Feature Data Objects (FDO)

# The Essential FDO

FDO Open Source

June 2006

## Copyright© 2006 Autodesk, Inc.

AUTODESK, INC., MAKES NO WARRANTY, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE REGARDING THESE MATERIALS, AND MAKES SUCH MATERIALS AVAILABLE SOLELY ON AN "AS-IS" BASIS.

IN NO EVENT SHALL AUTODESK, INC., BE LIABLE TO ANYONE FOR SPECIAL, COLLATERAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH OR ARISING OUT OF PURCHASE OR USE OF THESE MATERIALS. THE SOLE AND EXCLUSIVE LIABILITY TO AUTODESK, INC., REGARDLESS OF THE FORM OF ACTION, SHALL NOT EXCEED THE PURCHASE PRICE OF THE MATERIALS DESCRIBED HEREIN.

Autodesk, Inc., reserves the right to revise and improve its products as it sees fit. This publication describes the state of the product at the time of publication, and may not reflect the product at all times in the future.

## Trademarks

Autodesk, Autodesk Map and Autodesk MapGuide are registered trademarks of Autodesk, Inc., in the USA and/or other countries. DWF is a trademark of Autodesk, Inc., in the USA and/or other countries. All other brand names, product names or trademarks belong to their respective holders.

## FDO Third Party Software Program Credits

FDO contains certain technology licensed from third parties. The notices and/or other terms and conditions applicable to or associated with such third party technology are set out below.

Xerces and Xalan are Copyright © 1999-2005, The Apache Software Foundation. Licensed under the Apache License, Version 2.0; you may not use this file except in compliance with the license. You may obtain a copy of the license at the following web address: http://www.apache.org/licenses/LICENSE-2.0. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the license for the specific language governing permissions and limitations under the license.

Libcurl is Copyright © 1996 - 2006, Daniel Stenberg, <daniel@haxx.se>. All rights reserved. Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

Boost is offered under the Boost Software License - Version 1.0, which provides as follows: Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following: The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor. 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, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

GDAL is Copyright © 2000, Frank Warmerdam. 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 limitationthe rights to use, copy, modify, merge, publish, distribute, sublicense,and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

## GOVERNMENT USE

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in FAR 12.212 (Commercial Computer Software-Restricted Rights) and DFAR 227.7202 (Rights in Technical Data and Computer Software), as applicable.

Published By: Autodesk, Inc.
111 McInnis Parkway
San Rafael, CA 94903, USA
Government Use

1 2 3 4 5 6 7 8 9 10

# Contents

# The Essential FDO

1

You can use this document is to get up and running with the Feature Data Objects (FDO) API. The Objectives section provides more information on the topics covered in this document. Comments on this document can be sent to isd.docs@autodesk.com.

## In this chapter

# What Is the Feature Data Objects API?

The FDO API provides a generic interface to a number of back-end data source technologies for storing, retrieving, and updating GIS (Geographic Information System) data.

FDO Geometry is based on the *OpenGIS Simple Features Implementation Specification for SQL* (version 1.1, document 99-049), but is extended to include z and measure dimensions and circular arcs. The FDO XML format for schema is based on the Open GIS Consortium *Geography Markup Language* (version 2.1.1, document 02-009). The schema model is object-based using classes and properties and is independent of any physical storage format.

FDO provides a model for extending this interface to additional data source technologies. The generic API is extensible, and it is possible to add custom commands to a particular provider. A provider is a specifc implementation of the FDO API that provides access to date stored in a particular data source technology. For example, the MySQL provider provides access to GIS data stored in an MySQL database, and the OSGeo FDO Provider for SHP provides access to GIS data stored in a SHP file. The extent to which a provider implements the FDO API is limited by the native capabilities of the underlying data source technology.

A client application creates a connection to a provider and then uses the connection object to create command objects for FDO actions such as Select. The client uses the command object to set the command parameters with the option of leaving some parameters with default values. Where appropriate such as in the case of Select, execution of the command returns a reader object containing the results of the command.

A client application can use the FDO Capabilities API to determine what services a particular provider offers. For example, the OSGeo FDO Provider for SDF supports the insertion of data, and the OSGeo FDO Provider for WMS does not.

The storage of FDO feature data can be modeled as a containment hierarchy:

- A data source may contain one or more data stores.
- A data store may contain one or more feature schema and one or more spatial contexts.
- A feature schema may contain one or more feature classes.

- A feature class contains a feature geometry property, optionally one or more non-feature geometry properties, and optionally other types of properties.

The API is available in two formats, enterprise and open source, on two platforms, Windows and Linux. The enterprise version of the API contains all of the open source components plus some proprietary components. The enterprise version of this document describes both open source and proprietary components. The open source version of this document describes only the open source components.

The enterprise edition is released as part of the Autodesk Map and MapGuide products. The FDO API is not exposed in these products.

The open source edition is released as source code which may be downloaded from the Open Source Geospatial Foundation website (*http://www.osgeo.org*) in the form of gzipped tarfiles or directly from Subversion repositories. The project name is Fdo. The project documentation includes a document about how to build and install the Fdo software from source on both platforms.

# Objectives

The main objective of this document is to provide information to help you install the FDO software and then use it to build applications that connect to and configure providers. “Configure providers” means creating and deleting user accounts, granting and revoking database privileges to users, creating and destroying data stores, and creating and destroying spatial contexts. Not every provider supports configuration. See Command Capabilities (page 14).

The rdbms-based providers with the exception of SQL Server require rdbms client software to be installed and configured in order to make the provider fully functional. This document describes the installation and configuration of these clients. Having a SQL Server client on your machine can be useful so the installation of this client is described as well.

This document identifies the classes and methods you use for connecting to and configuring providers but does not provide details about their use. For more information, see the FDO API reference documentation. Doxygen is used to generate the API reference documentation from comments wrapped in doxygen tags and embedded in the C++ headers.

The *FDO Developer’s Guide* (FDG_FDODevGuide.pdf) describes additional FDO API customization and development features for FDO applications. The developer’s guide expands on the FDO API introduced in this document, further explains the role of a feature provider, and provides detailed information and examples about how to code your application.

## Related Documentation

The reference documentation includes this document as well as the following. The ReadMe and the API reference documents are formatted as HTML or CHM files on Windows and HTML on Linux. This document, the developer's guide and the open source build document are formatted as PDF.

- *FDO SDK ReadMe*
- *FDO Developer's Guide*
- *Building ProductName Open Source*
- *FDO API Reference*
- *FDO API Reference - Managed*
- *OSGeo FDO Provider for ArcSDE API Reference*
- *OSGeo FDO Provider for MySQL API Reference*
- *OSGeo FDO Provider for ODBC API Reference*
- *Autodesk FDO Provider for Oracle API Reference*
- *Autodesk FDO Provider for Raster API Reference*
- *OSGeo FDO Provider for SDF API Reference*
- *OSGeo FDO Provider for SHP API Reference*
- *Autodesk FDO Provider for SQL Server API Reference*
- *OSGeo FDO Provider for WMS API Reference*

**NOTE** There is no API reference document for the WFS Provider because this provider has no WFS-specific public API; all of its functionality is accessed through the FDO API.

# Provider Overview

The following RDBMS-based and file-based FDO providers are included in the FDO SDK. The full provider name is listed here; subsequent references in this document may use a shortened name. Autodesk® providers are only included with commercial Autodesk software. Open source providers are also part of the Open Source Geospatial Foundation (OSGeo).

| Provider Name | Description |
|---|---|
| Autodesk FDO Provider for Oracle | Read/write access to feature data in an Oracle-based data store. Supports spatial indexing, long transactions, and persistent locking. Custom API can gather provider information, transmit client services exceptions, list data stores, and create connection objects. |
| OSGeo FDO Provider for ArcSDE | Read/write access to feature data in an ESRI ArcSDE-based data store (that is, with an underlying Oracle or SQL Server database). Supports describing schema, and inserting, selecting, updating, and deleting feature data in existing schemas; does not support creating or deleting schemas. |
| Autodesk FDO Provider for SQL Server | Read/write access to feature data in a Microsoft SQL Server-based data store. A custom API supports schema read/write access and geospatial and non-geospatial data read/write access. |
| OSGeo FDO Provider for MySQL | Read/write access to feature data in a MySQL-based data store. Supports spatial data types and spatial query operations. Custom API can gather information, transmit exceptions, list data stores, and create connection objects. MySQL architecture supports various storage engines, characteristics, and capabilities. |
| OSGeo FDO Provider for SDF | Read-write access to feature data in an SDF-based data store. Autodesk's geospatial file format, SDF, supports multiple features/attributes, provides high performance for large data sets, and interoperability with other Autodesk products. Supports spatial indexing and is a valid alternative to database storage. Note that this release of the SDF provider supports version 3.0 of the SDF file format. |
| OSGeo FDO Provider for SHP | Read/write access to existing spatial and attribute data in an ESRI SHP-based data store, which consists of separate shape files for geometry, index, and at- |

| Provider Name | Description |
|---|---|
| | tributes. Each SHP file and its associated DBF file is treated as a feature class with a single geometry property. This is a valid alternative to database storage but does not support locking. |
| Autodesk FDO Provider for Raster | Read-only access to feature data in a raster-based file format. Supports various image and GIS data formats (for example, JPEG, PNG, MrSID, and others). Supports georeferenced file-based raster images and file-based grid coverages. Pixel-based images, such as satellite images, are useful underneath vector data. |
| OSGeo FDO Provider for ODBC | Read/write to feature data in a ODBC-based data store. Supports XYZ feature objects and can define feature classes for any relational database table with X, Y, and optionally, Z columns; does not support creating or deleting schema. Object locations are stored in separate properties in the object definition. |
| OSGeo FDO Provider for WFS | Read-only access to feature data in an OGC WFS-based data store. Supports a client/server environment and retrieves geospatial data encoded in GML from one or more Web Feature Services sites. Client/server communication is encoded in XML with the exception of feature geometries, which are encoded in GML. Note that there is no public API documentation for this provider; all WFS functionality is accessible through the base FDO API. |
| OSGeo Provider for WMS | Read-only to feature data in an OGC WMS-based data store. Web Map Service (WMS) produces maps of spatially referenced data dynamically from geographic information, which are generally rendered in PNG, GIF, or JPEG, or as vector-based Scalable Vector Graphics (SVG) or Web Computer Graphics Metafile (WebCGM) formats. |

# System Requirements

## Operating System and RDBMS Client Software

These requiresments are specified in the chapters describing the enterprise and open source providers.

## Development Environment Software

The FDO API is written in C++. The class references in this document are to C++ classes. An equivalent managed class interface is also available in Windows.

### Windows XP, SP2

Both the enterprise and open source editions of the FDO API require the use of Microsoft Visual Studio 2005 for development.

### Red Hat Enterprise Linux 3

The programming environment is GNU.

# C++ Memory Management

Wherever you create an FDO object whose class definition inherits from `GisIDisposable`, wrap it in a `GisPtr`, which is the FDO version of a reference counting smart pointer. When the object goes out of scope, the memory allocated for it is released.

You can use `GisPtr` for your own classes by inheriting from the abstract class `GisIDisposable` and providing an implementation for the `Dispose()` method (typically `delete this;`).

# Collections

You can use FDO collection template classes to store your own objects. The requirements for your collection class and the class used to instantiate the template are the same as those for wrapping a class in a `GisPtr`.

# Getting Started

## Install FDO

### Open Source Edition

The installation of the FDO open source edition is described in the *Building FDO Open Source* document. This document is available at *http://fdo.osgeo.org/documentation.html*. This involves getting the source, building the binaries, and then installing them.

### Enterprise Edition

FDO is installed as part of the Autodesk Map and Autodesk MapGuide installs. The FDO API is not exposed in these products.

Internal Autodesk developers install the SDK on Windows using a Microsoft installer package and on Linux using a gzipped tarfile. The binaries are pre-built. The source code is available in separate gzipped tarfiles.

## Setup the Application Development Environment

### Windows

Check out the settings in the solution (.sln) and project (.vcproj) files found in the FDO source folders.

Use the installation path to set up the environment so that the headers and libraries can be located at compile time, and the shared objects can be located at runtime.

Use the source path to set up the environment so that the debugger can locate program database files (symbol tables) and source files at runtime.

You can make the various files visible by either setting solution or project properties or by setting Visual Studio tool options or by setting an environment variable (PATH).

The first step for the properties method is to select the solution or project in the Solution Explorer, right-click to its pop-up menu and select Properties. The result is that the properties dialog is displayed. If the project properties dialog is displayed, the second step is to expand 'Configuration Properties' in the properties dialog contents pane.

The first step for the tool options method is to select 'Options...' from the Tools menu in the main menubar. The result is that the options dialog is displayed.

## Headers

In the properties dialog contents pane expand 'Configuration Properties' and then 'C/C++'. Select 'General' and edit 'Additional Include Directories'.

In the options dialog contents pane expand 'Projects and Solutions' and then 'VC++ Directories'. Select 'Include files' from the spinbox labeled 'Show directories for:'. Follow the procedure for adding a directory.

## Libraries

In the project properties dialog contents pane expand 'Configuration Properties' and then 'Linker'. Select 'General' and edit 'Additional LibraryDirectories'.

In the options dialog contents pane expand 'Projects and Solutions' and then 'VC++ Directories'. Select 'Library files' from the spinbox labeled 'Show directories for:'. Follow the procedure for adding a directory.

## Dynamically Linked Libraries

In the options dialog contents pane expand 'Projects and Solutions' and then 'VC++ Directories'. Select 'Executables' from the spinbox labeled 'Show directories for:'. Follow the procedure for adding a directory.

Use XP to set the PATH environment variable.

### Symbol Table Files (Program Database Files)

In the options dialog contents pane expand 'Debugging' and then 'Symbols'. Follow the procedure for adding a directory.

### Source

In the solution properties dialog contents pane expand 'Common Properties'. Select 'Debug Source Files'. Follow the procedure for adding a directory.

In the options dialog contents pane expand 'Projects and Solutions' and then 'VC++ Directories'. Select 'Source files' from the spinbox labeled 'Show directories for:'. Follow the procedure for adding a directory.

### Linux

Set LD_LIBRARY_PATH environment variable.

## Access the API Reference Documentation

The API reference documentation has been created by using doxygen to process comments in the header files. When doxygen generates the html files, it distributes them across a generated file structure to allow for quicker access. This prevents direct linking to the class html files.

When you are asked to read FDO class reference documentation, load the FDO C++ API Reference home page, click the Classes tab in the right-hand pane and click on the class name. The home page is located in the <install directory>/Docs/HTML/FDO_API directory.

## Write the Code to Connect to a Provider

Do the following:

1 Create a provider registry

2 Create a connection manager

3 Create a connection

4 Get the connection state

5 Get the connection properties

6 Get values for the connection properties

7 Set the connection properties

8 Open a connection

9 Open a pending connection

## Create a Provider Registry

Read about the `GetProviders` method in the `FdoProviderRegistry` class reference. This method returns the contents of the providers.xml file found in the <install>/bin folder on Windows and in the /usr/local/fdo-<version>/lib directory on Linux.

When you create the connection, you will use one of the strings returned by this method as an argument.

## Create a Connection Manager

The connection manager creates connections.

Read about the `GetConnectionManager` method in the `FdoFeatureAccessManager` class reference.

## Get the Connection Properties

First you get the connection information. Read about the `GetConnectionInfo` method in the `FdoIConnection` class reference.

Then you get the connection properties. Read about the `GetConnectionProperties` method in the `FdoIConnectionInfo` class reference. This method returns a connection property dictionary.

Read about the methods used to access the information in the connection properties dictionary in the `FdoIPropertyDictionary` class reference. Using these methods you can find out the names of the properties, whether they are optional or mandatory, whether they have default values, their type, whether they are enumerable and, if enumerable, what the set of enumerated values is.

## Get Values for the Connection Properties

The types of properties used in connections include:

- user names
- passwords
- file path names
- URLs
- symbolic host names
- service identifiers
- data store names

Each provider has its own unique set of connection properties. Read abou the connection properties in the chapter about the providers.

## Set the Connection Properties

Read about the methods used to access the information in the connection properties dictionary in the FdoIPropertyDictionary class reference.

## Open a Connection

Read about the Open method in the `FdoIConnection` class reference.

Confirm that the connection is open by calling the `GetConnectionState` method.

### Open a Pending Connection

Each of the RDBMS-based providers has a connection property, which is there to contain the name of a data store. The data store property is initially not required in order to make a connection. If the property is not set, then the resulting connection, if successful, is in a pending state.

As a result of the initial connection operation, the data store property is populated with the names of the data stores in the RDBMS, and its optionality attribute is changed to a value of required.

Use the EnumeratePropertyValues method on the `FdoIPropertyDictionary` object to read the set of data store names and set the data store property in the dictionary object to one of those names. Then call the Open method on the <strong>FdoIConnection</strong> object again. This will result in a fully open connection.

You can connect to an ArcSDE provider using a pending connection, you will find that there is only ever one name in the set of data store names returned as a result of the initial connection.

## Next Steps

### ArcSDE Provider

You use this provider to connect to an ArcSDE server, which is, in turn, connected to an Oracle or SQL Server data source.

The next step is to determine the number and composition of feature schema in the data store so that you can execute data maintenance (insert, update, and delete) and query (select) commands.

### MySQL Provider

The next steps are to create a data store and a user, and after that, create a feature schema and execute data maintenance and query commands.

### SDF, SHP and ODBC Providers

The next steps are to create a feature schema and execute data maintenance and query commands.

### WFS and WMS Providers

The next steps are to determine the composition of the feature schema and execute query commands.

## Capabilities API

You can use this API to determine what capabilities to expose or hide in the user interface depending on the current provider.

You access the Capabilities API by way of the connection object. The availability of this API is not dependent on the connection state; it is always available. This static definition represents the maximum set of capabilities available through the current provider under optimum conditions. For example, the capabilities API says that the Oracle provider supports the update command; however, the execution of an update command could throw an exception because the user has not been granted access to the table to be updated. In this example, the update capability is conditional on user privilege. The exception message provides the additional information you need to use the capability successfully.

Use the `FdoIConnection` object to obtain the capability objects whose class names are `FdoICommandCapabilities`, `FdoIConnectionCapabilities`, `FdoIExpressionCapabilities`, `FdoIFilterCapabilities`, `FdoIRasterCapabilities`, `FdoISchemaCapabilities`, and `FdoITopologyCapabilities`.

## Command Capabilities

Use the `FdoIConnection` object's `GetCommandCapabilities` method to get the list of commands and related attributes supported by the provider.

# User Management API

You can use this API to add a user to, or drop a user from, an Oracle, MySQL, or SQL Server provider. You can also use this API to grant privileges to users. To perform these operations, you must be connected to the provider as a user with administrative privileges, and the connection must be in the pending state. You perform these operations by calling methods on the FdoUserManager object. You create this object by calling the static `FdoUserManager::Create` method.

# User Management Capabilities

The user management capabilities are not reflected in the capabilities API. That is, they do not show up in the information returned by the `FdoIConnection` object's `GetCommandCapabilities` method. You can use the `FdoUserManager` object's `SupportsDatastoreGrants` and `SupportsWindowsAuthentication` to control the execution of some of the object's methods. For the rest of the object's methods you must test for the name of the provider to determine whether or not you can call the method.

# Data Store Management API

You can use this API to create a data store, destroy a data store, or list data stores. Three RDBMS providers (Oracle, MySQL, and SQL Server) and one file-based provider (SDF) support the data store commands. You can use the `FdoICommandCapabilities` object to verify this.

The procedure used for creating and destroying a data store in an RDBMS-based provider is slightly different than the procedure used for these operations in the SDF provider. The key difference is that the connection to the RDBMS-based provider is in the pending state, and the connection to the SDF provider is in the closed state.

### Oracle, MySQL, and SQL Server Providers

To create or destroy a data store, do the following:

1. Open a connection in the pending state to the provider as a user with administrative privileges.

2 Use the connection object to create the `FdoICreateDataStore` or `FdoIDestroyDataStore` command object

3 Use the command object to get the appropriate property dictionary.

4 Use the property dictionary first to determine what properties must be given values and then to set those properties values.

5 Call the command object's `Execute()` method.

6 To verify the result of these operations, use the connection object to create an `FdoIListDataStores` command. Calling the `Execute` method on this object returns a data store property reader containing a list of the data stores in the data source. By default this list includes all of the data stores in the data source. Optionally, you can exclude the data stores not created using the `FdoICreateDataStore` command.

The difference between data stores created using the `FdoICreateDataStore` command and ones created by non-FDO means (for example, from the SQL command-line) is that the former contain a set of FDO-specific tables whose names are prefixed with `F_` or `f_`, and which hold FDO metadata. You can view the list of tables created by the `FdoICreateDataStore` command using the following operations for each provider type:

- Oracle: `sql> select owner,table_name from dba_tables where owner = '<data store name>';`
- MySQL: `mysql> show tables from <database name>;`
- SQL Server: In the Enterprise Manager application window, under Console Root , click Microsoft SQL Servers, click SQL Server Group, click *SQL_Servername*, click Databases , click *database_name*, click Tables.

Depending on the provider, most of the following values are displayed:

- `f_associationdefinition`
- `f_attributedefinition`
- `f_attributedependencies`
- `f_classdefinition`
- `f_classtype`
- `f_dbopen`

- f_lockname
- f_options
- f_sad
- f_schemainfo
- f_schemaoptions
- f_sequence
- f_spatialcontext
- f_spatialcontextgeom
- f_spatialcontextgroup

### SDF Provider

To create or destroy a data store, do the following:

1. Use the FdoICreateDataStore command to create an SDF file and the FdoIDestroyDataStore command to delete it. These operations are performed with the connection in the closed state.
2. Verify these operations by looking in the file system. The SDF provider does not support the FdoIListDataStores command.

## Spatial Context API

A spatial context defines properties required for the storage and manipulation of geometric data. You use this API to manage the spatial context of a data store.

Use the connection object to create the desired spatial context command (FdoIActivateSpatialContext, FdoICreateSpatialContext, FdoIDestroySpatialContext, or FdoIGetSpatialContexts). The connection must be in the open state to execute these commands. All of the providers support the FdoIGetSpatialContexts command. The ArcSDE, MySQL, Oracle, and SQL Server providers support the FdoICreateSpatialContext and FdoIDestroySpatialContext commands. The SDF provider supports the FdoICreateSpatialContext command but not the FdoIDestroySpatialContext command.

# Glossary

## Tolerance

Used to determine when two points are close enough to be considered to be equivalent in certain spatial tests. In FDO, the tolerance value is used as a radius around a position rather than as a distance along an axis. In the majority of cases, the distance between the points is compared to the tolerance using the 'less than or equals to' predicate rather than the 'less than' predicate. If the X and Y tolerances are both 0.05, then point B (1.0, 1.0) is considered to be equivalent to point A (0.96, 0.96) and point C (1.04, 1.04), but is not considered to be equivalent to point D (0.96, 0.94) or point E (1.04, 1.06). The relationship is not commutative. That is, point A is not equivalent to point C. Repeated applications of the `isEquivalentTo` predicate to the same data always yields the same result.

# The Open Source Providers

# 2

This chapter describes how to connect to and configure the open source providers (ArcSDE, MySQL, ODBC, SDF, SHP, WFS, and WMS.

## In this chapter

# System Requirements

## Software

The following table maps the various providers to the operating system and RDBMS versions against which they have been tested (T). The RDBMS-based providers (ArcSDE and MySQL) require that the corresponding RDBMS client is installed. The ODBC provider requires that an ODBC driver be installed on Linux. Windows XP contains the required ODBC drivers.

**WXP=Windows XP Pro; W2K=Windows 2000 Pro; W2K3=Windows 2003 Server**

| Provider | WXP | W2K | W2K3 | Linux | RDBMS | Notes |
|---|---|---|---|---|---|---|
| ArcSDE | T | T | T | T | 9i R2 (9.2.0.6) Server - Standard Edition (32 bit) | 1 |
| MySQL | T | T | T | T | 5.0.x | 2 |
| ODBC | T | T | T | T | n/a | 3 |
| SDF | T | T | T | T | n/a | |
| SHP | T | T | T | T | n/a | |
| WFS | T | T | T | T | n/a | 4 |
| WMS | T | T | T | T | n/a | 5 |

**NOTE**

1. The ArcSDE 9.1 client and ArcSDE 9.1 server for Oracle and ArcSDE 9.1 server for SQL Server are required.
2. Testing the MyIsam database engine with spatial extension only.
3. The MDAC 2.8 ODBC driver manager is tested on Windows, and the unixODBC 2.2.11 driver manager is tested on Linux. On Windows you can use the provider to get read-only access to MS Access (MDB file type), Excel drivers (XLS file type), Oracle, MySQL, and SQL Server. On Linux read-only access to Oracle is tested.
4. WFS 1.0.0 and GML 2.1.2
5. WMS 1.1.0, WMS 1.1.1, WMS 1.3, and GML 2.1.2

# Disk Space and Memory

This table shows the disk space/memory requirements the RDBMS clients on an XP machine.

## Windows XP, SP2

This table shows the disk space/memory requirements for the RDBMS clients.

**D = Disk (MB); M = Memory (MB); N = Notes**

| Component | D | M | N |
|---|---|---|---|
| ArcSDE 9.1 client (3 DLLs) | 26.3 | 4 | 1 |
| MySQL 5.0 Community Edition (`mysqld-nt.exe`) | 94 | 13 | 2 |

**NOTE**

1 The ArcSDE client disk size is reflected in the total, but the memory size would be included in that of the running FDO application.

2 MySQL includes both server and client components. There is no separate client.

## Linux

This table shows the disk space/memory requirements for the RDBMS clients.

**D = Disk (MB); M = Memory (MB); N = Notes**

| Component | D | M | N |
|---|---|---|---|
| ArcSDE 9.1 client | 26.4 | 4 | 1 |
| MySQL 5.0 Community Edition (`mysqld`) | 49.6 | 1.25 | 2 |

**NOTE**

1 The ArcSDE client disk size is reflected in the total, but the memory size would be included in that of the running FDO application.

2 MySQL includes the following RPMs: MySQL-client-standard, MySQL-devel-standard, MySQL-server-standard, and MySQL-shared-standard.

# Connection API

The following table shows the connection properties for each provider. “T” means true, and “F” means false. For important details, refer to the associated Notes.

**DV=Default Value; R=Required; P=Protected; FN=FileName; FP=FilePath; D=Datastore; E=Enumerable**

| Provider/Properties | DV | R | P | FN | FP | D | E | Notes |
|---|---|---|---|---|---|---|---|---|
| `OSGeo.ArcSDE.3.0` | | | | | | | | |
| `Server` | | T | F | F | F | F | F | 1 |
| `Instance` | | T | F | F | F | F | F | 2 |
| `Username` | | T | F | F | F | F | F | 3,4 |
| `Password` | | T | T | F | F | F | F | 3,4 |
| `Datastore` | | F | F | F | F | T | T | 5,6 |
| `OSGeo.MySQL.3.0` | | | | | | | | 7 |
| `Username` | | T | F | F | F | F | F | 8 |
| `Password` | | T | T | F | F | F | F | 8 |
| `Service` | | T | F | F | F | F | F | 9 |
| `DataStore` | | F | F | F | F | T | T | 10 |
| `OSGeo.ODBC.3.0` | | | | | | | | 11 |
| `DataSourceName` | | F | F | F | F | F | T | 12 |
| `UserId` | | F | F | F | F | F | F | 13 |

**DV=Default Value; R=Required; P=Protected; FN=FileName; FP=FilePath; D=Datastore; E=Enumerable**

| Provider/Properties | DV | R | P | FN | FP | D | E | Notes |
|---|---|---|---|---|---|---|---|---|
| Password | | F | T | F | F | F | F | 13 |
| ConnectionString | | F | F | F | F | F | F | |
| GenerateDefaultGeometryProperty | True | F | F | F | F | F | F | |
| OSGeo.SDF.3.0 | | | | | | | | 19 |
| File | | T | F | T | F | F | F | 18 |
| ReadOnly | False | F | F | F | F | F | T | 20 |
| OSGeo.SHP.3.0 | | | | | | | | 21 |
| DefaultFileLocation | | F | F | T | F | F | F | 22 |
| TemporaryFileLocation | | F | F | T | F | F | F | 23 |
| OSGeo.WFS.3.0 | | | | | | | | |
| FeatureServer | | T | F | F | F | F | F | |
| Username | | F | F | F | F | F | F | |
| Password | | F | F | F | F | F | F | |
| OSGeo.WMS.3.0 | | | | | | | | |
| FeatureServer | | T | F | F | F | F | F | |
| Username | | F | F | F | F | F | F | |

**DV=Default Value; R=Required; P=Protected; FN=FileName; FP=FilePath; D=Datastore; E=Enumerable**

| Provider/Properties | DV | R | P | FN | FP | D | E | Notes |
|---|---|---|---|---|---|---|---|---|
| `Password` |  | F | F | F | F | F | F |  |
| `DefaultImageHeight` |  | F | F | F | F | F | F |  |

**NOTE**

1. The value of the Server property is the name or IP address of the host where the ArcSDE 9.1 Server instance runs.
2. The value of the Instance property defines the port and protocol to use when communicating with the ArcSDE server. This instance name must be mapped to the TCP/IP listen port used by the Server to accept connection requests. This mapping is stored in the `C:\WINDOWS\system32\drivers\etc\services` file and has the form "`instance_name portnum/`*protocol* `# comment`". For example, "`esri_sde_oracle 5151/tcp #ArcSDE Server listening port`". There can be multiple instances because the server can connect to multiple back-end database technologies, for example, both Oracle and SQL Server.
3. You must create the username and password from the `sqlplus` command-line when FDO Providerfor ArcSDE is using Oracle as the back-end database technology.
   1. `sql> connect sys/password as sysdba`
   2. `sql> create user username identified by password;`
   3. `sql> grant connect, resource, select any table, insert any table, update any table, delete any table to username;`
4. You must create the username and password from the Enterprise Manager when FDO Provider for ArcSDE is using SQL Server as the back-end database technology. The following procedure adds a new login user named *name* with password *password* who must login to the SQL Server using SQL Server authentication. The user named *name* is given the global privilege, Database Creators, and is added to a database with privileges, db_ddladmin, db_datareader, and db_datawriter.
   1. On the Windows Start menu, click Programs, click Microsoft SQL Server, click Enterprise Manager.
   2. In the Enterprise Manager application window, click Console Root, click Microsoft SQL Servers, click SQL Server Group, click *GroupName*, click Security, click Logins, click New Login. The SQL Server Login Properties - New Login dialog box is displayed.
   3. In the SQL Server Login Properties - New Login dialog box, click the tab labeled General, type *name* in the text entry box labeled Name, click the radio button labeled SQL Server Authentication, type a *password* in the text entry box labeled Password.

    4. Click the tab labeled Server Roles, click the check-box labeledDatabase Creators
    5. Click the tab labeled Database Access, click the check-box in the column labeled Permit next to the name of the database to which you are adding this user, click the check-boxes in the column labeled Permit in Database Role next to the labels db_ddl_admin, db_datareader, and db_datawriter.
    6. Click OK. The Confirm Password dialog box is displayed.
    7. In the Confirm Password dialog box, type *password* in the text entry box labeled Password, click OK.
    8. In the SQL Server Login Properties - New Login dialog box, click OK.

5. When FDO Provider for ArcSDE is using Oracle as the back-end database, the value of DataStore is always 'Default Datastore'.
6. When FDO Provider for ArcSDE is using SQL Server as the back-end database, the value of DataStore is one of the values in the list of databases supported by that instance of the SQL Server.
7. During the Windows installation of MySQL, be sure to specify a password for the root account.
8. SeeUser Management API (page 15)for information about adding a username and password.
9. The value of the Service property is `localhost` for a local connection and a valid domain name for a remote connection. A firewall at the remote end can prevent a connection from succeeding. However, note that the exception message will not tell you that this is the cause.
10. The value of the DataStore property is one of the values enumerated in the connection property dictionary after opening a partial connection to the provider. For more information about how this is done, see the paragraphs that introduce this table.
11. Initially, you use this provider to access feature source files created using either MS Access or Excel.

**12** Use a Windows OS administrative tool to define a DSN (data source name). This action associates a symbolic name with a file path. Do the following to add a User DSN for a Microsoft Access database file.

   **1** On the Windows Start menu, click Settings, click Control Panel, click Administrative Tools, click Data Sources (ODBC). The ODBC Data Source Administrator dialog box is displayed.

   **2** In the Data Source Administrator dialog box, click User DNS , click Add. The Create New Data Source dialog box is displayed.

   **3** In the Create New Data Source dialog box, click Microsoft Access Driver (*.mdb), click Finish. The ODBC Microsoft Access Setup dialog box is displayed.

   **4** In the ODBC Microsoft Access Setup dialog box, in the text entry box labeled Data Source Name , type the *name* of the data source, click Select. The Select Database dialog box is displayed.

   **5** In the Select Database dialog box, select *directory*, select *file*, click OK.

   **6** In the ODBC Microsoft Access Setup dialog box, click OK.

   **7** In the Data Source Administrator dialog box, click OK.

**13** Optional properties, depending on the data source.

**14** This is a file name. The named file is in the local file system.

**15** The SDF provider supports the `.sdf` file type.

**16** If True, only read operations on the file are permitted.

**17** The SHP provider supports the `.shp` file type and the `.dbf` file type.

**18** The DefaultFileLocation can be either a directory in the file system or the name of an existing file.

**19** The TemporaryFileLocation is a folder where the SHP provider creates files that are normally co-located with the `.shp` file named in the DefaultFileLocation property, but are, for some reason, missing. These files have file types `.shx`, `.sbn`, `.idx`, `.dbf`, and `.sbx`.

# Command Capabilities

The list of command class names is shown in the first table. For descriptions of what the commands do, see the API reference documentation. The command attributes are shown in the second table. For descriptions of what the attributes mean, see entry for the `FdoICommandCapabilities` class in the API reference documentation. “Y” means the command or attribute is supported in that particular provider.

| Command Class Name | ArcSDE | MySQL | ODBC | SDF | SHP | WFS | WMS |
|---|---|---|---|---|---|---|---|
| `FdoISelect` | Y | | Y | Y | Y | Y | Y |
| `FdoISelectAggregates` | Y | Y | Y | Y | Y | | Y |
| `FdoIInsert` | Y | Y | Y | Y | Y | | |
| `FdoIDelete` | Y | Y | Y | Y | Y | | |
| `FdoIUpdate` | Y | Y | Y | Y | Y | | |
| `FdoIDescribeSchema` | Y | Y | Y | Y | Y | Y | Y |
| `FdoIDescribeSchemaMapping` | | | Y | | Y | | Y |
| `FdoIApplySchema` | | Y | | Y | Y | | |
| `FdoIDestroySchema` | | Y | | | Y | | |
| `FdoIActivateSpatialContext` | Y | Y | | | | | Y |
| `FdoICreateSpatialContext` | Y | Y | | Y | | | |
| `FdoIDestroySpatialContext` | Y | Y | | | | | |
| `FdoIGetSpatialContexts` | Y | Y | Y | Y | Y | Y | Y |

| Command Class Name | ArcSDE | MySQL | ODBC | SDF | SHP | WFS | WMS |
|---|---|---|---|---|---|---|---|
| `FdoICreateDataStore` |  | Y |  | Y |  |  |  |
| `FdoIDestroyDataStore` |  | Y |  | Y |  |  |  |
| `FdoIListDataStores` |  | Y |  |  |  |  |  |
| `FdoICreateSDFFile` |  |  |  | Y |  |  |  |
| `FdoISQLCommand` | Y | Y | Y |  |  |  |  |
| `FdoIAcquireLock` | Y |  |  |  |  |  |  |
| `FdoIGetLockInfo` | Y |  |  |  |  |  |  |
| `FdoIGetLockedObjects` | Y |  |  |  |  |  |  |
| `FdoIGetLockOwners` | Y |  |  |  |  |  |  |
| `FdoIReleaseLock` | Y |  |  |  |  |  |  |
| `FdoIActivateLongTransaction` | Y |  |  |  |  |  |  |
| `FdoICommitLongTransaction` | Y |  |  |  |  |  |  |
| `FdoICreateLongTransaction` | Y |  |  |  |  |  |  |
| `FdoIDeactivateLongTransaction` | Y |  |  |  |  |  |  |
| `FdoIFreezeLongTransaction` |  |  |  |  |  |  |  |
| `FdoIGetLongTransactions` | Y |  |  |  |  |  |  |
| `FdoIRollbackLongTransaction` | Y |  |  |  |  |  |  |

| Command Class Name | ArcSDE | MySQL | ODBC | SDF | SHP | WFS | WMS |
|---|---|---|---|---|---|---|---|
| FdoICreateSpatialIndex |  | Y |  |  |  |  |  |
| FdoIDestroySpatialIndex |  | Y |  |  |  |  |  |
| FdoIGetSpatialIndexes |  | Y |  |  |  |  |  |

**NOTE** The capability to create and destroy a data store is associated with the capability to create and destroy a feature schema. To create a data store you specify values for data store properties. For more information about these properties are discussed in Data Store Management API (page 15).

| Attribute | ArcSDE | MySQL | ODBC | SDF | SHP | WFS | WMS |
|---|---|---|---|---|---|---|---|
| FdoIDeactivateLongTransaction | Y |  |  |  |  |  |  |
| FdoIFreezeLongTransaction |  |  |  |  |  |  |  |
| FdoIGetLongTransactions |  | Y | Y | Y | Y |  |  |
| FdoIRollbackLongTransaction | Y | Y | Y | Y | Y |  |  |
| FdoICreateSpatialIndex | Y | Y | Y | Y | Y |  |  |
| FdoIDestroySpatialIndex | Y | Y | Y |  | Y |  |  |
| FdoIGetSpatialIndexes |  | Y | Y |  |  |  |  |

**NOTE**

1 The capability to create and destroy a data store is associated with the capability to create and destroy a feature schema. To create a data store you specify values for data store properties. For more information about these properties are discussed in Data Store Management API (page 15).

# User Management API

In the open source domain there is no API for this functional domain. Instead it is a set of SQL commands that you must do to add a user to MySQL.

# SQL Commands

The following table maps a set of. The Operation column contains the name of methods in class `FdoUserManager`.

If the method returns a boolean, the table cell contains the boolean value returned by the provider. If the method performs an oiperation and the provider supports the operation, the table cell contains the SQL statements executed by the provider.

| Operation | MySQL | Notes |
|---|---|---|
| `AddUser(name, password)` | `create user ‘`*name*`’ identified by` *password* | 1 |
| `AssignRoleOrPrivilege(user, roleOrPriv)` | `grant` *roleOrPriv*`on *.* to` *user* | 2 |
| `DropUser(name)` | `drop user` *name* | |
| `GetDatastores()` | `FdoIListDatastores` | 3 |
| `GetDbUsers()` | `select user as name from mysql.user` | 4 |
| `GetDbUsers(inDatastore)` | `select distinct grantee as name from information_schema.schema_privilges where table_schema =` *inDatastore* | 5 |
| `GetRolesOrPrivileges()` | `select distinct privilege_type from information_schema.user_privileges where grantee like “’root’%” order by privilege_type` | 6 |
| `GrantAccessToDatastore(user, datastore)` | `grant all on` *datastore*`.* to` *user* | 7 |

| Operation | MySQL | Notes |
|---|---|---|
| `GrantAccessToDatastore(user, datastore, priv)` | `grant` *priv* `on` *datastore*`.*` `to` *user* | 8 |
| `RevokeAccessToDatastore( user, datastore, priv)` | `revoke` *priv* `on` *datastore*`.*` `from` *user* | 9 |
| `RevokeRoleOrPrivilege(user, roleOrPriv)` | `revoke` *roleOrPriv* `on` `*.*` `from` *user*. | 10 |

**NOTE**

1 A new MySQL user is created with no privileges. Assume that you have created a new user called newuser on the MySQL instance on your machine. Log in by executing `C:\ >mysql -u newuser -p`. Then execute `mysql> show grants for current_user;`. The result string is `"GRANT USAGE ON *.* TO 'newuser'@'%' IDENTIFIED BY PASSWORD '*94BDCEBE19083CE2A1F959FD02F964C7AF4CFC29'"`. This means that newuser has no privileges and can only log in locally. Call `FdoUserManager::GrantAccessToDatastore()`with user name and data store arguments to give the user default privileges.

2 To confirm the results of this method call, call `GetDBUsers()`.

3 You can get the same information from the connection properties dictionary for the DataStore property. The MySQL SQL command `mysql> show databases;` lists the database (data store) names.

4 If you are connected to MySQL and you created the user with AddUser, USAGE is shown as the privilege.

5 All users in a particular data store are listed along with the privileges that they have in the data store. If connected to MySQL and you used the two-argument version of `GrantAccessToDatastore`, the privileges are listed in the note for that method.

6 The list of supported privileges for MySQL consists of `ALTER`, `ALTER ROUTINE`, `CREATE`, `CREATE ROUTINE`, `CREATE TEMPORARY TABLES`, `CREATE USER`, `CREATE VIEW`, `DELETE`, `DROP`, `EXECUTE`, `FILE`, `INDEX`, `INSERT`, `LOCK TABLES`, `PROCESS`, `REFERENCES`, `RELOAD`, `REPLICATION CLIENT`, `REPLICATION SLAVE`, `SELECT`, `SHOW DATABASES`, `SHOW VIEW`, `SHUTDOWN`, `SUPER`, and `UPDATE`.

7 This overloaded method takes two arguments: a user name and a data store name. It grants a set of default privileges to the user when connected to that data store. For MySQL the privileges are alter, alter routine, create, create routine, create temporary tables, crew view, delete, drop, execute, index, insert, lock tables, references, select, show view, and update. Call `GetDBUsers(<datastore>)` to verify the result.

8 This overloaded method takes three arguments: a user name, a data store name, and a privilege name. It grants the specified privilege to that user for the named data store. The MySQL provider does not allow you to grant any privilege other than those assigned by a call to `GrantAccessToDatastore(<user>, <datastore>)`. The SQL Server provider does allow you to grant any of the supported privileges with the exception of the `public`privilege. Call `GetDBUsers(<datastore>)` to verify the result.

9 This method takes three arguments: a user name, a data store name, and a privilege name. It revokes the specified privilege from that user for the named data store. If the role is set to '*', the provider revokes all privileges in the data store from the user. In this case the SQL Server provider also drops the database user from the data store; the user's login status is not changed.

10 This method takes two arguments: a user name and a role or privilege name. It revokes the specified role or privilege.

## Data Store Management API

This table shows the data store management properties for each provider. "T" means true, and "F" means false.

**DV=DefaultValue; R=Required; P=Protected; FN=FileName; D=Datastore; E=Enumerable**

| Provider/Properties | DV | R | P | FN | D | E | Notes |
|---|---|---|---|---|---|---|---|
| OSGeo.MySQL.3.0 | | | | | | | |
| DataStore | | T | F | F | T | F | 1 |
| Description | | F | F | F | F | F | |

**NOTE**

1 A data store is a container in the data source to which you can apply a feature schema.

## Spatial Context API

The following table defines default values for the spatial context properties for the providers (MySQL and SDF) that support the creation of a data store and example sets of spatial context properties extracted from ODBC, WFS, and WMS sources created by a non-FDO process.

| Properties/Provider | MySQL | ODBC | SDF | WFS | WMS | Notes |
|---|---|---|---|---|---|---|
| Coordinate system | Cartesian | Cartesian | Cartesian | EPSG:4326 | EPSG:4326 | 1 |
| Extent type | Static | Static | Dynamic | Static | Static | 2 |
| Extent | Not NULL | Not NULL | NULL | Not NULL | Not NULL | 3 |
| Unit dimension | XY | XY | Un-defined | XY | XY | 4 |
| XY tolerance | .001 | .001 | 0 | .01 | .001 | 5 |
| Z tolerance | .001 | .001 | 0 | .01 | .001 | 5 |
| Active | False | F | True | F | F | 6 |

**NOTE**

1. EPSG:4326 is a Layer Coordinate Reference System (CRS) as defined in the OpenGIS Web Map Service (WMS) Implementation Specification, OGC 04-024, version 1.3, dated August 2, 2004.
2. A static extent type means that the extent is defined at the time of the creation of the spatial context using the `FdoICreateSpatialContext` command. A dynamic extent type means that the extent may change as features are added so that it always encloses all of the features.
3. The default extents expressed in AGF text format for the various providers are as follows:
    - MySQL, : POLYGON ((-2000000 -2000000, 2000000 -2000000, 2000000 2000000, -2000000 2000000, -2000000 -2000000))
4. The dimensionality is that of the extent. It is not set directly.
5. See Tolerance (page 18).
6. ArcSDE and MySQL providers can have multiple spatial contexts defined and can, therefore, show contexts in the reader that are not active.

# Exception Messages

## Connection Not Established

This exception message occurs when there is a closed connection to MySQL and a call is made to EnumeratePropertyValues on the connection properties dictionary object for the DataStore property.

## Incorrect usage of DB GRANT and GLOBAL PRIVILEGES

This exception message occurs when there is a pending connection to MySQL and a call is made to `GrantAccessToDatastore(<user>, <datastore>, <priv>)` on the user manage object where the privilege is not one of the defaults assigned by the call to `GrantAccessToDatastore(<user>, <datastore>)`. The privileges causing this exception are create user, file, process, reload, replication client, replication slave, show databases, shutdown, and super. This set is the difference between the list of privileges returned by `GetRolesOrPrivileges` and the defaults assigned by `GrantAccessToDatastore(<user>, <datastore>)`.

# 3

# Installing RDBMS Clients Used By The Open Source Providers

## In this chapter

# Introduction

The operation of the ArcSDE provider requires the installation of an ArcSDE client on the same machine. This chapter describes the installation of an ArcSDE 9.1 client on Windows XP and Linux x86.

The operation of the MySQL provider requires the installation of an MySQL client on the same machine. This chapter describes the installation of a MySQL 5.0 database (server and client) on Windows XP and Linux x86.

The operation of the ODBC provider on Linux x86 requires the installation of an ODBC driver manager as well as ODBC provider-specific drivers. No installation is required on XP because XP is installed with the requisite software. This chapter describes the installation of the unixODBC driver manager and the Easysoft ODBC Oracle driver.

# Installing and Configuring an ArcSDE Client on Windows

## Installation

You must purchase the ArcSDE 9.1 client SDK from an ESRI vendor. There are instructions on how to purchase the client at *http://www.esri.com/software/arcgis/arcsde/how-to-buy.html*.

Follow the instructions that come with the SDK to install the client to a folder outside of the FDO distribution. Make sure that the folder containing the following DLLs is visible in the %PATH% environment variable .

- pe91.dll
- sde91.dll
- sg91.dll

## Configuration

Before you can complete the configuration of the client on the local machine, you must install and configure the ArcSDE server to which the client will connect. An ArcSDE client does not have to be the same version as the ArcSDE server. Connecting an ArcSDE 9.0 or 9.1 client to an ArcSDE 8.3 server has been tested. Not all possible configurations are known.

It is possible for an ArcSDE client to connect directly to an Oracle or SQL Server database provided the required drivers are installed. This type of 'direct' connection has not been tested with the ArcSDE Provider, and so is not supported.

An ArcSDE server connects to only one kind of RDBMS. The ArcSDE Server for Oracle can connect to an Oracle database and the ArcSDE Server for SQL Server can connect to a Sql Server database. There are post-installations actions that you must take to create ArcSDE-specific tables in the target RDBMS before you can connect to the "ArcSDE-aware" database.

When you install an ArcSDE server, you configure it to listen on a port for database connect requests. Each ArcSDE server listens on its own dedicated listener port.

The ArcSDE client needs to know the number of the server's listener port and the protocol to use for sending requests to that port. The listener port and protocol is what the ArcSDE client needs to get from the ArcSDE provider in order to connect to the ArcSDE server. The ArcSDE provider API requires that the user put this information in the Instance connection property. The user can provide the information either directly or indirectly.

The direct way is simply the listener port and protocol. The syntax is "<port>/<protocol>", for example, "5151/tcp".

The indirect way is a symbolic name, for example, "esri_sde_oracle." The ArcSDE provider translates the symbolic name into a port and protocol by looking the symbolic name up in the C:\WINDOWS\System32\drivers\etc\services file. An example of an entry in the services file is `esri_sde_oracle 5151/tcp`.

You must also create usernames and passwords in each of the databases that you plan to connect to. You or your database administrator must connect to the database using an account with sufficient privileges to create a new user account and to grant the appropriate privileges to the new user.

# Installing and Configuring an ArcSDE Client on Linux

## Installation

You must purchase the ArcSDE 9.1 client SDK from an ESRI vendor. There are instructions on how to purchase the client at *http://www.esri.com/software/arcgis/arcsde/how-to-buy.html*.

Follow the instructions that come with the SDK to install the client to a directory outside of the Fdo distribution. Make sure that the folder containing the following shared objects is visible in the LD_LIBRARY_PATH environment variable .

- libpe91.so
- libsde91.so
- libsg91.so

## Configuration

Configuration on Linux is the same as configuration on Windows except that the location of the services file is /etc. You need root privileges to modify this file.

# Installing and Configuring the MySQL Client on Windows

## Installation

Install the MySQL database, which includes the client as well as the server.

Download a Windows (x86) installer of the latest production release of MySQL 5.0 from *http://dev.mysql.com/downloads/mysql/5.0.html*. Do a ‘typical’ install to the default location (C:\Program Files\MySQL\MySQL Server 5.0).

The install wizard does not configure MySQL. At the conclusion of the installation procedure, you are given the option of continuing with configuration or leaving that step to later. If you decide to continue, you simply click the Finish button, which triggers the launch of the configuration wizard. If you decide to configur later, you invoke the MySQL Server Instance Config Wizrd by following the menu path Start > Programs > MySQL > MySQL Server 5.0.

## Configuration

The configuration wizard presents you with the following configuration defaults.

1. Detailed Configuration
2. Multifunctional Database
3. default InnoDB Tablespace Setting
4. Decision Support (DSS)/OLAPA
5. Enable TCP/IP Networking (using Port Number 3306)
6. Enable Strict Mode
7. Standard Character Set
8. Install As Windows Service (Service Name = MySQL with the service started automatically)
9. Modify Security Settings (you must provide a root password (twice) and if you deinstalled an existing MySQL database, you must provide a current password)

You may optionally request that the bin directory be added the the Windows PATH environment variable. Do so. The wizard will add that directory to the end of %PATH%. The bin folder contains the libmysql.dll used by the MySQL provider.

**NOTE** If you have PHP installed on your machine, make sure that C:\Program Files\MySQL\MySQL Server 5.0\bin precedes the PHP path in the PATH environment variable. The PHP installation contains a copy of the libmysql.dll and it is important for the operation of the MySQL provider that the correct DLL is used.

The configuration wizard creates a configuration file (C:\Program Files\MySQL\MySQL Server 5.0\my.ini). It starts the MySQL service (mysqld) using options specified in my.ini.

You can use the Fdo User Manager API to create a user account with the usage privilege if you are using the enterprise version of FDO. Otherwise you can use the 2 SQL commands, `create user <user> identified by '<password>'` and `grant select, insert, update, delete, create, create view, drop, index, alter on *.* to '<user>'@'%' identified by '<password>'`. In either case you must be connected as a user, for example, root, with the grant privilege. The value <user> can be '<username>'@'localhost' if the connection local, or it must be '<username>'@',hostname.' if the connection is remote. The value of <hostname> is <host label>.<domain name>, for example, *someComputer.someCompany.com*.

**NOTE** If you enabled remote root access when you installed MySQL, you can create user accounts by way of a remote connection. Otherwise, you will have to connect locally in order to create a user account.

You can use the Fdo Data Store Management API to create a data store.

# Installing and Configuring the MySQL on Linux

## Installation

Install the MySQL database, which includes the client as well as the server.

Download a Linux x86 non RPM package Standard version (gzipped tarfile) containing the latest production release of MySQL 5.0 from *http://dev.mysql.com/downloads/mysql/5.0.html*. At the time of the writing of this document the latest version is 5.0.22 compiled with glibc-2.2.

Follow the installation instructions in the reference manual accessible at this URL *http://dev.mysql.com/doc/refman/5.0/en/index.html*. Click on 2. Installing and Upgrading MySQL. Click on 2.8 Installing MySQL on Other Unix-Like Systems.

**NOTE** The sequence of shell commands specified at the beginning of this section are executed with root privileges.

# Configuration

Configuration consists of setting up the mysql server as a service and of setting passwords for the 2 root and 2 anonymous accounts.

Return to the page containing the table of contents for Chapter 2. Click on 2. Installing and Upgrading MySQL. Click on 2.10.2 Unix Post-Installation Procedures. Scroll down to the paragraph whose first sentence is If you install MySQL from a source distribution or using a binary distribution format that does not install mysql.server automatically, you can install it manually.  and read to the line containing the shell command `shell> chkconfig --add mysql`.

**NOTE** `chkconfig` uses a commented line from the /etc/init.d/mysqlfile, namely,

```
# chkconfig: 2345 64 36
```

Verify that chkconfig has written the correct entries to the /etc/rc*.d directories.

- /etc/rc0.d, /etc/rc1.d, and /etc/rc6.d should contain K36mysql -> ../init.d/mysql
- /etc/rc2.d, /etc/rc3.d, /etc/rc4.d, and /etc/rc5.d should contain S64mysql -> ../init.d/mysql

Return to the page containing the table of contents for Chapter 2. Click on 2. Installing and Upgrading MySQL. Click on 2.10.3 Securing the Initial MySQL Accounts. Follow the instructions for securing the 2 root and 2 anonymous accounts.

**NOTE** If you get an error message like ERROR 2003 (HY00): Can't connect to MySQL server on '<hostname>' (<somenumber>) when setting the password for the ''@'<hostname>'' or the 'root'@'<hostname>' account, verify that the IP address that <hostname> maps to in the /etc/hosts file is correct.

You can use the Fdo User Manager API to create a user account with the usage privilege if you are using the enterprise version of FDO. Otherwise you can use the 2 SQL commands, `create user <user> identified by '<password>'` and `grant select, insert, update, delete, create, create view, drop, index, alter on *.* to '<user>'@'%' identified by '<password>'`. In either case you must be connected as a user, for example, root, with the grant privilege. The value <user> can be '<username>'@'localhost' if the connection local, or it must be '<username>'@',hostname.' if the connection is remote. The value of <hostname> is <host label>.<domain name>, for example, *someComputer.someCompany.com*.

**NOTE** If you enabled remote root access when you installed MySQL, you can create user accounts by way of a remote connection. Otherwise, you will have to connect locally in order to create a user account.

You can use the FDO Data Store Management API to create a data store.

# Installing and Configuring ODBC on Linux

First install the unixODBC driver manager and then install the drivers. Initially only the Easysoft ODBC Oracle driver is supported.

## Installation of the Driver Manager

Download the latest version of the driver manager from *http://www.unixODBC.org*. It is packaged as a gzipped tarfile. The file used for this exercise is unixODBC-2.2.11.tar.gz. It gunzips to a directory called unixODBC-2.2.11. Change directories to unixODBC-2.2.11, untar the file and run the following commands.

1. `./configure --prefix=/usr/local/unixODBC`
2. `make`
3. `make install` as root

# Index