ Wire Protocol 789
  - Sybase Wire Protocol 452
- Fetch Array Size
  - Progress OpenEdge Wire Protocol 365
  - Sybase IQ Wire Protocol 788
  - Sybase Wire Protocol 451
- Fetch Time Stamp With Time Zone as Timestamp, DB2 Wire Protocol 166
- Fetch TSWTZ as Timestamp
  - Greenplum Wire Protocol 685
  - Oracle Wire Protocol 284
  - PostgreSQL Wire Protocol 333
  - SQL Server Legacy Wire Protocol 528
  - SQL Server Wire Protocol 402
- Fetch TWFS as Time
  - Greenplum Wire Protocol 686
  - PostgreSQL Wire Protocol 333
  - SQL Server Legacy Wire Protocol 529
  - SQL Server Wire Protocol 402
  - Sybase Wire Protocol 451
- FetchRefCursors
  - Greenplum Wire Protocol 685
  - PostgreSQL Wire Protocol 332
- FetchSize, Salesforce 728
- Field Delimiter
  - DB2 Wire Protocol 167
  - Oracle Wire Protocol 284
  - Salesforce 728
  - SQL Server Wire Protocol 403
  - Sybase Wire Protocol 451
- File Open Cache
  - Btrieve 552
  - dBASE 573
  - Text 618
- Flush Every Change, XML 648
- formats, for text files 607
- FoxPro
  - configuring DBC data source 566
  - SQL statements 583

## G

- Get DB List From Informix, Informix 599
- grammar, tokens used in SQL 659
- Grant Execute to 167
- Greenplum Wire Protocol
  - Extended Options 673
- Greenplum Wire Protocol
  - Enable Keyset Cursors 682
  - Keyset Cursor Options 687
- Greenplum Wire Protocol driver
  - connection string attributes 678
  - connections supported 698
  - data source
    - configuring 669
    - connecting via connection string 676
    - connecting via logon dialog box 677
  - data types 694
  - driver requirements 669
  - isolation levels 697
  - locking levels 697
  - ODBC conformance 698
  - performance considerations 693
  - statements supported 698
  - stored procedures 696
  - Unicode support 695
- GSS Client Library
  - DB2 Wire Protocol 168
  - Oracle Wire Protocol 285
  - SQL Server Wire Protocol 403
  - Sybase Wire Protocol 452
- GUID literal 660
- guidelines for primary and alternate servers 70

## H

- HA Failover server, defining
  - Sybase 452
  - Sybase IQ 789
- hex literals 660
- hierarchical-formatted XML file support 630
- High Availability Disaster Recovery (HADR) 70
- hints, using with XML driver 653
- Host
  - MySQL Wire Protocol 226
  - Progress OpenEdge Wire Protocol 365
- Host Name
  - Greenplum Wire Protocol 686
  - Informix 599
  - Informix Wire Protocol 201
  - PostgreSQL Wire Protocol 334
  - SQL Server Wire Protocol 404
- Host Name In Certificate
  - DB2 Wire Protocol 168
  - MySQL Wire Protocol 226
  - Oracle Wire Protocol 286
  - PostgreSQL Wire Protocol 334
  - Progress OpenEdge Wire Protocol 365

- SQL Server Wire Protocol 404
- Sybase Wire Protocol 453
- HP-UX
  - See UNIX and Linux

## I

- IANAAppCodePage
  - DB2 Wire Protocol 169
  - dBASE 574
  - Greenplum Wire Protocol 686
  - Informix 599
  - Informix Wire Protocol 201
  - MySQL Wire Protocol 227
  - Oracle 497
  - Oracle Wire Protocol 286
  - PostgreSQL Wire Protocol 335
  - Progress OpenEdge Wire Protocol 366
  - SQL Server Legacy Wire Protocol 529
  - SQL Server Wire Protocol 405
  - Sybase IQ Wire Protocol 789
  - Sybase Wire Protocol 454
  - Teradata 835
  - Text 619
- identifiers
  - # symbol in regular 659
  - delimited 659
  - regular 659
- identifiers, using 754
- Include Files with Matching Extensions, Text 619
- Informix driver
  - See *also* Informix Wire Protocol driver
  - connection string attributes 595
  - connections supported 606
  - data source
    - configuring 588
    - connecting via connection string 593
    - connecting via logon dialog box 594
  - data types 602
  - driver requirements 587
  - enabling MTS 604
  - isolation levels 605
  - locking levels 605
  - MTS support 604
  - ODBC conformance 605, 606
  - performance considerations 602
  - persisting result set as XML 605
  - statements supported 606
- Informix Wire Protocol driver
  - See *also* Informix driver
  - connection string attributes 198
  - connections supported 207
  - data source
    - configuring 191
    - connecting via connection string 196
    - connecting via logon dialog box 197
  - data types 205
  - driver requirements 191

- enabling MTS 206
- isolation levels 207
- locking levels 207
- ODBC conformance 207
- performance considerations 204
- persisting result set as XML 206
- SQL grammar supported 207
- statements supported 207
- Initialization String
  - Greenplum Wire Protocol 687
  - Oracle Wire Protocol 287
  - PostgreSQL Wire Protocol 335
  - Salesforce 729
  - SQL Server Wire Protocol 405
  - Sybase IQ Wire Protocol 790
  - Sybase Wire Protocol 454
- Insert statement, XML 658
- integer numbers, XML 659
- Integrated Security, Teradata 835
- integrity (Oracle Wire Protocol driver) 308
- Interactive Client, MySQL Wire Protocol 228
- Interfaces File
  - Sybase IQ Wire Protocol 790
  - Sybase Wire Protocol 455
- InterfacesFileServerName
  - Sybase IQ Wire Protocol 796
  - Sybase Wire Protocol 461
- International Sort
  - Btrieve 552
  - dBASE 574
  - Text 620
- International Sort Order, XML 644
- Ip Address, DB2 Wire Protocol 169
- IP addresses 54
- IPv4 54
- IPv6 54
- isolation levels
  - Apache Hive, driver for 819
  - Btrieve 559
  - DB2 Wire Protocol 189
  - dBASE 585
  - Greenplum Wire Protocol 697
  - Informix 605
  - Informix Wire Protocol 207
  - MySQL Wire Protocol 238
  - Oracle Wire Protocol 313, 513
  - PostgreSQL Wire Protocol 349
  - Salesforce 767
  - SQL Server Legacy Wire Protocol 538
  - SQL Server Wire Protocol 419
  - Sybase IQ Wire Protocol 803
  - Sybase Wire Protocol 474
  - Teradata 844
- ivtestlib tool 99

## J

- Java Logging for the SQL Engine Server (Salesforce) 762
- Java Path (Salesforce driver) 761
- joins, XML 658
- JVMArgs, Salesforce 730

## K

- Kerberos authentication
  - kinit command 78
  - Ticket Granting Ticket (TGT) 77
- Key Password
  - DB2 Wire Protocol 170
  - MySQL Wire Protocol 228
  - Oracle Wire Protocol 287
  - PostgreSQL Wire Protocol 336
- Keyset Cursor Options
  - Greenplum Wire Protocol 687
  - PostgreSQL Wire Protocol 337
- Keystore
  - DB2 Wire Protocol 170
  - MySQL Wire Protocol 228
  - Oracle Wire Protocol 287
  - PostgreSQL Wire Protocol 336
- Keystore Password
  - DB2 Wire Protocol 170
  - MySQL Wire Protocol 229
  - Oracle Wire Protocol 288
  - PostgreSQL Wire Protocol 336
- keywords, reserved, XML 661
- kinit command, Kerberos authentication 78

## L

- Language
  - SQL Server Legacy Wire Protocol 530
  - SQL Server Wire Protocol 406
  - Sybase IQ Wire Protocol 791
  - Sybase Wire Protocol 455
- LDAP authentication method 833
- Left Outer Joins, XML 658
- library path environment variable
  - setting on UNIX/Linux (Salesforce) 38
  - setting on Windows (Salesforce) 35
- Linux
  - See UNIX and Linux
- literal tokens, XML 660
- literals
  - character string 660
  - date and time 660
  - GUID 660
  - hex 660
  - ODBC time and date 660
  - time 660
  - timestamp 660

- Load Balance Timeout
  - DB2 Wire Protocol 171
  - Greenplum Wire Protocol 688
  - MySQL Wire Protocol 229
  - Oracle 498
  - Oracle Wire Protocol 288
  - PostgreSQL Wire Protocol 337
  - SQL Server Wire Protocol 406
  - Sybase IQ Wire Protocol 791
  - Sybase Wire Protocol 455
- Load Balancing
  - DB2 Wire Protocol 171
  - Greenplum Wire Protocol 688
  - Informix Wire Protocol 202
  - MySQL Wire Protocol 229
  - Oracle 498
  - Oracle Wire Protocol 289
  - PostgreSQL Wire Protocol 338
  - Progress OpenEdge Wire Protocol 366
  - SQL Server Legacy Wire Protocol 530
  - SQL Server Wire Protocol 406
  - Sybase IQ Wire Protocol 791
  - Sybase Wire Protocol 456
- load balancing, about 71
- LoadBalance Timeout, Salesforce 731
- local tables, Salesforce 754
- Local Timezone Offset
  - Oracle 499
  - Oracle Wire Protocol 289
- Location Name, DB2 Wire Protocol 171
- Location Names, XML 644
- location used for storing client information for a connection 76
- location\_name, XML 649
- location\_name.Catalog Type Hint, XML 649
- location\_name.Create Type, XML 651
- location\_name.Initial Catalog, XML 649
- location\_name.Password, XML 650
- location\_name.Require Passwd, XML 650
- location\_name.Resolve External, XML 651
- location\_name.Row Hint, XML 651
- location\_name.Scan Rows, XML 649
- location\_name.Table Hint, XML 652
- location\_name.User ID, XML 652
- location\_name.Validate Schema, XML 652
- locations for XML driver 633
- Lock Compatibility, dBASE 574
- Lock Timeout
  - Oracle 499
  - Oracle Wire Protocol 289
- locking levels
  - Apache Hive, driver for 819
  - Btrieve 559
  - DB2 Wire Protocol 189
  - dBASE 585
  - Greenplum Wire Protocol 697
  - Informix 605
  - Informix Wire Protocol 207
  - MySQL Wire Protocol 238
  - Oracle 513
  - Oracle Wire Protocol 313

- PostgreSQL Wire Protocol 349
- Salesforce 767
- SQL Server Legacy Wire Protocol 538
- SQL Server Wire Protocol 419
- Sybase IQ Wire Protocol 803
- Sybase Wire Protocol 474
- Teradata 844
- Locking, dBASE 575
- Log Config File, Salesforce 731
- Logging, XML 644
- Login Timeout
  - Apache Hive Wire Protocol 814
  - DB2 Wire Protocol 172
  - Greenplum Wire Protocol 689
  - MySQL Wire Protocol 230
  - Oracle Wire Protocol 290
  - PostgreSQL Wire Protocol 338
  - Progress OpenEdge Wire Protocol 367
  - SQL Server Wire Protocol 407
  - Sybase IQ Wire Protocol 792
  - Sybase Wire Protocol 456
  - Teradata 836
- Login Timeout, Salesforce 732
- Logon Domain, Salesforce 732
- lost connections
  - extended connection failover 67
  - recovering work in progress 69
  - select failover 69

## M

- Map Call Escape to Exec, Teradata 836
- mapping objects to tables 745
- masks, date 625
- materialized views 818
  - Oracle 512
  - Oracle Wire protocol 310
- Max Char Size
  - Greenplum Wire Protocol 689
  - PostgreSQL Wire Protocol 338
- Max Long Varchar Size
  - Greenplum Wire Protocol 689
  - PostgreSQL Wire Protocol 339
- Max Pool Size
  - DB2 Wire Protocol 172
  - Greenplum Wire Protocol 690
  - MySQL Wire Protocol 230
  - Oracle 500
  - Oracle Wire Protocol 290
  - PostgreSQL Wire Protocol 339
  - Salesforce 733
  - SQL Server Wire Protocol 407
  - Sybase IQ Wire Protocol 792
  - Sybase Wire Protocol 457
- Max Varchar Size
  - Apache Hive, driver for 815
  - Greenplum Wire Protocol 690
  - PostgreSQL Wire Protocol 339

- Maximum Response Buffer Size, Teradata 836
- Microsoft Data Island format for XML 630
- Min Pool Size
  - DB2 Wire Protocol 173
  - Greenplum Wire Protocol 690
  - MySQL Wire Protocol 231
  - Oracle 500
  - Oracle Wire Protocol 291
  - PostgreSQL Wire Protocol 340
  - Salesforce 733
  - SQL Server Wire Protocol 408
  - Sybase IQ Wire Protocol 793
  - Sybase Wire Protocol 457
- Module, Oracle Wire Protocol 291
- MTS support
  - Informix 604
  - Informix Wire Protocol 206
  - Oracle 511
  - Oracle Wire Protocol 309
  - Sybase Wire Protocol 473
- MySQL Wire Protocol driver
  - connection string attributes 219
  - connections supported 239
  - data source
    - configuring 209
    - connecting via connection string 217
    - connecting via logon dialog box 218
  - data types 236
  - driver requirements 209
  - Extended Options 213
  - isolation levels 238
  - locking levels 238
  - ODBC conformance 239
  - performance considerations 219
  - persisting result set as XML 238
  - SQL grammar support 239
  - statements supported 239
  - Unicode support 237
  - wait timeout 228

## N

- Name, Teradata 837
- Network Address
  - Sybase IQ Wire Protocol 793
  - Sybase Wire Protocol 457
- Network, SQL Server Legacy Wire Protocol 531
- NTLM authentication 78
- NTLMDomain, SQL Server Wire Protocol 399
- numbers
  - integer, XML 659
  - real, XML 660

## O

- object
  - mapping to a table 757
  - mapping to tables 745
- ODBC
  - specification 44
  - time and date literals, XML 660
- ODBC conformance
  - Apache Hive, driver for 819
  - Btrieve 559
  - DB2 Wire Protocol 189
  - dBASE driver 585
  - Greenplum Wire Protocol 698
  - Informix 606
  - Informix Wire Protocol 207
  - MySQL Wire Protocol 239
  - Oracle 513
  - Oracle Wire Protocol 313
  - PostgreSQL Wire Protocol 350
  - Progress OpenEdge Wire Protocol 373
  - Salesforce 768
  - SQL Server Legacy Wire Protocol 539
  - SQL Server Wire Protocol 420
  - Sybase IQ Wire Protocol 804
  - Sybase Wire Protocol 474
  - Teradata 844
  - Text 628
  - XML 657
- odbc.ini (system information) file 100
- OEM to ANSI translation 60, 118
- Open Database Connectivity
  - See ODBC specification
- OpenSSL cipher suites
  - drivers support 79
- Optimize Long Performance, Oracle 500
- OptimizePrepare
  - Sybase Wire Protocol 459
- Oracle driver
  - See *also* Oracle Wire Protocol driver
  - authentication 484
  - connection string attributes 489
  - connections supported 514
  - data encryption 484
  - data source
    - configuring 479
    - connecting via connection string 487
    - connecting via logon dialog box 488
  - data types 506
  - isolation levels 513
  - locking levels 513
  - materialized views 512
  - MTS support 511
  - ODBC conformance 513
  - performance considerations 504
  - persisting result set as XML 513
  - statements supported 514
  - stored procedures 512
  - unexpected characters 510



- Unicode support 509
- XML data type 507
- Oracle Real Application Clusters (RAC) 310, 511
- Oracle Wire Protocol driver
  - See *also* Oracle driver
  - authentication 247
  - client information 303
  - connection pool
    - creating 82
    - dead connections 84
  - connection string attributes 264
  - connections supported 313
  - data encryption 247
  - data source
    - configuring 241
    - connecting via connection string 261
    - connecting via logon dialog box 262
  - data types 304
  - driver requirements 241
  - Extended Options 245
  - isolation levels 313
  - locking levels 313
  - materialized views 310
  - MTS support 309
  - ODBC conformance 313
  - performance considerations 300
  - persisting result set as XML 312
  - SQL grammar support 313
  - statements supported 313
  - stored procedures 311
  - unexpected characters 311
  - Unicode support 306
  - XML data type 305
- Outer Joins, XML 658
- overflow files for bulk load operations 95

## P

- Pack statement, using with dBASE driver 583
- Package Collection, DB2 Wire Protocol 173
- Package Name Prefix, DB2 Wire Protocol 173
- Package Owner, DB2 Wire Protocol 174
- Packet Size
  - SQL Server Wire Protocol 408
  - Sybase IQ Wire Protocol 793
  - Sybase Wire Protocol 458
- parameter markers, binding 55
- parameter metadata (Salesforce) 764
- parameters, using arrays 819
- Password
  - Btrieve 553
  - DB2 Wire Protocol 174
  - Greenplum Wire Protocol 691
  - Informix 600
  - Informix Wire Protocol 202
  - MySQL Wire Protocol 231
  - Oracle 501
  - Oracle Wire Protocol 292
  - PostgreSQL Wire Protocol 340
  - Progress OpenEdge Wire Protocol 367
  - Salesforce 733
  - SQL Server Wire Protocol 409
  - Sybase IQ Wire Protocol 794
  - Sybase Wire Protocol 458
  - Teradata 837
  - XML 650
- performance considerations
  - Apache Hive, driver for 820
  - Apache Hive, driver for 817
  - DB2 Wire Protocol 179
  - Greenplum Wire Protocol 693
  - Informix 602
  - Informix Wire Protocol 204
  - MySQL Wire Protocol 219
  - Oracle 504
  - Oracle Wire Protocol 300
  - PostgreSQL Wire Protocol 344
  - Progress OpenEdge Wire Protocol 370
  - Salesforce 741
  - SQL Server Legacy Wire Protocol 535
  - SQL Server Wire Protocol 413
  - Sybase IQ Wire Protocol 797
  - Sybase Wire Protocol 465, 471
- Performance Wizard
  - product requirements 40
  - starting 41
  - using 41
- persisting a result set as XML
  - DB2 Wire Protocol 189
  - Informix 605
  - Informix Wire Protocol 206
  - MySQL Wire Protocol 238
  - Oracle 513
  - Oracle Wire Protocol 312
  - PostgreSQL Wire Protocol 349
  - SQL Server Wire Protocol 419
  - Sybase IQ Wire Protocol 803
  - Sybase Wire Protocol 474
  - XML 657, 844
- Pervasive.SQL
  - See Btrieve driver
- Port Number
  - Apache Hive Wire Protocol 815
  - Greenplum Wire Protocol 691
  - Informix Wire Protocol 203
  - MySQL Wire Protocol 231
  - Oracle Wire Protocol 292
  - PostgreSQL Wire Protocol 340
  - Progress OpenEdge Wire Protocol 368
  - SQL Server Wire Protocol 409
  - Teradata 837
- PostgreSQL Wire Protocol driver
  - authentication 320
  - connection string attributes 325
  - connections supported 350
  - data encryption 320

- data source
  - configuring 315
  - connecting via connection string 323
  - connecting via logon dialog box 324
- data types 345
- driver requirements 315
- Enable Keyset Cursors 329
- Extended Options 318
- isolation levels 349
- Keyset Cursor Options 337
- locking levels 349
- performance considerations 344
- persisting result set as XML 349
- SQL grammar support 349
- statements supported 350
- stored procedures 347
- Unicode support 346
- XML data type 346
- Prepare Method (Sybase Wire Protocol) 459
- primary server, specifying guidelines for 70
- PrintOption, Teradata 837
- Procedure Returns Results
  - Oracle 501
  - Oracle Wire Protocol 292
- ProcedureWithPrintStmt, Teradata 837
- ProcedureWithSQLSource, Teradata 838
- Profile, Teradata 838
- Program ID
  - DB2 Wire Protocol 175
  - Oracle Wire Protocol 293
- Progress OpenEdge Wire Protocol driver
  - authentication 355
  - connection string attributes 359
  - connections supported 373
  - data encryption 355
  - data source
    - configuring 351
    - connecting via connection string 357
    - connecting via logon dialog box 358
  - data types 371
  - driver requirements 351
  - Extended Options 354
  - isolation levels 372
  - locking levels 372
  - ODBC conformance 373
  - SQL grammar support 373
  - statements supported 373
  - Unicode support 371
- Protocol Type, Informix 600
- Proxy Host, Salesforce 734
- Proxy Password, Salesforce 734
- Proxy Port, Salesforce 735
- proxy server (Salesforce), connecting through 758
- ProxyUser, Salesforce 735
- PWD, SQL Server Legacy Wire Protocol 531

## Q

- Query Timeout
  - DB2 Wire Protocol 175
  - Greenplum Wire Protocol 691
  - MySQL Wire Protocol 232
  - Oracle Wire Protocol 293
  - PostgreSQL Wire Protocol 341
  - Progress OpenEdge Wire Protocol 368
  - SQL Server Wire Protocol 409
  - Sybase IQ Wire Protocol 794
  - Sybase Wire Protocol 459
- QueryLog\_On, SQL Server Legacy Wire Protocol 531
- QueryLogFile, SQL Server Legacy Wire Protocol 531
- QueryLogTime, SQL Server Legacy Wire Protocol 532

## R

- ReadOnly
  - Salesforce 735
  - XML 645
- Real Application Clusters (RAC) 310, 511
- real numbers, XML 660
- Realm, Teradata 838
- Record Delimiter
  - DB2 Wire Protocol 176
  - Oracle Wire Protocol 293
  - Salesforce 735
  - SQL Server Wire Protocol 410
  - Sybase Wire Protocol 460
- refcursor, PostgreSQL Wire Protocol 332, 349
- references, aliasing table, XML 658
- Refresh Schema, Salesforce 736
- RefreshDirtyCache, Salesforce 736
- Regional, SQL Server Legacy Wire Protocol 532
- regular identifiers, XML 659
- Remove Column Qualifiers (Apache Hive, driver for) 815
- removing connections from a connection pool 83
- Report Codepage Conversion Errors
  - DB2 Wire Protocol 176
  - Greenplum Wire Protocol 692
  - MySQL Wire Protocol 232
  - Oracle 501
  - Oracle Wire Protocol 294
  - PostgreSQL Wire Protocol 341
  - Salesforce 737
  - SQL Server Wire Protocol 410
  - Sybase IQ Wire Protocol 795
  - Sybase Wire Protocol 460
  - Teradata 838
  - XML 645
- Report Recycle Bin
  - Oracle 502
  - Oracle Wire Protocol 294
- reports, Salesforce 758
- reserved keywords for XML driver 661
- row hints for XML documents 653

Rowid pseudo-column  
     Btrieve 557  
     dBASE 580  
 Rows to Scan, Text 620

## S

### Salesforce

- connection properties
  - Server Port Number 738
  - SQL Engine Mode 738
- proxy server, connecting through 758
- server mode, configuring 759
- SQL engine, starting 761
- SQL engine, stopping 762
- SQL engine, using 759

Salesforce driver

- behavior when maximum Web service limit is exceeded 739
- bulk load
  - using column mapping 766
  - using for single inserts/updates/deletes 766
- classes, data source and driver 701
- configuring data sources 701
- connecting to data sources 701
- connection string attributes 718
- Create Database 726
- data encryption 708
- data source
  - configuring 701
  - connecting via connection string 716
  - connecting via logon dialog box 717
- data types 742
- enabling manual transactions 739
- errors 767
- Extended Options 707
- isolation levels 767
- JVMClasspath 731
- Logon Domain 732
- mapping objects to tables 757
- number of rows of data fetched for each call 740
- number of times the driver retires a timed out Select request 740
- ODBC conformance 768
- parameter metadata 764
- Password 733
- Proxy Password 734
- Proxy Port 735
- ProxyUser 735
- RefreshDirtyCache 736
- reports 758
- requirements 701
- Security Token 737
- setting the library path environment variable on
  - UNIX/Linux 38
- setting the library path environment variable on
  - Windows 35
- SQL supported 768

- timeout for a Web service request 741
- Unicode support 763
  - using bulk load for single Inserts/Updates/Deletes 766
- SAVEFILE, SQL Server Legacy Wire Protocol 532
- ScanRows, Text 620
- schema
  - deleting externally linked files 648
  - format supported by XML driver 630
- schema mode for XML driver 646
- Schema Mode, XML 645
- Secure Sockets Layer
  - See SSL
- Security Mechanism, Teradata 839
- Security Parameter, Teradata 840
- Security Token, Salesforce 737
- select failover 69
- Select Method
  - Sybase IQ Wire Protocol 795
  - Sybase Wire Protocol 461
- Select statement
  - dBASE 580
  - Text 627
  - XML 658
- Server DB Directory (Salesforce driver) 761
- Server List 502
- server mode (Salesforce), configuring 759
- Server Name
  - Informix 601
  - Informix Wire Protocol 203
  - Oracle 503
  - Oracle Wire Protocol 295
  - Sybase IQ Wire Protocol 796
  - Sybase Wire Protocol 461
- Server Port Number (Salesforce) 738
- Server Process Type, Oracle Wire Protocol 295
- server result cache, Oracle Wire Protocol 279
- SERVER, SQL Server Legacy Wire Protocol 533
- service class
  - Oracle Wire Protocol 303
  - Workload Manager (WLM) 180, 183
- Service Name
  - Informix 601
  - Oracle Wire Protocol 296
- Service Principal Name, Sybase Wire Protocol 461
- Services (Salesforce driver) 761
- Session Character Set, Teradata 840
- setting library path environment variable (Salesforce)
  - on UNIX/Linux 38
  - on Windows 35
- Show Manufactured Schemas, XML 646, 647
- Show Selectable Tables, Teradata 841
- SID, Oracle Wire Protocol 296
- single inserts/updates/deletes, using bulk load for 96, 766
- Snapshot Isolation level
  - SQL Server Legacy Wire Protocol 538
  - SQL Server Wire Protocol 419
- Solaris
  - See UNIX and Linux

- SQL
  - comments 665
  - extensions to ANSI SQL 92 658
  - grammar token definitions 659
  - operators and symbols 661
  - standards, extensions to 658
  - syntax for date and time literals 660
- SQL engine (Salesforce), starting 761
- SQL engine (Salesforce), stopping 762
- SQL engine, using (Salesforce) 759
- Window Azure for SQL Databaser
  - See *also* SQL Server Wire Protocol driver
- SQL Server Legacy Wire Protocol driver
  - connection string attributes
    - UNIX and Linux 523
    - Windows 523
  - connections supported 539
  - data source
    - configuring 516
    - connecting via connection string 520
    - connecting via logon dialog box 521
  - data types 536
  - driver requirements 515
  - isolation levels 538
  - locking levels 538
  - ODBC conformance 539
  - SQL grammar support 419, 538
  - statements supported 539
  - Unicode support 537
- SQL Server Wire Protocol driver
  - authentication 380
  - connection pool
    - creating 82
    - dead connections 84
  - connection string attributes 390
  - connections supported 420
  - data encryption 380
  - data source
    - configuring 375
    - connecting via connection string 388
    - connecting via logon dialog box 389
  - data types 415
  - driver requirements 375
  - Extended Options 378
  - isolation levels 419
  - locking levels 419
  - ODBC conformance 420
  - persisting result set as XML 419
  - statements supported 420
  - Unicode support 416
  - XML data type 417
- SQL standards, extensions to 658
- SQL support
  - Apache Hive Wire Protocol 819
  - Btrieve 559
  - dBASE driver 585
  - Greenplum Wire Protocol 698
  - Informix 605
  - Informix Wire Protocol 207
  - MySQL Wire Protocol 239
  - Oracle 513
  - Oracle Wire Protocol 313
  - PostgreSQL Wire Protocol 349
  - Progress OpenEdge Wire Protocol 373
  - SQL Server Legacy Wire Protocol 419, 538
  - Sybase IQ Wire Protocol 803
  - Sybase Wire Protocol 189, 474
  - Teradata driver 844
  - Text 627
- SQLDescribeParam (Sybase) 448
- SQLEngineMode (Salesforce) 738
- SSL
  - client authentication 80
  - encryption 79
  - server authentication 80
- standards, ODBC specification compliance 44
- starting
  - SQL engine server (Salesforce) 761
- Statement Call Limit, Salesforce 739
- statements supported
  - Btrieve 559
  - DB2 Wire Protocol 189
  - dBASE 585
  - Greenplum Wire Protocol 698
  - Informix 606
  - Informix Wire Protocol 207
  - MySQL Wire Protocol 239
  - Oracle Wire Protocol 313, 514
  - PostgreSQL Wire Protocol 350
  - Progress OpenEdge Wire Protocol 373
  - Salesforce 768
  - SQL Server Legacy Wire Protocol 539
  - SQL Server Wire Protocol 420
  - Sybase IQ Wire Protocol 804
  - Sybase Wire Protocol 475
  - Teradata 844
  - Text 628
  - XML 658
- StatsLog\_On, SQL Server Legacy Wire Protocol 533
- StatsLogFile, SQL Server Legacy Wire Protocol 533
- Stmt Call Limit Behavior, Salesforce 739
- stopping the SQL engine server (Salesforce) 762
- stored procedures
  - Apache Hive, driver for 818
  - DB2 Wire Protocol 187
  - Oracle 512
  - Oracle Wire Protocol 311
  - Sybase Wire Protocol 459
- stored results
  - Greenplum Wire Protocol 696
  - Oracle 512
  - Oracle Wire Protocol 311
  - PostgreSQL Wire Protocol 347
- storing client information 76
- String Describe Type
  - Apache Hive Wire Protocol 816
- string literals, character, XML 660

- Sybase IQ Wire Protocol driver
  - ansinull option 802
  - connection string attributes 780
  - connections supported 804
  - data source
    - configuring 769
    - connecting via connection string 778
    - connecting via logon dialog box 779
  - data types 799
  - driver requirements 769
  - Extended Options 772
  - isolation levels 803
  - locking levels 803
  - NULL values, interpreting 802
  - ODBC conformance 627, 698, 803, 804, 844
  - parameter arrays, emulating 804
  - performance considerations 797
  - persisting result set as XML 803
  - statements supported 804
  - unexpected characters 801
  - Unicode support 800
- Sybase Wire Protocol driver
  - ansinull option 473
  - authentication 426
  - bulk load
    - transactions 471
  - connection pool
    - creating 82
    - dead connections 84
  - connection string attributes 438
  - connections supported 475
  - data encryption 426
  - data source
    - configuring 421
    - connecting via connection string 436
    - connecting via logon dialog box 437
  - data types 467
  - driver requirements 421
  - Extended Options 424
  - isolation levels 474
  - locking levels 474
  - MTS support 473
  - NULL values, interpreting 473
  - ODBC conformance 474
  - performance considerations 465, 471
  - persisting result set as XML 474
  - SQL grammar support 474
  - statements supported 475
  - stored procedures 459
  - unexpected characters 471
  - Unicode support 468
- symbol in regular identifiers, #, XML 659
- symbols, SQL, XML 661
- system information file (.odbc.ini) 100
- system requirements
  - See driver requirements
- SYSTEM\_CACHES catalog table 748
- SYSTEM\_REMOTE\_SESSIONS system table 750
- SYSTEM\_SESSIONINFO system table 752
- SYSTEM\_SESSIONS table 752

## T

- table hints for XML documents 653
- table references, aliasing, XML 658
- table structure, defining
  - Btrieve 553
  - Text 621
  - Text under UNIX 624
- tables
  - local 754
  - mapping objects to 757
- tabular formats supported for XML 630
- Tcp Port, DB2 Wire Protocol 176
- TDProfile, Teradata 838
- TDRole, Teradata 833
- TDUserName, Teradata 841
- Teradata, driver for
  - connections supported 844
  - data source
    - configuring 821
    - connecting via connection string 827
  - data types 842
  - driver requirements 821
  - isolation levels 844
  - locking levels 844
  - ODBC conformance 844
  - persisting result set as XML 844
  - statements supported 844
  - Unicode support 843
- Text driver
  - Alter Table statement 627
  - connection string attributes 613
  - connections supported 628
  - data source
    - configuring 608
    - connecting via connection string 612
  - data types 626
  - date masks 625
  - defining table structure
    - on UNIX 624
    - on Windows 621
  - driver requirements 607
  - formats 607
  - ODBC conformance 628
  - Select statement 627
  - statements supported 628
- Ticket Granting Ticket (TGT), Kerberos authentication 77
- Tightly Coupled Distributed Transactions
  - Sybase Wire Protocol 462
- time literals, XML 660
- timeouts 753
- Timestamp Escape Mapping
  - Oracle 503
  - Oracle Wire Protocol 296
- timestamp literals, XML 660
- TNSNames File, Oracle Wire Protocol 297
- tokens, XML
  - literal 660
  - used in SQL grammar 659
- tracing, ODBC 102

- Transaction Error Behavior
  - Greenplum Wire Protocol 692
  - PostgreSQL Wire Protocol 342
- Transaction Mode, Apache Hive Wire Protocol 816
- Transaction Mode, Salesforce 739
- transactions
  - Btrieve 544
  - bulk copy on Sybase 471
  - effect of record locks with dBASE 584
- translators 63, 118
- Treat Binary Data as Character Data, MySQL Wire Protocol 233
- Trim Blank From Index Name
  - Informix 601
  - Informix Wire Protocol 203
- Truncate Time Type Fractions
  - Sybase IQ Wire Protocol 796
  - Sybase Wire Protocol 462
- Trusted\_Connection, SQL Server Legacy Wire Protocol 534
- Truststore
  - DB2 Wire Protocol 177
  - MySQL Wire Protocol 233
  - Oracle Wire Protocol 298
  - PostgreSQL Wire Protocol 342
  - Progress OpenEdge Wire Protocol 368
  - SQL Server Wire Protocol 411
  - Sybase Wire Protocol 462
- Truststore Password
  - DB2 Wire Protocol 177
  - MySQL Wire Protocol 234
  - Oracle Wire Protocol 298
  - PostgreSQL Wire Protocol 342
  - Progress OpenEdge Wire Protocol 369
  - SQL Server Wire Protocol 411
  - Sybase Wire Protocol 463
- typographical conventions 28

## U

- undefined tables 614
- undocumented connection options
  - Apache Hive, driver for 808
  - DB2 Wire Protocol 135
  - Greenplum Wire Protocol 673
  - MySQL Wire Protocol 213
  - Oracle 482
  - Oracle Wire Protocol 245
  - PostgreSQL Wire Protocol 318
  - Progress OpenEdge Wire Protocol 354
  - Salesforce 707
  - SQL Server Wire Protocol 378
  - Sybase IQ Wire Protocol 772
  - Sybase Wire Protocol 424
- Unicode support
  - Apache Hive, driver for 818
  - DB2 Wire Protocol 183
  - Greenplum Wire Protocol 695
  - MySQL Wire Protocol 237
  - Oracle 509
  - Oracle Wire Protocol 306
  - PostgreSQL Wire Protocol 346
  - Progress OpenEdge 371
  - Salesforce 763
  - SQL Server Legacy Wire Protocol 537
  - SQL Server Wire Protocol 416
  - Sybase IQ Wire Protocol 800
  - Sybase Wire Protocol 468
  - Teradata 843
  - Windows Azure SQL Database 416
  - XML 657
- UNIX and Linux
  - configuring through the system information (odbc.ini) file 103
  - Data Source Administrator 39, 100
  - data source configuration 37, 100
  - driver names 53
  - drivers
    - Apache Hive, driver for 805
    - DB2 Wire Protocol 129
    - dBASE 561
    - Greenplum Wire Protocol 669
    - Informix 587
    - Informix Wire Protocol 191
    - MySQL Wire Protocol 209
    - Oracle 477
    - Oracle Wire Protocol 241
    - PostgreSQL Wire Protocol 315
    - Progress OpenEdge Wire Protocol 351
    - Salesforce 699
    - SQL Server Legacy Wire Protocol 515
    - SQL Server Wire Protocol 375
    - Sybase IQ Wire Protocol 769
    - Sybase Wire Protocol 421
    - Teradata 821
    - Text 607
  - environment
    - DD\_INSTALLDIR 99
    - ddtestlib tool 99
    - introduction 97
    - ivtestlib tool 99
    - library search path 97
    - ODBCINI 98
    - ODBCINST 98
    - system information file (.odbc.ini) 100
    - variables 97
  - system information (odbc.ini) file, configuring through 103
  - system requirements 47
- Update statement
  - XML driver 658
- Use Current Schema for Catalog Functions
  - Apache Hive, driver for 816
  - DB2 Wire Protocol 178
- Use Current Schema for SQLProcedures
  - Oracle 504
  - Oracle Wire Protocol 298
- Use Default Login, Informix 601
- Use Delimited Identifier, Informix Wire Protocol 203
- Use Long Names, dBASE 576

- Use Long Qualifiers
  - Btrieve 553
  - dBASE 576
  - Text 620
- Use NT Authentication, SQL Server Legacy Wire Protocol 534
- Use Snapshot Transactions
  - SQL Server Legacy Wire Protocol 534
  - SQL Server Wire Protocol 411
- Use Wide Character Types, Progress OpenEdge Wire Protocol 369
- used in SQL grammar, tokens, XML 659
- user authentication 77
- User ID, Progress OpenEdge Wire Protocol 369
- User Name
  - Apache Hive, driver for 817
  - DB2 Wire Protocol 178
  - Informix Wire Protocol 204
  - MySQL Wire Protocol 234
  - Oracle Wire Protocol 299
  - PostgreSQL Wire Protocol 343
  - Salesforce 740
  - SQL Server Wire Protocol 412
- user-defined functions, PostgreSQL Wire Protocol 347
- using
  - bulk load for batch inserts 95
  - bulk load for single inserts/updates/deletes 96
  - identifiers 754
  - SQL engine (Salesforce) 759

## V

- Validate Server Certificate
  - DB2 Wire Protocol 178
  - MySQL Wire Protocol 234
  - Oracle Wire Protocol 299
  - PostgreSQL Wire Protocol 343
  - Progress OpenEdge Wire Protocol 369
  - SQL Server Wire Protocol 412
  - Sybase Wire Protocol 463
- validating bulk load files 91
- version string information 57
- views
  - Apache Hive, driver for 818
  - Salesforce 754

## W

- Windows
  - Data Source Administrator 36
  - data source configuration 35
  - driver names 47
  - drivers
    - Apache Hive, driver for 805
    - Btrieve 543
    - DB2 Wire Protocol 129
    - dBASE 561
    - Greenplum Wire Protocol 669

- Informix 587
- Informix Wire Protocol 191
- MySQL Wire Protocol 209
- Oracle 477
- Oracle Wire Protocol 241
- PostgreSQL Wire Protocol 315
- Progress OpenEdge Wire Protocol 351
- Salesforce 699
- SQL Server Legacy Wire Protocol 515
- SQL Server Wire Protocol 375
- Sybase IQ Wire Protocol 769
- Sybase Wire Protocol 421
- Teradata 821
- Text 607
- XML 629
  - system requirements 45
- Windows Azure SQL Database
  - ANSI-defined behaviors 393
  - Authentication Method. 394
  - mapping data types 415
  - Unicode data types 416
  - version supported 375
- Wire Protocol Mode, Oracle Wire Protocol 300
- With Hold Cursors, DB2 Wire Protocol 179
- Workload Manager (WLM) service class 180, 183
- Workstation ID
  - SQL Server Legacy Wire Protocol 535
  - SQL Server Wire Protocol 413
  - Sybase IQ Wire Protocol 797
  - Sybase Wire Protocol 464
- WS Fetch Size, Salesforce 740
- WS Retry Count, Salesforce 740
- WS Timeout, Salesforce 741

## X

- XA Open String Parameters, Sybase Wire Protocol 464
- XA Protocol
  - Informix Wire Protocol 206
  - Oracle 511
  - Oracle Wire Protocol 309
  - Sybase Wire Protocol 473
- XML data file
  - DB2 Wire Protocol 189
  - Informix 605
  - Informix Wire Protocol 206
  - MySQL Wire Protocol 238
  - Oracle Wire Protocol 312
  - PostgreSQL Wire Protocol 349
  - SQL Server Wire Protocol 419
  - Sybase IQ Wire Protocol 803
  - Sybase Wire Protocol 474
  - XML 657

- XML data type
  - DB2 Wire Protocol 182
  - Oracle 507
  - Oracle Wire Protocol 305
  - PostgreSQL Wire Protocol 346
  - SQL Server Wire Protocol 417
- XML Describe Type
  - DB2 Wire Protocol 179
  - PostgreSQL Wire Protocol 344
  - SQL Server Wire Protocol 413
- XML driver
  - connection string attributes 642, 647
  - connections supported 658
  - data source
    - configuring 635
    - connecting via connection string 641
    - connecting via logon dialog box 641
  - data types 655
  - defining locations 633
  - driver requirements 629
  - file formats supported 629
  - hierarchical file format support 630
  - hints for tabular-formatted XML documents 653
  - ODBC conformance 657
  - persisting result set as XML 657
  - reserved keywords 661
  - schema mode 646
  - specifying table names in SQL statements 633
  - SQL statements supported 629
  - statements supported 658
  - tabular file formats supported 630
  - Unicode support 657
- XQuery expressions, DB2 Wire Protocol 187