

Feature Data Objects (FDO)

# The Essential FDO

March 2006

Copyright© 2006 Autodesk, Inc.

#### All Rights Reserved

This publication, or parts thereof, may not be reproduced in any form, by any method, for any purpose.

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 limitation the 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

# Contents

<b>Chapter 1</b>	<b>The Essential FDO</b>	<b>1</b>
	What Is the Feature Data Objects API?	3
	Objectives	3
	Related Documentation	4
	Provider Overview	5
	System Requirements	6
	Software	6
	Disk Space and Memory	8
	Windows XP, SP2	8
	Installing the FDO SDK	11
	Enterprise Edition	11
	Windows XP	11
	Community Edition	12
	Directory Structure	13
	Enterprise Edition	13
	Windows	13
	Programming Environment	16
	Microsoft Visual Studio 2005	16
	Compile-Line Options	17
	Link-Line Options	17
	Application Tasks	18
	C++ Memory Management	18
	Collections	19

Provider Connection Targets . . . . .	19
Provider Registry API . . . . .	20
Connection API . . . . .	20
Capabilities API . . . . .	28
Command Capabilities . . . . .	28
User Management API . . . . .	32
User Management Capabilities/SQL Commands . . . . .	32
Data Store Management API . . . . .	38
Data Store Management SQL Commands . . . . .	43
FdoICreateDataStore . . . . .	43
FdoIDestroyDataStore . . . . .	44
FdoIListDataStores . . . . .	45
Spatial Context API . . . . .	45
Known Problems . . . . .	49
MySQL Provider . . . . .	49
Exception Messages . . . . .	49
Connection Not Established . . . . .	49
Incorrect usage of DB GRANT and GLOBAL PRIVILEGES . . . . .	49
Membership of the public role cannot be changed . . . . .	50
Glossary . . . . .	50
Tolerance . . . . .	50
<b>Index . . . . .</b>	<b>51</b>

# The Essential FDO

You can use this document 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](mailto:isd.docs@autodesk.com).

## In this chapter

- [What Is the Feature Data Objects API?](#)
- [Objectives](#)
- [Related Documentation](#)
- [Provider Overview](#)
- [System Requirements](#)
- [Installing the FDO SDK](#)
- [Directory Structure](#)
- [Programming Environment](#)
- [Application Tasks](#)
- [C++ Memory Management](#)
- [Collections](#)
- [Provider Connection Targets](#)
- [Provider Registry API](#)
- [Connection API](#)
- [Capabilities API](#)
- [User Management API](#)
- [Data Store Management API](#)
- [Spatial Context API](#)
- [Known Problems](#)
- [Exception Messages](#)
- [Glossary](#)



# 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. It also provides a model for extending this interface to additional data source technologies. A provider is a specific implementation of the FDO API that provides access to data stored in a particular data source technology. For example, the Autodesk FDO Provider for Oracle provides access to GIS data stored in an Oracle database.

Some data source technologies are file-based, such as SHP and SDE. Others, such as Oracle and MySQL, use relational databases. FDO has a capabilities API for determining what operations are supported by each provider of a particular technology.

Each provider stores feature data. The storage 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.

## Objectives

The main objective of this document is to provide information to help you install the FDO SDK 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 28).

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:

- *FDO SDK ReadMe* (FDO\_SDK\_ReadMe.chm)
- *FDO Developer's Guide* (FDG\_FDODevGuide.pdf)
- *FDO API Reference* (FDO\_API.chm)
- *FDO API Reference - Managed* (FDO\_API\_managed.chm)
- *FDO Provider for ArcSDE* (ArcSDE\_Provider\_API.chm)
- *FDO Provider for MySQL* (MySQL\_Provider\_API.chm)
- *FDO Provider for ODBC* (ODBC\_Provider\_API.chm)
- *FDO Provider for Oracle* (Oracle\_Provider\_API.chm)
- *FDO Provider for Raster* (Raster\_Provider\_API.chm)
- *FDO Provider for SDF* (SDF\_Provider\_API.chm)
- *FDO Provider for SHP* (SHP\_Provider\_API.chm)
- *FDO Provider for SQL Server* (SQLServer\_Provider\_API.chm)
- *FDO Provider for WMS* (WMS\_Provider\_API.chm)

---

**NOTE** There is no CHM file 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

### Software

The following table maps the various providers to the operating system and RDBMS versions against which they have been tested (T). With the exception of the SQL Server provider, the RDBMS-based providers require that the corresponding RDBMS client is installed. Connection to a remote SQL Server

server is done through the standard ODBC component that is installed with the Windows operating system.

Provider	Windows XP Pro	Windows 2000 Pro	Windows 2003 Server	RDBMS	Notes
Oracle	T	T	T	10g R1 Server - Standard Edition (32-bit or 64-bit)	
ArcSDE	T	T	T	9i R2 (9.2.0.6) Server - Standard Edition (32 bit)	1
ArcSDE	T	T	T	SQL Server 2000 SP 4	1
SQL Server	T	T	T	SQL Server 2000 SP 4	
SQL Server	T	T	T	SQL Server 2005	
MySQL	T	T	T	5.0.x	2
SDF	T	T	T	n/a	
SHP	T	T	T	n/a	
Raster	T	T	T	n/a	3
ODBC	T	T	T	n/a	4
WFS	T	T	T	n/a	5
WMS	T	T	T	n/a	6

---

**NOTE**

- 1 The ArcSDE 9.1 client and ArcSDE 9.1 database format are required.
  - 2 Testing the MyIsam database engine with spatial extension only.
  - 3 The following raster formats are tested: TIFF, MrSID, PNG, JPG, JPG2K, DEM, DTED, ECW, ESRI ascii grid, and ESRI binary grid.
  - 4 Testing is against the MDAC 2.8 ODBC driver manager on Windows. Initially only the MS Access (MDB file type) and Excel drivers (XLS file type) are supported.
  - 5 WFS 1.0.0 and GML 2.1.2
  - 6 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 for an example client machine. The Oracle instance on this machine could be replaced by the Oracle instant client. The SQL Server server runs on a Windows 2000 or 2003 server; its disk and memory footprints are 223 MB and 4.5 MB (`sqlmangr.exe`). The ArcSDE server disk and memory footprints are 74 MB and 21 MB (`giomgr.exe` (17.4 MB) and `ARCGIS.exe` (3.6 MB)).

### Windows XP, SP2

The minimum configuration includes the FDO SDK, Visual Studio, and .NET.

Component	Disk (MB)	Memory (MB)	Notes
FDO SDK (inc, lib, docs, and sample directories)	242	n/a	1
FDO DLLs	33	n/a	2

Component	Disk (MB)	Memory (MB)	Notes
A small example program (FDO Command-line Interpreter sample)	55	7-27	3
Microsoft Visual Studio 8, Microsoft .NET (devenv.exe)	1,779	85	4
Oracle 10g R1 instance (Db_1 directory) (oracle.exe, tnslnsr.exe, isqlplussvc.exe)	932	232	5
Oracle 10g R1 Instant Client	94	19	6
MySQL 5.0 Community Edition (mysqld-nt.exe)	94	13	7
SQL Server Express This is the SQL Server service (sqlservr.exe).	257	26	8
ArcSDE 9.1 client (3 DLLs)	4	4	9

---

**NOTE**

- 1 The disk space size is that of the `C:\Map ObjectARX SDK\FDO\` folder.
  - 2 The disk space size is that of the `C:\Program Files\Autodesk Map 3D\FDO\bin` folder.
  - 3 The program exercises connection, data store management, user management and spatial context configuration. It is very limited in scope. The lowest memory size range represents initial loading and the highest represents connecting to an Oracle provider. The FDO Command-line Interpreter (FdoCI) is located in the `\sample\APIDocSandbox\FdoCI\` folder.
  - 4 The disk and memory size shown are for Visual Studio and .NET framework combined.
  - 5 The disk size includes the Oracle `Db_1` directory only; it does not include the default database or administration directories. The memory size is an approximate sum of the memory sizes for an Oracle instance running as a service. The processes are `oracle.exe`, `tnslsnr.exe`, and `isqlplussvc.exe`. The Oracle instance includes both client and server components. There is also a separate client application.
  - 6 When the FDO application connects to Oracle, it loads the Oracle dynamically linked libraries (DLLs)
  - 7 MySQL includes both server and client components. There is no separate client.
  - 8 The SQL Server Express application runs on a Windows XP machine. It includes both a client and a server. It allows you to have a database on your local machine rather than having to connect to a database on a remote SQL Server server.
  - 9 The ArcSDE client disk size is reflected in the total, but the memory size would be included in that of the running FDO application.
-

# Installing the FDO SDK

## Enterprise Edition

This edition contains the following providers: ArcSDE, MySQL, ODBC, Oracle, Raster, SDE, SHP, SQL Server, WFS, and WMS. The following description covers the installation of the FDO SDK and the ArcSDE 9.1 client. It does not cover the installation of the Oracle and SQL Server clients and servers or the installation of the ArcSDE server or the installation of MySQL.

## Windows XP

The FDO SDK is divided into two parts. The first part consists of the DLLs, which are bundled with the Autodesk Map® 3D 2007 product. The second part is the header files, library files, documentation, and sample code. These components are bundled with the Autodesk Map ObjectARX SDK.

Autodesk Map 3D 2007 is installed from a CD-ROM. For more information, see the Autodesk Map 3D documentation. After you have installed Autodesk Map 3D, the FDO DLLs are located in C:\Program Files\Autodesk Map 3D\FDO\bin.

The Autodesk Map ObjectARX SDK can be downloaded and installed from the Developer Center on the Autodesk web site.

## Installing the ObjectARX SDK

To install the Autodesk Map ObjectARX SDK, which includes the FDO SDK:

- 1 Go to <http://www.autodesk.com> ► Developers ► Products and Technologies ► Autodesk Map 3D ► ObjectARX ► License and Download.
- 2 Review the Notice To User and fill in the requested information form.
- 3 Click the radio button beside the statement I Accept The License Agreement. Download ObjectARX for Autodesk Map 3D 2007 (includes FDO SDK 3.0).
- 4 Click Submit. A dialog box is displayed.

- 5 In the dialog box, select the location where you want to save the zip file saved, and initiate the download.
- 6 Unzip the file. By default the file is unzipped to `C:\Map ObjectARX SDK\`.

If you plan to use the FDO Provider for ArcSDE, you must obtain the ArcSDE 9.1 client DLLs (`sg91.dll`, `sde91.dll`, and `pe91.dll`). Add the absolute path name of the folder containing the client DLLs to the system PATH environment variable.

## Uninstalling the ObjectARX SDK

To uninstall ObjectARX SDK, delete the `C:\Map ObjectARX SDK\` folder.

## Community Edition

The FDO source code is available along with the MapGuide® Open Source code. The source code for the Oracle, SQL Server, and Raster providers is not available as open source. For more information about this project, see the Open Source Geospatial Foundation at [www.OSGeo.org](http://www.OSGeo.org).



# Directory Structure

## Enterprise Edition

### Windows

#### C:\Program Files\Autodesk Map 3D\FDO\bin

```
Bin\  
    com\  
        <sql scripts>  
        <57 DLLs>  
        FDOATILConfiguration.xml  
        providers.xml  
        FDOUserManager.exe  
        XmiToFDO.xslt  
  
C:\Program Files\Autodesk Map 3D\Help\  
    FDG_FDODevGuide.pdf  
    FET_TheEssentialFDO.pdf  
    FDO_API.chm  
    FDO_API_managed.chm  
    FDO_SDK_ReadMe.chm  
    <provider chm files>  
    <Autodesk Map 3D and other Help files>
```

#### C:\Map ObjectARX SDK\FDO\

```
docs\  
    FDG_FDODevGuide.pdf  
    FET_TheEssentialFDO.pdf  
    FDO_API.chm  
    FDO_API_managed.chm  
    FDO_SDK_ReadMe.chm  
    <provider chm files>
```

```

XmlSchema\
  GML\
    <xsd files>
  XLink\
    CopyRight.txt
    xlink.xsd

  XmlSchema\
    base.css
    datatypes.xsd
    XmlSchema.xsd
    W3C Software License.html
  <xsd files>
  ReadMe.txt

inc\
  ATILSession\
    3 files

  common\
    Gml212\
      2 files
    Io\
      8 files
    Xml\
      12 files
    Xsl\
      1 file

  fdo\ 6 files
    ClientServices\ 8 files
    Commands\ 11 files
      DataStore\ 5 files
      Feature\ 9 files
      Locking\ 11 files
      LongTransaction\ 26 files
      Network\ 10 files
      Schema\ 10 files
      SpatialContext\ 7 files
      Sql\ 2 files
      Topology\ 6 files
      UnitOfMeasure\ 5 files

```

- Connections\ 7 files
  - Capabilities\ 14 files
  - Expression\ 32 files
  - Filter\ 20 files
  - Raster\ 7 files
  - Schema\ 45 files
  - Xml\ 22 files
- Geometry\ 28 files
  - AgfGeometry\ 1 file
  - GeometryStream\ 3 files
- RasterFile\ 1 file
  - Override\ 9 files
- Rdbms\ 12 files
  - Override\ 22 files
    - MySQL\ 20 files
    - ODBC\ 10 files
    - Oracle\ 23 files
    - SqlServer\ 20 files
- SDF\ 2 files
- Shp\ 2 files
  - Override\ 6 files
- Utilities\ 5 files
  - BulkCopy\ 7 files
  - Schema\ 1 file
  - UserMgr\ 2 files
- WMS\ 1 file
  - Override\ 9 files
- lib\ 18 files
- sample\
  - APIDocSandbox\ 30 files
  - DevGuideExample\ 44 files
  - OdbcSample\ 9 files
  - ProviderCapabilities\ 6 files
  - ProviderList\ 6 files
  - RasterFile\ 11 files
  - SampleArxApp\ 5 files

## About the Samples

The following sample programs are included with the FDO SDK:

- `FdoCI`. The FDO Command-line Interpreter can exercise the connection, user and data store creation, and spatial context interfaces in the FDO API. For more information, see [Microsoft Visual Studio 2005](#) (page 16) and the ReadMe file in this directory
- `DevGuideExample`. Exercises the FDO API for connections, capabilities, schema creation, inserts, deletes, updates, and selects. The code examples in the FDO Developer's Guide is taken from this source. For more information, see the ReadMe file in this directory.
- `ProviderCapabilities`. Returns the list of capabilities of each provider.
- `ProviderList`. Returns the list of FDO providers that are installed.
- `SampleArxApp`. This is an ObjectARX application that requires Autodesk Map 3D 2007 and the ObjectARX executable. See the ObjectARX Toolkit documentation. To compile the source code for this application, you must set the `OBJECTARXDIR` environment variable: set `OBJECTARXDIR=objectarx_install directory`.

## Programming Environment

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.

## Microsoft Visual Studio 2005

The compile-line and link-line options presented here are for a command-line FDO command interpreter built to exercise the connection, user creation, data store creation, and spatial context interfaces in this API. `main()` is in `CIMain.cpp`.

---

**NOTE** The use of the `/Yu` option (use precompiled headers) on the compile-line.

---

The contents of the `stdafx.h` file are as follows:

```

#pragma once
#define WIN32_LEAN_AND_MEAN
#include <stdio.h>
#include <tchar.h>
#include <cstring>
#include <cstdio>
#include <cstdlib>
#include <ctime>
#include <iostream>
#include <ostream>
#include <istream>
#include <fstream>
#include <sstream>
#include <exception>
#include <GisCommon.h>
#include <FDO.h>
#include <FDOStd.h>
#include <GisGeometry.h>
#include <FDOMessage.h>
#include <GisMessage.h>
#include <Rdbms/FDORdbmsCommandType.h>
#include <SDF/SdfCommandType.h>
#include <Utilities/UserMgr/FDOUserManager.h>

```

## Compile-Line Options

```

/Od /I "..\..\..\..\inc" /D "WIN32" /D "_DEBUG" /D "_CONSOLE" /D
"_UNICODE" /D "UNICODE" /Gm /EHsc /RTC1 /MDd /Yu"stdafx.h"
/Fp"Debug\CIMain.pch" /Fo"Debug\\" /Fd"Debug\vc80.pdb" /W3 /nologo
/c /Wp64 /ZI /TP /errorReport:prompt

```

## Link-Line Options

```

/OUT:"<somePath>\CIMain.exe" /INCREMENTAL /NOLOGO
/LIBPATH:"..\..\..\..\lib" /MANIFEST
/MANIFESTFILE:"Debug\CIMain.exe.intermediate.manifest" /DEBUG
/PDB:"<somePath>\CIMain.pdb" /SUBSYSTEM:CONSOLE /MACHINE:X86
/ERRORREPORT:PROMPT FDO.lib GisCommon.lib FDORdbms.lib BulkCopy.lib
FDOATILSession.lib RFPOverrides.lib FDOWmsOverrides.lib
GisGeometry.lib OdbcOverrides.lib ShpOverrides.lib MySQLOverrides.lib

```

```
OracleOverrides.lib RdbmsOverrides.lib Schema.lib UserMgr.lib  
SqlServerOverrides.lib kernel32.lib user32.lib gdi32.lib winspool.lib  
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib  
odbc32.lib odbccp32.lib
```

## Application Tasks

This document describes the following tasks. The various user-related and data store-related tasks are not supported by all providers. See the sections describing these APIs for details.

- 1 Create a connection object. For more information about any of the connection-related setps, see [Connection API](#) (page 20).
- 2 Set the connection properties
- 3 Open the connection in the pending state, if supported by the provider.
  - 1 Create a user account, if supported by the provider. For more information about creating a user account and granting privileges to a user, see [User Management API](#) (page 32).
  - 2 Create a data store, if supported by the provider. For more information, see the [Data Store Management API](#) (page 38).
  - 3 Grant privileges or roles to a user, if supported by a the provider.
- 4 Open the connection in the open state
- 5 Get a spatial context. For more information, see [Spatial Context API](#) (page 45).

## 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`.

## Provider Connection Targets

In a few cases you can verify the provider behavior documented here because you can connect to the same provider data source used to create this document as shown in the following table:

Provider	Connection String	Notes
OSGeo.ODBC.3.0	DataSourceName=Cities	1
Autodesk.Raster.3.0	DefaultRasterFileLocation=C:\Map ObjectARX SDK\FDO\sample\Raster- File\lefttop.jpg	
OSGeo.WFS.3.0	FeatureServer=http://www.bsc- eoc.org/cgi-bin/bsc_ows.asp	
OSGeo.WMS.3.0	FeatureServer= http://wms.jpl.nasa.gov/wms.cgi	

### NOTE

- 1 This assumes that you followed the procedure to map a data source name to a file. This is specified in the notes for the `DataSourceName` property in the connection properties table in [Connection API](#) (page 20). The file is `Cities.mdb`, and it is located in `<FDO SDK Install Location>\sample\OdbcSample\`.

## Provider Registry API

Use the `FdoFeatureAccessManager::GetProviderRegistry` static method to create an `IProviderRegistry` object. Query this object to get information about the providers available in this distribution. This information is also available in the XML file `C:\Program Files\Autodesk Map 3D\FDO\bin\providers.xml`.

## Connection API

You can use this API to connect to a provider. In general do the following:

- 1 Use the `FdoConnectionManager::CreateConnection` static method with a provider name argument, for example, `OSGeo.SHP.3.0`, to create an `FdoIConnection` object. The connection state is closed.
- 2 Use the `FdoIConnection` object to get an `FdoIConnectionInfo` object.
- 3 Use the `FdoIConnectionInfo` object to get an `FdoIConnectionPropertyDictionary` object.
- 4 Use the dictionary to query and set the properties required for the connection.
- 5 Use the connection object to open the connection. The resulting connection state may be open or pending. See the note.

---

**NOTE** Some providers having a data store property (ArcSDE, Oracle, MySQL, and SQL Server) support the existence of multiple data stores. You can only connect to one data store. To get a list of the available data stores, you must set the rest of the connection properties (Username, Password, and Service for all 4 providers plus the Instance property for the ArcSDE provider) and call the connection object's open method. The result is a pending connection state, and the data store property in the connection property dictionary is populated with the list of data stores. The data store property can then be set to one of the data store names in the list and another call made to the open method. This should result in an open connection state.

---



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.

Provider/Properties	Default Value	Required	Protected	Filename	Filepath	Datastore	Enumerable	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

Provider/Properties	Default Value	Required	Protected	Filename	Filepath	Datastore	Enumerable	Notes
Password		F	T	F	F	F	F	13
ConnectionString		F	F	F	F	F	F	
GenerateDefaultGeometryProperty	True	F	F	F	F	F	F	
Autodesk.Oracle.3.0								14
Username		T	F	F	F	F	F	8
Password		T	F	F	F	F	F	8
Service		T	F	F	F	F	F	15
DataStore		F	F	F	F	T	T	16
Autodesk.Raster.3.0								17
DefaultRasterFileLocation		F	F	F	F	F	F	18
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

Provider/Properties	Default Value	Required	Protected	Filename	Filepath	Datastore	Enumerable	Notes
TemporaryFileLocation		F	F	T	F	F	F	23
Autodesk.SqlServer.3.0								24
Username		F	F	F	F	F	F	25
Password		F	T	F	F	F	F	25
Service		T	F	F	F	F	F	26
DataStore		F	F	F	F	T	T	27
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	
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 Provider for 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 labeled Database 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 See [User Management API](#) (page 32) 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 Oracle 10g R1 Server - Standard Edition - 32-bit or 64-bit; Oracle 9i R2 (9.2.0.4) - Server - Standard Edition - 32-bit; Oracle 10g client required.
- 15 The Service property value identifies an entry in the `tnsnames.ora` file located in `C:\oracle\product\10.1.0\Db_1\NETWORK\ADMIN`. An entry has a format of `Service = (...)`.
- 16 An Oracle instance can contain many data stores. Follow the instructions in the note attached to the connection procedure that precedes this table to obtain a list of the available data stores.
- 17 See the list of raster file formats supported by this provider in [System Requirements](#) (page 6).
- 18 This is a file name. The named file is in the local file system.
- 19 The SDF provider supports the `.sdf` file type.
- 20 If True, only read operations on the file are permitted.
- 21 The SHP provider supports the `.shp` file type and the `.dbf` file type.

- 22 The DefaultFileLocation can be either a directory in the file system or the name of an existing file.
- 23 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.
- 24 The SQL Server provider uses the Windows ODBC component, in particular, the SQLDriverConnect function, to connect to a SQL Server instance.
- 25 If your Windows user name has been added to the remote SQL Server by an administrator or you are connecting to the SQL Express instance on your local machine, you do not have to provide values for the Username and Password properties. If you are connecting by way of SQL Server Authentication, you must provide a user name and password that has been added by an administrator. If you connect to a SQL Server instance with administrator privileges, you can use the FDO User Management API to add users.
- 26 If you can see the server name in the list of SQL Servers known to the ODBC Data Source Administrator, then you can connect to it using FDO. Use the name of the remote SQL Server as it appears in the list as the value of the Service property. If you want to connect to the SQL Express instance running on the local host, use <hostname>\SQLEXPRESS or .\SQLEXPRESS as the value of the Service property. Be sure the the SQL Server service is running on the local machine. To see the list of remote SQL Servers, do the following:
  - 1 In 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 ODBC Data Source Administrator dialog box, click the User DSN tab, click Add. The Create New Data Source dialog box is displayed.
  - 3 In the Create New Data Source dialog box, scroll down and select SQL Server, click OK. The Create a New Data Source to SQL Server dialog box is displayed.
  - 4 In the Create a New Data Source to SQL Server dialog box, click the down arrow-head on the spin box labeled Server to make the list of SQL Servers visible.

- 27 In SQL Server terms this is a database. If your connection to the SQL Server is in a pending state, you can create a data store using the [Data Store Management API](#) (page 38). Once you have created the data store, you can connect to it by setting the `DataStore` connection property and calling the connection object's `Open` method again.
- 

## 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. The list of commands is shown in the first table. The command attributes are shown in



the second table. “Y” means the command is supported in that particular provider.

Command	ArcSDE	MySQL	ODBC	Oracle	Raster	SDF	SHP	SqlServer	WFS	WMS	Notes
Select	Y		Y	Y	Y	Y	Y	Y	Y	Y	
SelectAggregates	Y	Y	Y	Y	Y	Y	Y	Y		Y	
Insert	Y	Y	Y	Y		Y	Y	Y			
Delete	Y	Y	Y	Y		Y	Y	Y			
Update	Y	Y	Y	Y		Y	Y	Y			
DescribeSchema	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
DescribeSchemaMapping			Y		Y		Y			Y	
ApplySchema		Y		Y		Y	Y	Y			
DestroySchema		Y		Y			Y	Y			
ActivateSpatialContext	Y	Y		Y				Y		Y	
CreateSpatialContext	Y	Y		Y		Y		Y			
DestroySpatialContext	Y	Y		Y				Y			
GetSpatialContexts	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
CreateDataStore		Y		Y		Y		Y			1

Command	ArcSDE	MySQL	ODBC	Oracle	Raster	SDF	SHP	SqlServer	WFS	WMS	Notes
DestroyDataStore		Y		Y		Y		Y			1
ListDataStores		Y		Y				Y			
CreateSDFFile						Y					
SQLCommand	Y	Y	Y	Y				Y			
AcquireLock	Y	Y		Y							
GetLockInfo	Y	Y		Y							
GetLockedObjects	Y	Y		Y							
GetLockOwners	Y	Y		Y							
ReleaseLock	Y	Y		Y							
ActivateLongTransaction	Y	Y		Y							
CommitLongTransaction	Y	Y		Y							
CreateLongTransaction	Y	Y		Y							
DeactivateLongTransaction	Y	Y		Y							
FreezeLongTransaction											
GetLongTransactions	Y	Y		Y							

Command	ArcSDE	MySQL	ODBC	Oracle	Raster	SDF	SHP	SqlServer	WFS	WMS	Notes
RollbackLongTransaction	Y	Y		Y							
CreateSpatialIndex		Y		Y				Y			
DestroySpatialIndex		Y		Y				Y			
GetSpatialIndexes		Y		Y				Y			
Attribute	ArcSDE	MySQL	ODBC	Oracle	Raster	SDF	SHP	SqlServer	WFS	WMS	Notes
Parameters	Y										
Timeout											
SelectExpressions		Y	Y	Y		Y	Y	Y			
SelectFunctions	Y	Y	Y	Y		Y	Y	Y			
SelectDistinct	Y	Y	Y	Y		Y	Y	Y			
SelectOrdering	Y	Y	Y	Y			Y	Y			
SelectGrouping		Y	Y	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 38).
- 

## 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/SQL Commands

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.

The following table maps methods to providers. 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 operation and the provider supports the operation, the table cell contains the SQL statements executed by the provider.

Operation	MySQL	Oracle	SQL Server	Notes
AddUser(name, password)	create user 'name' identified by password	create user name identified by password	exec sp_addlogin @loginame = 'name', @passwd	1,2,3

Operation	MySQL	Oracle	SQL Server	Notes
			<code>= 'pwd'; exec sp_addsrvrolemem- ber @loginname = 'name' @rolename = 'dbcreator'</code>	
ApplyWindowsUser(domain, name)	Not supported	Not supported	<code>EXEC sp_grantlogin @loginname = 'ads\camerod'"do- main/username'</code>	
AssignRoleOrPrivilege(name, roleOrPriv)	<code>grant <i>roleOrPrivilege</i> on *.* to <i>username</i></code>	<code>grant <i>roleOrPrivilege</i> to <i>username</i></code>	Not supported	4,5
DropUser(name)	<code>drop user name</code>	<code>drop user name</code>	<code>exec sp_revoked- baccess @name_in_db = 'name'; EXEC sp_droplogin @loginname = 'username'</code>	6
GetDatastores()	<code>FdoIListDatastores</code>	<code>FdoIListDatastores</code>	<code>FdoIListDatastores</code>	7
GetDbUsers()	<code>select user as name from mysql.user</code>	<code>select grantee as name from sys.dba_role_privs order by grantee</code>	<code>use master and then for each user on server, drop table Temp_RoleMembers52; create table Temp_RoleMembers52 (DbRole varchar(50), MemberName varchar(128), MemberSID varchar(50)); insert into</code>	8

Operation	MySQL	Oracle	SQL Server	Notes
			Temp_RoleMembers52 execute sp_helprolemember; drop table Temp_RoleMembers52	
GetDbUsers(in-Datastore)	select distinct grantee as name from information_schema.privileges where table_schema = <i>datastore</i>	Not supported	use <i>datastore</i> ; select name from sysusers where hasdbaccess=1 order by name asc;	9
GetRolesOrPrivileges()	select distinct privilege_type from information_schema.user_privileges where grantee like "'root'%" order by privilege_type	select role as role from sys.dba_roles order by role	select name as role from sysusers where issqlrole=1 order by name	10,11,12
GrantAccessTo-Datastore(user, datastore)	grant all on <i>datastore</i> .* to <i>user</i>	Not supported	USE <i>datastore</i> ; exec sp_grantdbaccess @loginame = ' <i>username</i> '; exec sp_addrolemember @rolename = 'db_datareader' @membername = ' <i>username</i> '	13
GrantAccessTo-Datastore(user, datastore)		.	exec sp_addrolemember @rolename = 'db_datawriter' @membername =	13

Operation	MySQL	Oracle	SQL Server	Notes
			<code>'username' ; exec sp_addrolemember @rolename = 'db_ddladmin' @membername = 'username'</code>	
GrantAccessTo- Datastore(user, datastore, priv)	<code>grant <i>priv</i> on <i>datastore</i>.* to <i>user</i></code>	Not supported	<code>USE <i>datastore</i>; exec sp_grantdbac- cess @loginame = 'username' ; exec sp_addrolemember, @rolename = 'priv', @member- name = 'user' ; use master</code>	14
RevokeAccessTo- Datastore( user, datastore, priv)	<code>revoke <i>roleOrPrivi- lege</i> on <i>datastore</i>.* from <i>username</i></code>	Not supported	<code>use <i>datastore</i>; exec sp_droprole- member @rolename = 'priv', @member- name = 'user' ; use master</code>	15
RevokeRoleOrPriv- ilege(name, roleOrPriv)	<code>revoke <i>roleOrPrivi- lege</i> on *.* from <i>username</i>.</code>	<code>revoke <i>roleOrPrivi- lege</i> from <i>username</i></code>	Not supported	16
SupportsDatas- toreGrants()	True	False	True	17
SupportsWin- dowsAuthentica- tion()	False	False	True	18

---

**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 The new Oracle user is assigned the `F_USER_ROLE` role. The SQL statement used to obtain this information is `select granted_role from sys.dba_role_privs where grantee = 'username';`. A new Oracle user is assigned the 'unlimited tablespace,' 'execute any indextype,' and 'execute any procedure' privileges. The SQL statement used to obtain this information is `select privilege from sys.dba_sys_privs where grantee = 'username';` A new Oracle user is assigned a long list of privileges as a result of being assigned the role of `F_USER_ROLE`. You can determine the contents of this list using the SQL statement `select privilege from sys.role_sys_privs where role = 'F_USER_ROLE';` The list is drop user, alter user, analyze any, create user, alter session, drop any view, create session, drop any index, drop any table, grant any role, lock any table, alter any index, alter any table, create any view, create any index, create any table, create procedure, delete any table, drop any synonym, insert any table, select any table, update any table, drop any sequence, alter any sequence, create any synonym, create any sequence, grant any privilege, select any sequence, global query rewrite, and grant any object privilege. The `F_USER_ROLE` also has default table privileges. You can determine what these are by the SQL statement `select table_name, privilege from sys.role_tab_privs where role = 'rolename';`
- 3 A new SQL Server user created using the stored procedure `sp_login` is given one system privilege `db_creator`. The user can login using SQL Server authentication. Call `GrantAccessToDatastore(username, datastore)` to grant the user default privileges for the data store.
- 4 To confirm the results of this method call, call `GetDBUsers()`.
- 5 User privileges extend across all data stores defined in the Oracle instance.



- 6 To drop a windows user (SQL Server only), format the argument as `L"domain name\user name"`
- 7 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. In Oracle, user names and data store names are classed as Oracle users. The Oracle SQL command `SQL> select username from dba_users;` shows data store and user names
- 8 All users are listed. SQL Server reports the user privileges as well as the user name. If you are connected to MySQL and you created the user with `AddUser`, `USAGE` is shown as the privilege. If you are connected to SQL Server and you created the user with `AddUser`, `db_creator` is shown as the privilege.
- 9 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.
- 10 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`.
- 11 The list of supported roles for Oracle consists of `AQ_ADMINISTRATIVE_ROLE`, `AQ_USER_ROLE`, `AUTHENTICATED_USER`, `CONNECT`, `CTXAPP`, `DBA`, `DELETE_CATALOG_ROLE`, `DMUSER_ROLE`, `DM_CATALOG_ROLE`, `EJBCCLIENT`, `EXECUTE_CATALOG_ROLE`, `EXP_FULL_DATABASE`, `F_SCHEMA_ROLE`, `F_USER_ROLE`, `GATHER_SYSTEM_STATISTICS`, `GLOBAL_AQ_USER_ROLE`, `HS_ADMIN_ROLE`, `IMP_FULL_DATABASE`, `JAVADEBUGPRIV`, `JAVAIDPRIV`, `JAVASYSPRIV`, `JAVAUSERPRIV`, `JAVA_ADMIN`, `JAVA_DEPLOY`, `LOGSTDBY_ADMINISTRATOR`, `MGMT_USER`, `OEM_MONITOR`, `RECOVERY_CATALOG_OWNER`, `RESOURCE`, `SCHEDULER_ADMIN`, `SELECT_CATALOG_ROLE`, `WKUSER`, `WM_ADMIN_ROLE`, and `XDBADMIN`.
- 12 The list of supported privileges for SQL Server consists of `db_accessadmin`, `db_backupoperator`, `db_datareader`, `db_datawriter`, `db_ddladmin`, `db_denydatareader`, `db_denydatawriter`, `db_owner`, `db_securityadmin`, and `public`.

- 13 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, create view, delete, drop, execute, index, insert, lock tables, references, select, show view, and update. For SQL Server the privileges are db\_datareader, db\_datawriter, and db\_ddladmin. Call `GetDBUsers(<datastore>)` to verify the result.
  - 14 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.
  - 15 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.
  - 16 This method takes two arguments: a user name and a role or privilege name. It revokes the specified role or privilege.
  - 17 If True, you can call the two `GrantAccessToDatastore` methods and the `RevokeAccessToDatastore` method.
  - 18 If True, you can call the `ApplyWindowUser` 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 `Fdo ICommandCapabilities` 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.

## Data Store Management Properties

This table shows the data store management properties for each provider. “T” means true, and “F” means false. For important details, refer to the associated Notes.

Provider/Properties	Default Value	Required	Protected	Filename	Datastore	Enumerable	Notes
OSGeo.MySQL.3.0							
DataStore		T	F	F	T	F	1
Description		F	F	F	F	F	
Autodesk.Oracle.3.0							
DataStore		T	F	F	T	F	1
Password		T	T	F	F	F	2
Description		F	F	F	F	F	
LtMode	OWM	F	F	F	F	T	3
TableSpace		F	F	F	F	F	4
OSGeo.SDF.3.0							
File		T	F	T	F	F	5
Autodesk.SqlServer.3.0							
DataStore		T	F	F	T	F	1

Provider/Properties	Default Value	Required	Protected	Filename	Datastore	Enumerable	Notes
Password		T	T	F	F	F	2
Description		F	F	F	F	F	
LtMode	FDO	F	F	F	F	T	3
LockMode	FDO	F	F	F	F	T	6

#### NOTE

- 1 A data store is a container in the data source to which you can apply a feature schema.
- 2 A data store is password protected.
- 3 LtMode stands for Long Transaction Mode. .
  - For Oracle the possible values are OWM and NONE. OWM stands for Oracle Workspace Manager. NONE means that long transactions and locking are not supported; OWM means that they are.
- 4 If no value is given to the `TableSpace` property, the data store is created in the default `USERS` tablespace. Otherwise, the data store is created in the named tablespace.
- 5 The File property takes a file name value. The named file is in the local filesystem.
- 6 LockMode indicates whether persistent locks are supported. FDO means that the class tables have additional columns for maintaining persistent locks. NONE means that persistent locks are not supported. As with Oracle, LtMode pertains to long transactions. LockMode and LtMode must both be either FDO or NONE. That is, they are either both enabled or both disabled.

## Data Store Management SQL Commands

These are the SQL commands executed when the `FdoICreateDataStore`, `FdoIDestroyDataStore`, and `FdoIListDataStores` commands are executed.

### FdoICreateDataStore

#### Oracle

- `grant create session, alter session, unlimited tablespace, create table, create view, create synonym, create procedure, create sequence, WM_ADMIN_ROLE to, datastorename identified by "password"`
- `grant select, update, insert, delete on mdsys.sdo_geom_metadata_table to datastorename`
- `alter user datastorename temporary tablespace temp`
- `alter user datastorename default tablespace users`
- `insert into f_schemainfo (schemaname, description, creationdate, owner, schemaversionid) values (syscontext('userenv', 'current_schema'), 'System Metaschema', SYSDATE, user, 3000)`
- 86 additional SQL commands

#### SQL Server

- `create database datastorename`
- 49 additional SQL commands

#### MySQL

- `create database databaseName character set latin1 collate latin1_bin`

- `insert into f_schemainfo (schemaname, description, creationdate, owner, schemaversionid ) values (database(), 'System Metaschema', current_date(), substring_index(current_user(), '_utf8'@', 1), 3.000)`
- `insert into f_schemainfo (schemaname, description, creationdate, owner, schemaversionid ) values ('F_MetaClass', 'Special classes for FDO Feature metaclasses', current_date(), substring_index(current_user(), '_utf8'@', 1), 3.000)`
- 45 other SQL commands

## FdoIDestroyDataStore

### MySQL

- `drop database if exists datastorename`

### Oracle

- `drop user datastorename cascade`

### SQL Server

- `drop database database name`



## FdoListDataStores

### SQL Server

For each database the following SQL is executed.

```
■ select table_schema as schemas_table from
   database.information_schema.tables T where T.table_name =
   'F_SCHEMAINFO'
```

## 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.

The following table defines default values for the spatial context properties for the providers that support the creation of a data store:

Properties/Provider	MySQL	Oracle	SDF	SqlServer	Notes
Coordinate system	Cartesian	Cartesian	Cartesian	Cartesian	
Extent type	Static	Static	Dynamic	Static	1

Properties/Provider	MySQL	Oracle	SDF	SqlServer	Notes
Extent	Not NULL	Not NULL	NULL	Not NULL	2
Unit dimension	XY	XY	Unde- fined	XY	3
XY tolerance	.001	.001	0	.001	4
Z tolerance	.001	.001	0	.001	4
Active	False	False	True	False	5

#### NOTE

- 1 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.
- 2 The default extents expressed in AGF text format for the various providers are as follows:
  - MySQL, Oracle and SQL Server: POLYGON ((-2000000 -2000000, 2000000 -2000000, 2000000 2000000, -2000000 2000000, -2000000 -2000000))
- 3 The dimensionality is that of the extent. It is not set directly.
- 4 See [Tolerance](#) (page 50).
- 5 Only ArcSDE, MySQL, Oracle and SQL Server providers can have multiple spatial contexts defined and can, therefore, show contexts in the reader that are not active.

The following table defines values for the spatial context properties for the providers whose data store has been created by a non-FDO process. The data sources for ODBC, Raster, WFS, and WMS are listed in [Provider Connection Targets](#) (page 19).

Properties/Provider	ODBC	Raster	WFS	WMS	Notes
Coordinate System	Cartesian	Cartesian	EPSG:4326	EPSG:4326	1
Extent type	Static	Static	Static	Static	2
Extent	Not NULL	Not NULL	Not NULL	Not NULL	3
Unit dimension	XY	XY	XY	XY	4
XY tolerance	.001	.001	.01	.001	5
Z tolerance	.001	.001	.01	.001	5
Active	F	T	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 extents expressed in AGF text format are as follows:
    - ODBC: POLYGON ((-2000000 -2000000, 2000000 -2000000, 2000000 2000000, -2000000 2000000, -2000000 -2000000))
    - Raster: POLYGON ((-10000000 -10000000, 10000000 -10000000, 10000000 10000000, -10000000 10000000, -10000000 -10000000))
    - WFS: POLYGON ((-141.238 40.4143, -52.5407 40.4143, -52.5407 78.1059, -141.238 78.1059, -141.238 40.4143))
    - WMS: POLYGON ((-180 -90, 180 -90, 180 90, -180 90, -180 -90))
  - 4 The dimensionality is that of the extent. It is not set directly.
  - 5 See [Tolerance](#) (page 50).
  - 6 Only ArcSDE, MySQL, Oracle and SQL Server providers can have multiple spatial contexts defined and can, therefore, show contexts in the reader that are not active.
-

## Known Problems

### MySQL Provider

This provider was throwing exceptions because the FDO program was loading a version of the `libMySQL.dll` from the PHP installation rather than the MySQL installation. The problem was corrected by placing the PHP path after the MySQL path in the system PATH variable.

## Exception Messages

### Connection Not Established

This exception message occurs when there is a closed connection to either SQL Server or 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>)`.

## Membership of the public role cannot be changed

This exception message occurs when there is a pending connection to SQL Server and a call is made to `GrantAccessToDatastore(<user>, <datastore>, L"public")`.

## 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.

# Index

## A

API 16, 18, 20, 25, 27–28, 32, 45  
    capabilities 16, 28  
    connection 18, 20  
    provider registry 20  
    spatial context 18, 45  
    user management 18, 25, 27, 32  
ArcSDE client 8–12  
ArcSDE server 7–8, 24  
Autodesk Map 3D 2007 11, 16  
Autodesk Map ObjectARX SDK 11–12, 16

## C

Capabilities API 16, 28  
capabilities, command 28  
capabilities, user management 32  
collection classes 19  
Connection API 18–20  
connection properties 21, 25, 45, 49  
connection target 19

## D

data store 3, 18, 20, 32, 38–42  
Data Store Management API 18, 28, 38  
data store management properties 41  
data store, create or destroy 39–40

## F

FDO Command-line Interpreter 10, 16  
FDO provider 3, 5  
FDO Provider for ArcSDE 5, 11–12, 20–21, 24–25, 45–46, 48  
FDO Provider for MySQL 5, 11, 20–21, 35, 38–39, 41, 45–46, 48–49  
FDO Provider for ODBC 6, 11, 19, 21, 47–48

FDO Provider for Oracle 5, 10–12, 20, 22, 28, 35, 38–39, 41–42, 45–46, 48  
FDO Provider for Raster 6, 11–12, 19, 22, 47–48  
FDO Provider for SDF 5, 11, 22, 26, 38, 40–41, 45  
FDO Provider for SHP 5, 11, 20, 22, 26  
FDO Provider for SQL Server 5–6, 11–12, 20, 27, 35, 38–39, 41, 45–46, 48–50  
FDO Provider for WFS 4, 6, 11, 19, 23, 47–48  
FDO Provider for WMS 6, 11, 19, 23, 47–48  
FdoConnectionManager 20  
FdoIActivateSpatialContext 45  
FdoICommandCapabilities 28, 38  
FdoIConnection 20  
FdoIConnectionCapabilities 28  
FdoIConnectionInfo 20  
FdoIConnectionPropertyDictionary 20  
FdoICreateDataStore 39–40, 43  
FdoICreateSpatialContext 45–46, 48  
FdoIDestroyDataStore 39–40, 43  
FdoIDestroySpatialContext 45  
FdoIExpressionCapabilities 28  
FdoIFilterCapabilities 28  
FdoIGetSpatialContexts 45  
FdoIListDataStores 32, 39–40, 43  
FdoIRasterCapabilities 28  
FdoISchemaCapabilities 28  
FdoUserManager 32, 36

## M

MapGuide Open Source 12  
memory management 8, 18  
Method 20, 28, 32–38, 49–50  
    AddUser 32, 37  
    ApplyWindowsUser 32

- AssignRoleOrPrivilege 32
- CreateConnection 20
- DropUser 32
- GetCommandCapabilities 28, 32
- GetDatastores 32
- GetDbUsers 33, 36, 38
- GetRolesOrPrivileges 34, 49
- GrantAccessToDatastore 34–38, 49–50
- RevokeAccessToDatastore 35, 38
- RevokeRoleOrPrivilege 35
- SupportDataStoreGrants 32
- SupportsDataStoreGrants 35
- SupportsWindowsAuthentication 35
- Microsoft Access 8, 25–26
- Microsoft Excel 8, 25
- MySQL database 3, 7, 9–10, 25, 36

## O

- ODBC data source 26–27
- Open Source Geospatial Foundation 5, 12
- Oracle database 3, 7–10, 24–26
- Oracle user 36

## P

- privileges, MySQL 38, 49
- privileges, SQL Server 24, 36–38
- Property 19–28, 37, 41–42, 49
  - ConnectionString 22
  - DataSourceName 19, 21
  - Datastore 21
  - DataStore 21–23, 25, 28, 37, 41, 49
  - DefaultFileLocation 22, 27
  - DefaultImageHeight 23
  - DefaultRasterFileLocation 19, 22
  - Description 41–42
  - FeatureServer 19, 23
  - File 22, 41–42

- GenerateDefaultGeometryProperty22
- Instance 20–21, 24
- LockMode 42
- LtMode 41–42
- Password 20–23, 27, 41–42
- ReadOnly 22
- Server 21, 24
- Service 20–23, 25–27
- TableSpace 41–42
- TemporaryFileLocation 23, 27
- UserId 21
- Username 20–23, 27
- Provider Registry API 20

## R

- raster file formats 7–8, 26
- RDBMS 5–6, 38
- roles, Oracle 36, 38

## S

- sample code 9
- SDF file 3, 7
- SHP file 3, 7
- Spatial Context API 18, 45
- spatial context properties 45
- SQL command 32, 39, 43–44
- SQL Server database 6–8, 24–25, 27–28
- SQL Server Express 9–10
- SQL Server user 36

## U

- User Management API 18, 25, 27, 32

## W

- Web Feature Service 7–8
- Web Map Service 7–8, 48