

Feature Data Objects Open Source

Building FDO Open Source

FDO Open Source

July 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

Chapter 1	Building FDO on Windows	1
	Introduction	2
	Disk Space	2
	Install the Repository Tool	2
	Get the Source	3
	Subversion Repositories	3
	Review the Copyright and Licensing Information	4
	Update the Source	4
	Install the Build Tools	6
	Binary Build Tools	6
	Documentation Build Tools	7
	Install RDBMS Clients	8
	ArcSDE 9.1 Client SDK	8
	MySQL Client	10
	Set the Environment Variables	11
	Modifying Microsoft Solution or Project Files	11
	Building FDO	12
	Build Scripts	12
	Build Order	12
	Run the build_thirdparty.bat Script	12
	Optionally Run the build_parse.bat Script	13
	Run the build.bat Script	14
	Unit Tests	17

	Introduction	17
	Run the FDO Unit Tests	17
	Configure the ArcSDE Server	18
	Run the ArcSDE Unit Tests	20
	Run the GDAL Unit Tests	22
	Run the MySQL Unit Tests	23
	Configure the ODBC Unit Tests	23
	Run the ODBC Unit Tests	24
	Run the SDF Unit Tests	25
	Run the SHP Unit Tests	25
	Run the WMS Unit Tests	25
Chapter 2	Building FDO on Linux	27
	Introduction	28
	Disk Space	28
	Install the Repository Tool	28
	Get the Source	29
	Get the Source	29
	Subversion Repositories	29
	Review the Copyright and Licensing Information	31
	Updating the Source	31
	Install the Build Tools	33
	Binary Build Tools	33
	Documentation Build Tools	34
	Install RDBMS Clients	34
	ArcSDE 9.1 Client SDK	34
	MySQL Client	36
	ODBC Oracle Driver	36
	Environment Variables	38
	Building FDO	39
	Build Scripts	39
	Build Order	40
	Run the build_thirdparty.sh Script	40
	Optionally Run the build_parse.sh Script	40
	Run the build_linux.sh Script	41
	Unit Tests	44
	Introduction	44
	Run the FDO Unit Tests	44
	Run the ArcSDE Unit Tests	44
	Run the GDAL Unit Tests	45
	Run the MySQL Unit Tests	45
	Run the ODBC Unit Tests	45
	Run the SDF Unit Tests	46
	Run the SHP Unit Tests	46
	Run the WMS Unit Tests	47

Index	49
------------------------	-----------

Building FDO on Windows

In this chapter

- [Introduction](#)
- [Disk Space](#)
- [Install the Repository Tool](#)
- [Get the Source](#)
- [Review the Copyright and Licensing Information](#)
- [Update the Source](#)
- [Install the Build Tools](#)
- [Install RDBMS Clients](#)
- [Set the Environment Variables](#)
- [Modifying Microsoft Solution or Project Files](#)
- [Building FDO](#)
- [Unit Tests](#)

Introduction

This chapter describes:

- how to build the Feature Data Object binaries from the source files and install them on a Windows machine
- the build tools that you must install to do the build
- the database clients that you must install to do the build

NOTE The debug binaries built by this process include the unit test executables.

NOTE The Windows system documented here is XP.

A briefer description of the build process is in the `OpenSourceBuild_README.txt` file in the build folder.

Whether you intend to build applications on top of FDO or modify the FDO code itself, you must build the binaries from the source.

Disk Space

The worse case requirement for build and install disk space occurs if you build both the debug and release versions including the documentation and install the debug version. In this case you require 5.84GB of disk space. This requirement is exclusive of the disk space required for the Subversion client (), the build tools (1.84GB) and the ArcSDE and MySQL provider clients (113MB). The final total is 7.8GB.

Install the Repository Tool

The FDO code is stored in Subversion repositories on the Open Source Geospatial Foundation website. Use a Subversion client to obtain the FDO source from these repositories. To browse the list of available clients, click the Get help? link at the left-hand side of the <https://www.osgeo.org> page. A new browser window is displayed; click the About source code version control with Subversion link. Follow the instructions there to obtain and install a client. The instructions in this document are based on the use of the SVN command-line client.

Get the Source

The FDO source code is located on the Open Source Geospatial Foundation website. You must first become a registered user of the site and then login before you can download source from the site. Go to <https://www.osgeo.org>. In the upper right-hand corner there is a link labeled Register. Click on this link and follow the instructions for registering and logging in. Once you have logged in, you may get the source by doing a “checkout” from a set of Subversion repositories.

NOTE The “checkout” is read-only.

Subversion Repositories

The SVN client enables you to download files from a Subversion repository into a build folder. There is one repository for the utilities, fdo core and thirdparty components and one repository for each of the providers with the exception of the MySQL and ODBC providers, which share a repository. A URL identifies the repository and a local path identifies the folder which receives the downloaded files. In this document the build folder is called C:\OpenSource.

To get the source from the repositories do the following in a cmd.exe window.

NOTE You will be prompted for a password.

NOTE The parallelism between the third and fourth arguments of the svn command, for example, between “...trunk/Providers/ArcSDE” and “C:\OpenSource\Providers\ArcSDE”.

NOTE The fourth argument can be an absolute or relative path.

- 1 `svn checkout https://fdocore.osgeo.org/svn/fdocore/trunk
C:\OpenSource --username yourusername`
- 2 `svn checkout
https://fdoarcsde.osgeo.org/svn/fdoarcsde/trunk/Providers/ArcSDE
C:\OpenSource\Providers\ArcSDE --username yourusername`
- 3 `svn checkout
https://fdogdal.osgeo.org/svn/fdogdal/trunk/Providers/GDAL
C:\OpenSource\Providers\GDAL --username yourusername`

- 4 `svn checkout`
`https://fdordbms.osgeo.org/svn/fdordbms/trunk/Providers/GenericRdbms`
`C:\OpenSource\Providers\GenericRdbms --username yourusername`
- 5 `svn checkout`
`https://fdosdf.osgeo.org/svn/fdosdf/trunk/Providers/SDF`
`C:\OpenSource\Providers\SDF --username yourusername`
- 6 `svn checkout`
`https://fdoshp.osgeo.org/svn/fdoshp/trunk/Providers/SHP`
`C:\OpenSource\Providers\SHP --username yourusername`
- 7 `svn checkout`
`https://fdowfs.osgeo.org/svn/fdowfs/trunk/Providers/WFS`
`C:\OpenSource\Providers\WFS --username yourusername`
- 8 `svn checkout`
`https://fdowms.osgeo.org/svn/fdowms/trunk/Providers/WMS`
`C:\OpenSource\Providers\WMS --username yourusername`

NOTE The fdcore components includes a script called `checkoutsvn.bat`, which can be used to get updates for the all of the components from the Subversion repositories.

Review the Copyright and Licensing Information

The copyright and licensing information for the FDO API is contained in the `License_README.txt` file in the build folder.

Update the Source

You can update the source by running the `checkoutsvn.bat` script.

In a `cmd.exe` window where the current directory is the build folder, enter `checkoutsvn -h`. The help text displayed is as follows:

- `checkoutsvn.bat [-h] [-o=OutFolder] [-w=WithModule] -s=Source -u=UserId -p=UserPassword`
- Help: `-h[elp]`

- Branch: -s[source]=location of source file, either branch location, tag location or trunk

NOTE The possible arguments are -s=trunk, -s=<tagIdentifier>, and -s=<branchIdentifier>. Get the tag or branch identifier by browsing <http://fdo.osgeo.org>, clicking the Subversion link and then clicking the tag or branches link.

- OutFolder: -o[utpath]=destination folder for files
- WithModule: -w[ith]=all(default), fdo, fdcore, thirdparty, providers, utilities, shp, sdf, wfs, wms, arcsde, rdbms, gdal
- User: -u[ser]=user id
- Password: -p[assword]=user password

You can update all of the source by running the checkoutsvn.bat script in the build folder. The syntax is `checkoutsvn.bat -s=<source> -u=yourusername -p=yourpassword`.

You can update the source for specified components by using the with option. The syntax is `checkoutsvn.bat -s=<source> -u=yourusername -p=yourpassword -w=desiredComponent`.

The possible arguments for the with option are:

- all - checkout all source from the fdcore, fdosdp, fdosdf, fdowms, fdowfs, fdoarcsde, fdordbms and fdogdal subversions
- providers - checkout all source from the fdosdp, fdosdf, fdowms, fdowfs, fdoarcsde, fdordbms and gdal subversions
- fdcore - checkout all source from the fdcore.osgeo.org subversion
- thirdparty - checkout all source from the Thirdparty folder in the fdcore.osgeo.org subversion
- fdo - checkout all source from the Fdo folder in the fdcore.osgeo.org subversion
- utilities - checkout all source from the Utilities folder in the fdcore.osgeo.org subversion
- arcsde - checkout all source from the fdoarcsde.osgeo.org subversion
- gdal - checkout all source from the fdogdal.osgeo.org subversion

- `rdbms` - checkout all source from the `fdordbms.osgeo.org` subversion
- `sdf` - checkout all source from the `fdosdf.osgeo.org` subversion
- `shp` - checkout all source from the `fdoshp.osgeo.org` subversion
- `wfs` - checkout all source from the `fdowfs.osgeo.org` subversion
- `wms` - checkout all source from the `fdowms.osgeo.org` subversion

Install the Build Tools

Binary Build Tools

The following software is used during the process of building the FDO binaries.

Tool	Description	Path	Download from
msbuild	the Visual Studio .NET build engine	C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727	http://msdn.microsoft.com/net-framework/downloads/updates/default.aspx
vcbuild	Builds Visual C++ projects and solutions.	C:\Program Files\Microsoft Visual Studio 8\VC\vcpackages	Microsoft Visual Studio 2005 from install CD
devenv	Builds a solution using a specified solution configuration file.	C:\Program Files\Microsoft Visual Studio 8\Common7\IDE	Microsoft Visual Studio 2005 from install CD
cscript	the command-line version of the Windows Script Host that provides command-line options for setting script properties	C:\Windows\system32	Comes with the XP operating system
bison	a general purpose parser generator that converts a grammar description for an	C:\Program Files\GnuWin32\bin	bison-1.875-4.exe from http://prdownloads.sourceforge.net/gnuwin32/

Tool	Description	Path	Download from
	LALR context-free grammar into a C program to parse that grammar.. If you intend to change the .y files that define the FDO expression and filter language, you must install this utility.		
sed	(streams editor) is a text filter tool. It takes text input, performs one or more operations on the text and outputs the modified text. If you intend to change the .y files that define the FDO expression and filter language, you must install this utility.	C:\Program Files\GnuWin32\bin	sed-4.1.4.exe from http://gnuwin32.sourceforge.net/packages.html
perl	scripting language. This is used to build the FDO third-party OpenSSL components and to run the ArcSDE unit tests.	C:\Perl\bin	http://www.activestate.com/ActivePerl

NOTE FDO has been tested using Bison 1.875 and Sed 4.1.4 and will not work with other versions of these binaries.

Documentation Build Tools

The following software is used during the process of building the FDO documentation.

Tool	Description	Path	Download from
doxygen	generates API documentation from specially formatted comments	C:\Program Files\doxygen\bin	doxygen-1.4.6-setup.exe from http://www.stack.nl/~di-

Tool	Description	Path	Download from
	ments embedded in C++, C, Java, Objective-C, Python, IDL, PHP, C#, and D.		mitri/doxygen/download.html#latestsrc
graphviz	is a graph drawing toolkit used by doxygen to draw class diagrams.	C:\Program Files\ATT\Graphviz\bin	graphviz-2.8.exe from http://www.graphviz.org/Download_windows.php
HTML Help Workshop	is the standard help system for the Windows platform. It is used to generate compressed html help (.chm) files. It is used by doxygen to generate .chm files containing the API documentation.	C:\Program Files\HTML Help Workshop	Htmlhelp.exe from http://msdn.microsoft.com/library/en-us/htmlhelp/hwMicrosoftHTMLHelpDownloads.asp
<p>NOTE The doxygen and graphviz installers add their respective bin folders to the system PATH variable.</p>			
<p>NOTE An alternate way to get to the HTML Help downloads page is to go to http://msdn.microsoft.com/library, double-click Win32 and COM Development, double-click Tools, double-click HTML Help, double-click Microsoft HTML Help 1.4 SDK, and then click Downloads.</p>			

Install RDBMS Clients

Building the two Rdbms-based FDO Providers, ArcSDE and MySQL, is dependent on client libraries being available.

ArcSDE 9.1 Client SDK

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/arcade/how-to-buy.html>.

Install the client to a location outside of the build folder. For the purpose this document the location of the ArcSDE client is
C:\ESRI\ArcSDEClient91\Windows\.

The list of ArcSDE client files follows.

- lib\icuuc.lib
- lib\pe91.lib
- lib\sde91.lib
- lib\sg91.lib
- include\sg.h
- include\sgerr.h
- include\pe.h
- include\pe_coordsys_from_prj.h
- include\pedef.h
- include\pef.h
- include\sdeerno.h
- include\sderaster.h
- include\sdetype.h
- bin\edgmt.dll
- bin\gsrvrdb291.dll
- bin\gsrvrinf91.dll
- bin\gsrvrora8i91.dll
- bin\gsrvrora9i91.dll
- bin\gsrvrsg191.dll
- bin\icudt221.dll
- bin\icuuc22.dll
- bin\libtiff.dll
- bin\loceng.dll

- bin\locssa.dll
- bin\mtchloc.dll
- bin\mtchmt.dll
- bin\pe91.dll
- bin\sde91.dll
- bin\sdedb2srvr91.dll
- bin\sdeinfsrvr91.dll
- bin\sdeora8isrvr91.dll
- bin\sdeora9isrvr91.dll
- bin\sdesqlsrvr91.dll
- bin\sg91.dll
- bin\xerces-c_2_1_0.dll

MySQL Client

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 custom install to the default location (C:\Program Files\MySQL\MySQL Server 5.0); this allows you to request the installation of all of the developer components. The developer components are required for the MySQL provider build.

NOTE Check that the include and lib folders are present and that the lib folder contains Opt and Debug subfolders. If you notice that the Debug folder is missing, create it and copy the .lib files from Opt to Debug.

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.

Set the Environment Variables

The setenvironment.bat script in the build folder sets the following environment variables required by the build.bat script.

- FDO
- FDOMYSQL
- FDOTHIRDPARTY
- FDOUTILITIES
- NLSDIR
- PATH
- SDEHOME
- XALANROOT
- XERCESCROOT

NOTE Review the script to verify that the paths for the .NET Framework, Graphviz, doxygen, and Visual Studio and the values for SDEHOME and FDOMYSQL are correct. You may wish to change some of these values to agree with how you have set up your environment.

NOTE The path to the HTML Help compiler (hhc.exe) does not have to be added to the PATH variable since the full path to the executable is specified in the configuration files used by doxygen to generate the API documentation.

In a cmd.exe window change directory to the build folder and run the setenvironment.bat script.

Modifying Microsoft Solution or Project Files

If you decide to modify the solution (.sln) or project (.vcproj) files, be aware that they use path references that contain environment variables defined in the setenvironment.bat script. Launch Visual Studio from a shell in which setenvironment.bat has been run. Then open the solution or project file using a Visual Studio file menu operation.

Building FDO

Build Scripts

The build folder contains three bat files:

- `build_thirdparty.bat` - builds and installs third-party DLLs used by the providers
- `build_parse.bat` - recreates the FDO filter and expression language grammar source files from yacc files
- `build.bat` - builds and installs the fdo core and provider DLLs

Build Order

Build the thirdparty components first. Optionally generate the FDO filter and expression language grammar source files if you modified them. Build the FDO core and provider components.

Run the `build_thirdparty.bat` Script

The assumption is that you are not changing the third-party software. So these binaries need only be built once. If you notice a third-party component has been updated during a run of `checkoutsvn.bat`, you must run this script again. You can use command-line options to limit the build to the components that have changed.

In the same `cmd.exe` window where you ran the `setenvironment.bat` script enter `build_thirdparty -h`. The help text is contained in the following list:

- `build_thirdparty.bat [-h] [-o=OutFolder] [-c=BuildType] [-a=Action] [-w=WithModule]`
- Help: `-h[elp]`
- OutFolder: `-o[utpath]=destination folder for binaries`
- BuildType: `-c[onfig]=release(default), debug`
- Action: `-a[ction]=build(default), buildinstall, install, clean`

- WithModule: -w[ith]=all(default), fdo, providers, sdf, wfs, wms

- BuildDocs: -d[ocs]=skip(default), build

The default action is to build all of the thirdparty release DLLs and write them to Thirdparty subfolders.

If you only specify the -c=debug option, you get all of the thirdparty debug DLLs written to Thirdparty subfolders.

If you specify the -a=buildinstall option, you get DLLs copied from Thirdparty subfolders to Fdo\Bin in the folder specified using the -o option.

If you specify the -w=fdo option, you get apache DLLs built and written to Thirdparty\apache subfolders. If you specified an install option, you get the apache DLLs copied to Fdo\Bin in the folder specified using the -o option.

If you specify the -w=sdf option, you get a .lib built and written to a Thirdparty\Sqllite3.1.5 subfolder. This .lib is used during the SDF provider build and is only used by the SDF provider. An install option has no effect.

If you specify the -w=wfs option, you get a curl lib built and written to a libcurl subfolder, an openssl lib built and written to an openssl subfolder, and a boost DLL built and written to a boost subfolder. If you specified an install option, you get the boost DLL copied to Fdo\Bin in the folder specified using the -o option.

If you specify the -w=wfs option, you get a curl lib built and written to a libcurl subfolder, an openssl lib built and written to an openssl subfolder, a boost DLL built and written to a boost subfolder, and a GDAL DLL built and written to a GDAL1.3 subfolder. If you specified an install option, you get the boost and GDAL (gdal13.dll) DLLs copied to Fdo\Bin in the folder specified using the -o option.

NOTE The gdal13.dll file is used by the WMS provider. The GDAL provider is a different .dll file.

Optionally Run the build_parse.bat Script

The FDO build uses several source (.cpp and .h) files generated from .y files by the Bison and Sed utilities. These .y files define a grammar for the FDO expression and filter language. If you change the .y files, you must run the build_parse.bat script to regenerate the source files.

Run the build.bat Script

In the same cmd.exe window where you ran the setenvironment.bat script enter `build -h`. The build help text is contained in the following list:

- `build.bat [-h] [-o=OutFolder] [-c=BuildType] [-a=Action] [-w=WithModule] [-d=BuildDocs]`
- Help: `-h[elp]`
- OutFolder: `-o[utpath]=destination folder for binaries`
- BuildType: `-c[onfig]=release(default), debug`
- Action: `-a[ction]=build(default), buildinstall, install, clean`
- WithModule: `-w[ith]=all(default), fdo, providers, shp, sdf, wfs, wms, arcsde, odbc, mysql, gdal`
- BuildDocs: `-d[ocs]=skip(default), build`

NOTE If you give the action option argument a value of `buildinstall` or `install`, you must also provide an `outpath` option argument whose value is not the build folder or one of its subfolders.

The default action is to build all of the FDO core release .dll and .lib files and write them to Fdo subfolders and to build all of the provider release .dll and .lib files and write them to Provider subfolders.

If you only specify the `-c=debug` option, you get all of the FDO core debug .dll and .lib files written to Fdo subfolders and all of the provider debug .dll and .lib files written to Provider subfolders.

If you specify the `-a=buildinstall` option, you get the Fdo core and provider .dll files copied to Fdo\Bin in the folder specified using the `-o` option, .lib files copied to Fdo\Lib in the folder specified using the `-o` option, and the providers.xml file copied to Fdo\Bin in the folder specified using the `-o` option.

NOTE The providers.xml file is used by FDO to identify where FDO provider binaries are installed.

If you specify the `-w=fdo` option, you get FDO .dll and .lib files built and written to Fdo subfolders. If you specified an install option, you get the .dlls copied to Fdo\Bin in the folder specified using the `-o` option, the .lib files copied to Fdo\Lib in the folder specified using the `-o` option, and some .h files copied to Fdo\Inc in the folder specified using the `-o` option.

If you specify the `-w=arcsde` option, you get the `arcsde .dll` files built and written to `Providers\ArcSDE` subfolders. If you specified an install option, you get the `.dlls` copied to `Fdo\Bin` in the folder specified using the `-o` option.

If you specify the `-w=gdal` option, you get the `GRFP* .dll` and `.lib` files built and written to `Providers\GDAL` subfolders. If you specified an install option, you get the `.dll` files copied to `Fdo\Bin` in the folder specified using the `-o` option and the `.lib` file copied to `Fdo\Lib` in the folder specified using the `-o` option.

If you specify the `-w=mysql` option, you get the `mysql .dll` and `.lib` files and the generic `rdbms .dll` and `.lib` files built and written to `Providers\GenericRdbms` subfolders. If you specified an install option, you get the `.dll` files copied to `Fdo\Bin` in the folder specified using the `-o` option, the `.lib` files copied to `Fdo\Lib` in the folder specified using the `-o` option, some `.sql` files copied to `Fdo\Bin\com` in the folder specified using the `-o` option, and some `.h` files copied to `Fdo\Inc` in the folder specified using the `-o` option.

If you specify the `-w=odbc` option, you get the `odbc .dll` and `.lib` files and the generic `rdbms .dll` and `.lib` files built and written to `Providers\GenericRdbms` subfolders. If you specified an install option, you get the `.dll` files copied to `Fdo\Bin` in the folder specified using the `-o` option, the `.lib` files copied to `Fdo\Lib`, in the folder specified using the `-o` option and some `.h` files copied to `Fdo\Inc` in the folder specified using the `-o` option.

If you specify the `-w=sdf` option, you get the `sdf .dll` files built and written to a `Providers\SDF` subfolder. If you specified an install option, you get the `.dll` files copied to `Fdo\Bin` in the folder specified using the `-o` option and some `.h` files copied to `Fdo\Inc\SDF` in the folder specified using the `-o` option.

If you specify the `-w=shp` option, you get the `shp .dll` and `.lib` files built and written to a `Providers\SHP` subfolder. If you specified an install option, you get the `.dll` files copied to `Fdo\Bin` in the folder specified using the `-o` option, the `.lib` file copied to `Fdo\Lib` in the folder specified using the `-o` option and some `.h` files copied to `Fdo\Inc\SHP` in the folder specified using the `-o` option.

If you specify the `-w=wfs` option, you get the `wfs .dll` files built and written to a `Provider\WFS` subfolder and you get the `ows .dll` file built and written to a `Utilities\OWS` subfolder. If you specified an install option, you get the `wfs` and `ows .dll` files copied to `Fdo\Bin` in the folder specified using the `-o` option.

If you specify the `-w=wms` option, you get the `wms .dll` and `.lib` files built and written to a `Provider\WMS` subfolder and an `ows .dll` file built and written to a `Utilities\OWS` subfolder. If you specified an install option, you get the `wms` and `ows .dll` files copied to `Fdo\Bin` in the folder specified using the `-o`

option and some .h files copied to Fdo\Inc in the folder specified using the -o option.

If you specified the -d=build option

- and you are building the thirdparty .dll files, you get no effect.
- and you are building the FDO core, you get the FDO API unmanaged reference .html files built and written to <buildFolderPath>\Fdo\Docs\HTML\FDO_API and the FDO_API.chm files built and written to <buildFolderPath>\Fdo\Docs. You also get the FDO API managed reference .html files built and written to <buildFolderPath>\Fdo\Docs\HTML\FDO_API_managed and the FDO_API_managed.chm built and written to <buildFolderPath>\Fdo\Docs. If you specified an install option, you get FDO API reference .html copied to <outputFolder\Fdo\Docs\HTML\FDO_API, the FDO API managed reference .html files copied to <outputFolder\Fdo\Docs\HTML\FDO_API_managed, the FDO_API.chm, FDO_API_managed.chm, FDG_FDODevGuide.pdf, and the FET_TheEssentialFDOL.pdf copied to <outputFolder>\Fdo\Docs.
- and you are building the ArcSDE provider, you get the ArcSDE API reference .html files built and written to Providers\ArcSDE\Docs\HTML\ArcSDE and the ArcSDE_Provider_API.chm built and written to Providers\ArcSDE\Docs. If you specified an install option, you get the .html files copied to <outputFolder>\Fdo\Docs\HTML\ArcSDE and the .chm file copied to <outputFolder>\Fdo\Docs.
- and you are building the GDAL provider, you get no effect.
- and you are building the MySQL provider, you get the MySQL API reference .html files built and written to Providers\MySQL\Docs\HTML\MySQL and the MySQL_Provider_API.chm built and written to Providers\MySQL\Docs. If you specified an install option, you get the .html files copied to <outputFolder>\Fdo\Docs\HTML\MySQL and the .chm file copied to <outputFolder>\Fdo\Docs.
- and you are building the ODBC provider, you get the ODBC API reference .html files built and written to Providers\ODBC\Docs\HTML\ODBC and the ODBC_Provider_API.chm built and written to Providers\ODBC\Docs. If you specified an install option, you get the .html files copied to <outputFolder>\Fdo\Docs\HTML\ODBC and the .chm file copied to <outputFolder>\Fdo\Docs.
- and you are building the SDF provider, you get the SDF API reference .html files built and written to Providers\SDF\Docs\HTML\SDF and the

SDF_Provider_API.chm built and written to Providers\SDF\Docs. If you specified an install option, you get the .html files copied to <outputFolder>\Fdo\Docs\HTML\SDF and the .chm file copied to <outputFolder>\Fdo\Docs.

- and you are building the SHP provider, you get the SHP API reference .html files built and written to Providers\SHP\Docs\HTML\SHP and the SHP_Provider_API.chm built and written to Providers\SHP\Docs. If you specified an install option, you get the .html files copied to <outputFolder>\Fdo\Docs\HTML\SHP and the .chm file copied to <outputFolder>\Fdo\Docs.
- and you are building the WFS provider, you get the WFS API reference .html files built and written to Providers\WFS\Docs\HTML\WFS and the WFS_Provider_API.chm built and written to Providers\WFS\Docs. If you specified an install option, you get the .html files copied to <outputFolder>\Fdo\Docs\HTML\WFS and the .chm file copied to <outputFolder>\Fdo\Docs.
- and you are building the WMS provider, you get the WMS API reference .html files built and written to Providers\WMS\Docs\HTML\WMS and the WMS_Provider_API.chm built and written to Providers\WMS\Docs. If you specified an install option, you get the .html files copied to <outputFolder>\Fdo\Docs\HTML\WMS and the .chm file copied to <outputFolder>\Fdo\Docs.

Unit Tests

Introduction

Unit test executables for the FDO core components and for the ArcSDE, GDAL, MySQL, ODBC, SDF, SHP and WMS providers are generated during the build of the debug version.

Run the FDO Unit Tests

Do the following in a cmd.exe window:

```
1 cd C:\OpenSource\Fdo\Unmanaged\bin\win32\debug
```

Configure the ArcSDE Server

You must configure the ArcSDE Server before you can execute the unit tests. The tests are actually run on the server. The unit tests can be run against an Oracle database or a Sql Server 2000 database. You must configure the ArcSDE server for running the unit tests against each type of database. Most of the configuration is done by a perl script called SetupTestData.perl. Before running the script on the server, you must edit it to change some hard-coded values and you must set the SDEHOME environment variable. Configuring the server to run tests in the Sql Server 2000 database requires some additional manual steps.

The first step is to copy the C:\OpenSource\Providers\ArcSDE\TestData folder to the machine hosting the ArcSDE server. This folder contains the test data and the SetupTestData.perl script. Alternatively, you can use subversion to checkout the fdoarcsde.osgeo.org repository on the server.

The second step is to install ActiveState perl on the server. This is available from <http://www.activestate.com/ActivePerl>.

The third step is to create two Sql Server databases. If you want to run the tests in Sql Server 2000, you must create a database called 'testmultidb' using the SQL Server Enterprise Manager tool and create a database called 'testsingledb' using the ArcSDE server post installation tool.

To create the 'testmultidb' database, do the following:

- 1 Invoke the Enterprise Manager tool.
- 2 In the explorer pane right-click Console Root\Microsoft SQL Servers\SQL Server Group\<SQL Server Name>\Databases and select New Database... from the pop-up menu.
- 3 Type testmultidb in the text box labeled Name in the tab labeled General in the dialog box labeled Database Properties and click OK.

To create the 'testsingledb' database, do the following:

- 1 Set the SDEHOME environment variable. The value is such that running the command `dir %SDEHOME%\etc` in a cmd.exe window results in a list including giomgr.defs and dbinit.sde. The value should be something like C:\ArcGIS\ArcSDE\sql.exe. You can set this variable permanently by

using the Control Panel, or you can set it temporarily in a cmd.exe window. In the latter case you must then run the post installation tool from the command-line.

- 2 Invoke the post installation tool. The menu path for this tool will be something like Start ► Programs ► ArcGIS ► ArcSDE ► ArcSDE for Microsoft SQL Server Post Installation. The equivalent filesystem path is something like C:\ArcGIS\ArcSDE\sqllexe\tools\ArcSdeSetupSql.exe.
- 3 In the window labeled ArcSDE for SqlServer click the radio button labeled Custom. Click Next.

NOTE The reason for doing a custom install is to disable the authorization step.

- 4 In the window labeled Select ArcSDE Setup Wizard Option uncheck Authorize ArcSDE and leave checked Define Database and SDE User, Repository Setup, and Create ArcSDE Service. Click Next.
- 5 In the window labeled User Information click the radio button labeled SQL Server Authentication. Fill in the text boxes labeled sysadmin user name and sysadmin user password. Click Next.
- 6 In the window labeled Create Spatial Database fill in the text boxes labeled SDE user password and Database name. Click Next.
- 7 In the window labeled ArcSDE configuration files leave the defaults selected. Click Next.
- 8 In the window labeled User Information review the connection information. The values are either defaults or ones that you provided in previous screens. Click Next.
- 9 A dialog appears displaying the text The ArcSDE Repository was successfully completed. Would you like to view the status?. Click No. If you are told that the repository was unsuccessfully completed, try re-executing the operation (click Back to return to the previous screen and click Next. If you are not rewarded with success, you will have to view the status and debug the operation.
- 10 In the window labeled ArcSDE service information review the information that will be used to create the service that listens for requests pertaining to the database that you just created. The service name and service port number values must be unique to the services file. On WINNT the services files is located in C:\WINNT\system32\drivers\etc. Click Next. This will cause the service name and port number to be added to the services file,

and the service to be started. You will be notified by an information dialog that the service was successfully started. Click OK in this dialog, and the final screen of the wizard is displayed. If you get an error message, try re-executing the operation; doing this has been known to correct the problem.

The fourth step is to edit the “HARDCODED SETTINGS” section within the SetupTestData.perl script. If you wish to install test data in an oracle instance, set `$INSTALLORACLE="Y"`. If you wish to install test data in a SQL Server instance, set `$INSTANCESQLSERVER="Y"`. Verify the values of all of the settings; each setting is described within the script.

The fifth step is to run the SetupTestData.perl script from the command-line on the machine hosting the ArcSDE server. This requires that Oracle’s sqlplus.exe and Sql Server’s osql.exe must be in the PATH environment variable so that the script can find them.

NOTE You can install test data against both Oracle and Sql Server simultaneously.

The script will ask you to specify whether you want to install or uninstall test data. The test data in question is contained in the `C:\OpenSource\Providers\ArcSDE\TestData` folder.

The script will ask you to specify a test dataset name. This name is used to create RDBMS test users with names in the form “FDO_<datasetName>_<suffix>.” The test users will own the test data.

NOTE Dataset names may be no longer than five characters.

If you encounter any problems installing test users or test data, you can run the script again and type ‘U’ for Uninstall, when prompted, to undo all of the changes. After you have identified and solved the problem, you can try the install again.

Run the ArcSDE Unit Tests

You initiate test execution from your machine. The tests are executed on the machine hosting the ArcSDE server. The ArcSDE server runs the tests in the Oracle or Sql Server database depending on the command-line arguments that you provided.

Do the following in a cmd.exe window:

```
1 cd c:\OpenSource\Providers\ArcSDE\Src\UnitTest
```

- ```
2 ..\..\bin\win32\debug\UnitTest.exe server=<dnsHostnameOrIPAddr>
port_multi=<port>/tcp port_single=<port>/tcp
sdepwd=<sdeUserPassword> dataset=<datasetName> rdbms=ORACLE

3 ..\..\bin\win32\debug\UnitTest.exe server=<dnsHostnameOrIPAddr>
port_multi=<multiPort>/tcp port_single=<singlePort>/tcp
sdepwd=<sdeUserPassword> dataset=<datasetName> rdbms=SQLSERVER
```

---

**NOTE** The server value refers to the machine hosting the ArcSDE server.

---

**NOTE** The port\_multi and port\_single values are identical when the value of rdbms is ORACLE. The <port> value is the one used by the ArcSDE server for Oracle to listen for connection requests; you can obtain this value from the services file on the machine hosting the ArcSDE server.

---

**NOTE** The port\_multi and port\_single values are NOT identical when the value of rdbms is SQLSERVER. The <multiPort> value is the one used by the ArcSDE server for SQL Server to listen for connection requests to databases created using the Enterprise Manager tool. The <singlePort> value is the one used by the ArcSDE server for SQL Server to listen for connection requests to the database that you created using the post installation tool.

---

**NOTE** The dataset value is the one that you specified when requested by the SetupTestData.perl script.

---

All of the command-line arguments with the exception of one are of the form <key>=<value>. The one exception is an optional list of test suite names. This list can occur before or after the <key>=<value> arguments. If no test suite names are specified, all test suites are executed. The test suite names follow:

- ABasicOpenTests
- BasicArcSDETests
- BasicConnectionTests
- BasicDeleteTests
- BasicInsertTests
- BasicSchemaTests
- BasicTransactionTests
- LockTests

- LongTransactionTests
- RecommitTest
- SelectAggregatesTests
- SelectTests
- SpatialContextTests
- SQLTests
- ThreadingTests

The following UnitTest.exe invocations provide examples of the execution of the ThreadingTests suite in both Oracle and SQL Server.

```

1 cd c:\OpenSource\Providers\ArcSDE\Src\UnitTest

2 ..\..\bin\win32\debug\UnitTest.exe ThreadingTests
 server=<dnsHostnameOrIPAddr> port_multi=<port>/tcp
 port_single=<port>/tcp sdepwd=<sdeUserPassword>
 dataset=<datasetName> rdbms=ORACLE

3 ..\..\bin\win32\debug\UnitTest.exe ThreadingTests
 server=<dnsHostnameOrIPAddr> port_multi=<multiPort>/tcp
 port_single=<singlePort>/tcp sdepwd=<sdeUserPassword>
 dataset=<datasetName> rdbms=SQLSERVER

```

## Run the GDAL Unit Tests

Do the following in a cmd.exe window:

```

1 cd C:\OpenSource\Providers\GDAL\Src\UnitTest

2 ..\..\bin\Win32\Debug\UnitTest.exe

```

---

**NOTE** The GDAL unit tests use relative paths to locate the test data. The test data is located at C:\OpenSource\Providers\GDAL\TestData.

---

## Run the MySQL Unit Tests

Do the following in a cmd.exe window:

- 1 `cd C:\OpenSource\Providers\GenericRdbms\Src\UnitTest`
- 2 `copy /A MySqlInit.txt MySqlInitEx.txt`
- 3 **NOTE** The contents of this file is  
`"provider=MySQL;service=mysqlserver;username=root;password=xxxx;clean=true;"`
- 4 Edit `MySqlInitEx.txt`. Replace `"service=mysqlserver"` with `"service=<hostname>"` where `<hostname>` is the symbolic name of the IP address of the machine hosting the MySQL server. Replace `"password=xxxx"` with `"password=<rootPassword>"`.
- 5 `Dbg\UnitTestMySQL.exe initfiletest=MySqlInitEx.txt`

## Configure the ODBC Unit Tests

Before running the tests you must do the following:

- Connect to the Oracle database that you will use for the Oracle tests using the sys as sysdba user, create a user and grant the user two roles: dba and wm\_admin\_role.
  - Install a MySQL ODBC driver. Go to <http://dev.mysql.com/downloads/connector/odbc/3.51.html>, download the Windows driver installer (MSI), and run the installer. For this exercise a file named `mysql-connector-odbc-3.51.12-win32.msi` was downloaded.
  - Open Sql Server 2000 Enterprise Manager. Create a new login for the target Sql Server. Use SQL Server authentication and grant the new user the "Database Creators" role.
- `cmd> cd C:\OpenSource\Providers\GenericRdbms\Src\UnitTest`
  - `cmd> copy /A OdbcInit.txt OdbcInitEx.txt`
  - Edit `OdbcInitEx.txt`. Replace the `serviceMySQL` and `passwordMySQL` values in `"serviceMySQL=mysqlserver;usernameMySQL=root;passwordMySQL=xxxx;"` with valid values. The `usernameMySQL` value must remain as root.

- Edit OdbcInitEx.txt. Replace the serviceOracle, usernameOracle and passwordOracle values in  
"serviceOracle=oraserver;usernameOracle=xxxx;passwordOracle=xxxx;"  
with the actual service name, the name of the user that you created and the password that you assigned to that user.
- Edit OdbcInitEx.txt. Replace the serviceSqlServer, usernameSqlServer and passwordSqlServer values in  
"serviceSqlServer=sqlserver;usernameSqlServer=xxxx;passwordSqlServer=xxxx;"  
with the actual service name, the name of the user that you created and the password that you assigned to that user.

## Run the ODBC Unit Tests

You can run all of the ODBC unit tests at once or run them individually.

To run all of the tests, do the following in a cmd.exe window:

```
1 cd C:\OpenSource\Providers\GenericRdbms\Src\UnitTest
2 Dbg\UnitTestOdbc.exe initfiletest=OdbcInitEx.txt
```

To run the tests individually, do the following in a cmd.exe window:

```
1 cd C:\OpenSource\Providers\GenericRdbms\Src\UnitTest
2 Dbg\UnitTestOdbc.exe OdbcTextTests
3 Dbg\UnitTestOdbc.exe OdbcExcelTests
4 Dbg\UnitTestOdbc.exe OdbcAccessTests
5 Dbg\UnitTestOdbc.exe OdbcMySQLTests initfiletest=OdbcInitEx.txt
6 Dbg\UnitTestOdbc.exe OdbcOdbcTests initfiletest=OdbcInitEx.txt
7 Dbg\UnitTestOdbc.exe OdbcSqlServerTests
 initfiletest=OdbcInitEx.txt
```

## Run the SDF Unit Tests

Do the following in a cmd.exe window:

- 1 `cd C:\OpenSource\Providers\SDF\Src\UnitTest`
- 2 `..\..\bin\Win32\Debug\UnitTest.exe`

---

**NOTE** The SDF unit tests use relative paths to locate the test data. The test data is located at C:\OpenSource\Providers\SDF\TestData.

---

**NOTE** Running UnitTest.exe causes the PARCEL\_linux.test.SDX to be modified and creates some files. Before running the UnitTest.exe again, do a revert on PARCL\_linux.test.SDX and delete the files created by the UnitTest.exe.

---

## Run the SHP Unit Tests

Do the following in a cmd.exe window:

- 1 `cd C:\OpenSource\Providers\SHP\Src\UnitTest`
- 2 `..\..\Bin\Win32\Debug\UnitTest.exe`

---

**NOTE** The SHP unit tests use relative paths to locate the test data. The test data is located at C:\OpenSource\Providers\SHP\TestData.

---

**NOTE** If you experience any failures, you might try executing `C:\OpenSource\Providers\SHP\TestData\clean` in a cmd.exe window and retry executing the unit tests.

---

## Run the WMS Unit Tests

Do the following in a cmd.exe window:

- 1 `cd c:\OpenSource\Providers\WMS\Bin\Win32\Debug`
- 2 `UnitTest.exe`

---

**NOTE** These tests involve connections to websites. If a website is not available, the related tests fail.

---





# Building FDO on Linux

# 2

## In this chapter

- [Introduction](#)
- [Disk Space](#)
- [Install the Repository Tool](#)
- [Get the Source](#)
- [Get the Source](#)
- [Review the Copyright and Licensing Information](#)
- [Updating the Source](#)
- [Install the Build Tools](#)
- [Install RDBMS Clients](#)
- [Environment Variables](#)
- [Building FDO](#)
- [Unit Tests](#)

## Introduction

This chapter describes:

- how to build the Feature Data Object binaries from the source files and install them on a Linux machine
- the build tools that you must install to do the build
- the database clients that you must install to do the build

---

**NOTE** The debug binaries built by this process include the unit test executables.

---

**NOTE** The Linux system documented here is Red Hat ES 3.0.

---

A briefer description of the build process is in `OpenSourceBuild_README.txt` file in the build directory.

Whether you intend to build applications on top of FDO or modify the FDO code itself, you must build the binaries.

## Disk Space

The worse case requirement for build and install disk space occurs if you build both the debug and release versions and install the debug version. In this case you require 2 GB of disk space. This requirement is exclusive of the disk space required for the Subversion client including dependencies (6.1 MB), the build tools (12 MB) and the ArcSDE and MySQL provider clients and ODBC driver (115MB). The final total is 2.13 GB.

## Install the Repository Tool

The FDO code is stored in Subversion repositories on the Open Source Geospatial Foundation website. Use a Subversion client to obtain the FDO source from these repositories. To browse the list of available clients, click the Get help? link at the left-hand side of the <https://www.osgeo.org> page. A new browser window is displayed; click the About source code version control with Subversion link. Follow the instructions there to obtain and install a client. The instructions in this document are based on the use of the SVN command-line client.

For the purpose of this document, the following RPMs from <http://the.earth.li/pub/subversion/summersoft.fay.ar.us/pub/subversion/latest/rhel-3/bin/> were installed.

- subversion-1.3.1-1.rhel3.i386.rpm
- neon-0.24.7-1.i386.rpm
- mod\_dav\_svn-1.3.1-1.rhel3.i386.rpm

---

**NOTE** mod\_dav\_svn-1.3.1-1 requires a version of httpd  $\geq$  2.0.46. The required version of httpd was already present on the test machine.

---

---

**NOTE** The rpm install puts the svn binary in /usr/bin

---

## Get the Source

The FDO source code is located on the Open Source Geospatial Foundation website. You must first become a registered user of the site and then login before you can download source from the site. Go to <https://www.osgeo.org>. In the upper right-hand corner there is a link labeled Register. Click on this link and follow the instructions for registering and logging in. Once you have logged in, you may get the source by either downloading a gzipped tar file or doing a “checkout” from a set of Subversion repositories.

---

**NOTE** The “checkout” is read-only.

---

## Get the Source

### Subversion Repositories

The SVN client enables you to download files from a Subversion repository into a build directory. There is one repository for the utilities, fdo core and thirdparty components and one repository for each of the providers with the exception of the MySQL and ODBC providers, which share a repository. A URL identifies the repository and a local path identifies the directory which receives the downloaded files. In this document the build directory is called /home/OpenSource.

To get the source from the repositories execute the following commands in a terminal window.

---

**NOTE** You will be prompted for a password.

---

**NOTE** The parallelism between the third and fourth arguments of the svn command, for example, between "...trunk/Providers/ArcSDE" and "/home/OpenSource/Providers/ArcSDE."

---

**NOTE** The fourth argument can be an absolute or relative path.

---

- 1 `svn checkout https://fdocore.osgeo.org/svn/fdocore/trunk  
/home/OpenSource --username yourusername`
- 2 `svn checkout  
https://fdoarcsde.osgeo.org/svn/fdoarcsde/trunk/Providers/ArcSDE  
/home/OpenSource/Providers/ArcSDE --username yourusername`
- 3 `svn checkout  
https://fdogdal.osgeo.org/svn/fdogdal/trunk/Providers/GDAL  
/home/OpenSource/Providers/GDAL --username yourusername`
- 4 `svn checkout  
https://fdordbms.osgeo.org/svn/fdordbms/trunk/Providers/GenericRdbms  
/home/OpenSource/Providers/GenericRdbms --username yourusername`
- 5 `svn checkout  
https://fdosdf.osgeo.org/svn/fdosdf/trunk/Providers/SDF  
/home/OpenSource/Providers/SDF --username yourusername`
- 6 `svn checkout  
https://fdoshp.osgeo.org/svn/fdoshp/trunk/Providers/SHP  
/home/OpenSource/Providers/SHP --username yourusername`
- 7 `svn checkout  
https://fdowfs.osgeo.org/svn/fdowfs/trunk/Providers/WFS  
/home/OpenSource/Providers/WFS --username yourusername`
- 8 `svn checkout  
https://fdowms.osgeo.org/svn/fdowms/trunk/Providers/WMS  
/home/OpenSource/Providers/WMS --username yourusername`

---

**NOTE** The fdocore components includes a script called `checkoutsvn.sh`, which can be used to get updates for the all of the components from the Subversion repositories.

---

## Review the Copyright and Licensing Information

The copyright and licensing information for the FDO API is contained in the License\_README.txt file in the build folder.

## Updating the Source

You can update the source by running the checkoutsvn.sh script.

In a terminal window where the current directory is the build directory, enter checkoutsvn -h. The help text displayed is as follows:

- checkoutsvn.sh [--h]
- [--o OutFolder]
- [--w WithModule]
- --s Source
- --u UserId
- --p UserPassword
- Help: --h[elp]
- Source: --s[source]=location of source file, either branch location, tag location or trunk

---

**NOTE** The possible arguments are --s trunk, --s <tagIdentifier>, and --s <branchIdentifier>. Get the tag or branch identifier by browsing <http://fdo.osgeo.org>, clicking the Subversion link and then clicking the tag or branches link.

---

- OutFolder: --o[utpath]=destination folder for files
- WithModule: --w[ith]=all(default) fdo thirdparty providers utilities shp sdf wfs wms arcsde rdbms gdal
- User: --u[ser]=user id
- Password: --p[assword]=user password

You can update all of the source by running the `checkoutsvn.sh` script in the build directory. The syntax is `./checkoutsvn.sh --o /home/OpenSource --s <source> --u yourusername --p yourpassword`.

You can update the source for specified components by using the `with` option. The syntax is `./checkoutsvn.sh --o /home/OpenSource --s <source> --u yourusername --p yourpassword --w desiredComponent`.

The possible arguments for the `with` option are:

- `all` - updates all directories
- `providers` - updates the ArcSDE, GDAL, GenericRdbms, SDF, SHP, WFS, and WMS directories
- `fdocore` - updates the Fdo, Thirdparty, and Utilities directories
- `thirdparty` - updates the Thirdparty directory
- `fdo` - updates the Fdo directory
- `utilities` - updates the Utilities directory
- `arcsde` - updates the ArcSDE directory
- `gdal` - updates the GDAL directory
- `rdbms` - updates the GenericRdbms directory, which contains the MySQL and ODBC components
- `sdf` - updates the SDF directory
- `shp` - updates the SHP directory
- `wfs` - updates the WFS directory
- `wms` - updates the WMS directory

# Install the Build Tools

## Binary Build Tools

The following software is used during the process of building the FDO binaries.

| Tool      | Description                                                                                                                                     | Path     | Download from |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------------|
| auto-conf | produces shell scripts to configure software source code packages automatically; depends on GNU m4 (version 1.4 or greater)                     | /usr/bin | Part of RHEL3 |
| auto-make | generates makefile.in files from makefile.am input files                                                                                        | /usr/bin | Part of RHEL3 |
| make      | controls the generation of executables and other non-source files of a program from the program's source files                                  | /usr/bin | Part of RHEL3 |
| bison     | a general purpose parser generator that converts a grammar description for an LALR context-free grammar into a C program to parse that grammar. | /usr/bin | Part of RHEL3 |
| sed       | (streams editor) is a text filter tool. It takes text input, performs one or more operations on the text and outputs the modified text.         | /bin     | Part of RHEL3 |
| perl      | scripting language. This is used to build the FDO third-party OpenSSL components and to run the ArcSDE unit tests.                              | /usr/bin | Part of RHEL3 |

**NOTE** The FDO build requires the use of bison version 1.875. This happens to be the installed version on the Red Hat Linux machine used during the writing of this document. You can obtain an rpm for version 1.875-5 from <http://rpmfind.net/linux/RPM/fedora/1/i386/bison-1.875-5.i386.html>.

## Documentation Build Tools

The following software is used during the process of building the FDO documentation.

| Tool     | Description                                                                                                                       | Path           | Download from |
|----------|-----------------------------------------------------------------------------------------------------------------------------------|----------------|---------------|
| doxygen  | generates API documentation from specially formatted comments embedded in C++, C, Java, Objective-C, Python, IDL, PHP, C#, and D. | /usr/local/bin | Part of RHEL3 |
| graphviz | is a graph drawing toolkit used by doxygen to draw class diagrams. The graphviz executable used by doxygen is dot.exe.            | /usr/bin       | Part of RHEL3 |

## Install RDBMS Clients

Building the ArcSDE, MySQL, and ODBC providers is dependent on client libraries being available.

## ArcSDE 9.1 Client SDK

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/arcade/how-to-buy.html>.

Install the client to a location outside of the build directory. The build script uses the SDEHOME environment variable to find the client headers and libraries. For more information about the SDEHOME variable, read the section entitled “Environment Variables.”

The list of ArcSDE client files follows.

- etc/services.sde
- include/pe.h
- include/pedef.h
- include/pef.h



- include/pe\_coordsys\_from\_prj.h
- include/sg.h
- include/sgerr.h
- include/sdeerno.h
- include/sderaster.h
- include/sdetype.h
- lib/libedgmt.so
- lib/libgsrvrdb291.so
- lib/libgsrvrora9i91.so
- lib/libicudata.so.22.0
- lib/libicuuc.so.22.0
- lib/libloceng.so
- lib/liblocssa.so
- lib/libmtchloc.so
- lib/libmtchmt.so
- lib/libpe91.a
- lib/libpe91.so
- lib/libsde91.a
- lib/libsde91.so
- lib/libsdedb2srvr91.so
- lib/libseora9isrvr91.so
- lib/libsg91.a
- lib/libsg91.so
- lib/libxerces-c.so.21.0

## MySQL Client

Download a Linux x86 RPM package containing the latest production release of MySQL 5.0 from <http://dev.mysql.com/downloads/mysql/5.0.html>. The builds done during the writing of this document used the 5.0.22 version.

## ODBC Oracle Driver

Download version 3.0 of the Easysoft ODBC Oracle driver. This driver works with both the 10gR1 database client or instant client. The install package includes a version of the Unix ODBC Driver Manager. For more information about the driver manager software, browse <http://www.unixodbc.org>.

- 1 Browse <http://www.easysoft.com>.
- 2 Click Products.
- 3 Scroll. Click Easysoft ODBC-Oracle driver.
- 4 Login.
- 5 Select Linux - x86 - (32-bit) from the platform list box.
- 6 Click Download. This action initiates the download of `odbc-oracle-3.0.0-linux-x86-glibc.tar`.

Untar the download file. Untarring `odbc-oracle-3.0.0-linux-x86-glibc.tar` causes the contents of the tar file to be written to a directory called `odbc-oracle-3.0.0-linux-x86-glibc`. Read the `INSTALL.txt` file. This document recommends that you do the installation as root so that you can install to `/usr/local`. It also recommends that you install the version of the unixODBC driver manager that comes bundled with the Easysoft ODBC-Oracle driver installation package. The document explains what options were used to build this version of the driver manager and thus how it differs in behavior from what you will get if you install a version from <http://www.unixODBC.org>. The document explains how to install the licensed version of the driver.

The installation described here followed the recommendations to install as root to `/usr/local` and to use the bundled unixODBC driver manager. It also installed a licensed version using the license script called by the installer script. This is optional. You have the option of leaving the step of acquiring a license to a later time.

---

**NOTE** The unlicensed version does not work with Oracle 10g databases.

---

In a terminal window do the following:

Before you begin, verify that there is no previous easysoft installation. Check /etc for the odbc.ini and odbcinst.ini files. Check the /etc/ld.so.conf file for easysoft paths. Check for an easysoft directory in /usr/local. The latter is the default install location.

1 su - root

2 export ORACLE\_HOME=<pathToOracleClient>

---

**NOTE** The target RHEL3.0 machine has a 10gR1 full client installed in /app/oracle/OraHome\_1. So for this exercise <pathToOracleClient> is /app/oracle/OraHome\_1. **DO NOT** use the Oracle instant client.

---

3 cd <path>/odbc-oracle-3.0.0-linux-x86-glibc

4 ./install

5 Read the license and type yes to accept it.

6 Press enter to accept the default location (/usr/local).

7 Press enter to accept the default no answer to the question about there being a unixODBC driver manager already installed.

8 Press enter to accept the yes answer to the question about requesting an Easysoft ODBC-Oracle Driver License now. This causes the license script to run. It runs inline with the installer script.

9 Type 2 to initiate the acquisition of a license for the driver.

10 Supply values for your name, company, email address, phone number, fax number, and license authorization number.

11 Select the option to obtain the license automatically from the Easysoft license daemon.

12 Select the option to exit from licensing script and return to the installer script. The installer script tells you that it will create a data source called 'ORACLE.'

13 Enter the Oracle Database Name. Since we connect to the database by way of a client, we specify the Net Service Name. This is one of the names found in the \$ORACLE\_HOME/network/admin/tnsnames.ora file. This name indexes a structure containing the hostname, port, and database service name of the instance to connect to.

- 14 Enter an Oracle user name. This name must be a valid user in the Oracle instance.
- 15 Enter the Oracle password for <user> where <user> is value you just entered in the previous step.
- 16 Press enter to answer yes to the question about installing the driver data source. This completes the installation.
- 17 

```
cd /usr/local/easysoft/unixODBC/bin
```
- 18 

```
./isql -v ORACLE
```

 You should get a message saying 'Connected' and the SQL prompt.

---

**NOTE** 'ORACLE' is the name of the data source created by the installer. The information about the data source is in the /etc/odbc.ini file.

---

---

**NOTE** The build script uses the FDOODBC environment variable to locate the ODBC libraries. The script expects to find a lib directory in the path value contained in FDOODBC. The FDOODBC variable is set by the setenvironment.sh script; for more information, read the section entitled "Environment Variables."

---

As a result of the installation the odbc.ini and odbcinst.ini files are written to /etc. The odbcinst.ini identifies the shared object files for the driver, and the odbc.ini contains mapping of the Data Source Name (DSN) to the Oracle database. You will need this DSN to run the unit tests.

## Environment Variables

Use the setenvironment.sh script in the build directory to set the environment variables before commencing the build. The script assigns default values to most of these variables as noted below and tests for the existence of the directories. If the script detects that a directory does not exist, it will request that you modify the script. The default locations for the MySQL and ArcSDE clients are inside the build directory hierarchy. You change the default values of FDOMYSQL, SDEHOME, FDOODBC, and LD\_LIBRARY\_PATH.

---

**NOTE** Source the script (`source setenvironment.sh`), so that the environment variables are set in the parent shell.

---

---

**NOTE** Add the following line to the `setenvironment.sh` script before running it:  
`export FDOODBC=<directory>` where `<directory>` is the location of the lib and include directories for the ODBC Driver Manager.

---

- FDO (default is `$PWD/Fdo`)
- FDOMYSQL (default is `$FDOTHIRDPARTY/mysql/rhlinux`). This is optional. If you installed MySQL RPMs, you will find the libs in `/usr/lib/mysql` and the includes in `/usr/include/mysql`. The build script will find them.
- FDOTHIRDPARTY (default is `$PWD/Thirdparty`)
- FDOUTILITIES (default is `$PWD/Utilities`)
- SDEHOME (default is `$FDOTHIRDPARTY/ESRI/ArcSDEClient91/Linux`). Set it to the location of the ArcSDE client lib and include directories.
- FDOODBC Set it to the location of the unixODBC driver manager's include and libraries files. Typically, this is `/usr/local/easysoft/unixODBC`.
- LD\_LIBRARY\_PATH. Adds `/usr/local/fdo-3.2.0/lib`. Adds `$SDEHOME/lib` so that the ArcSDE UnitTest binary can find the ArcSDE client shared objects.

## Building FDO

### Build Scripts

The build folder contains three bat files:

- `build_thirdparty.sh` - builds and installs third-party DLLs used by the providers
- `build_parse.sh` - recreates the FDO filter and expression language grammar source files from yacc files
- `build_linux.sh` - builds and installs the fdo core and provider DLLs

## Build Order

Build the thirdparty components first. Optionally generate the FDO filter and expression language grammar source files. Build the FDO and utility components. Finally build the providers.

## Run the build\_thirdparty.sh Script

The assumption is that you are not changing the third-party software. So these binaries need only be built once. If you notice a third-party component has been updated during a run of checkoutsvn.sh, you must run this script again.

In a terminal window where you ran the setenvironment.sh script, enter  
`./build_thirdparty.sh --h.`

- `build_thirdparty.sh [--h] [--a Action] [--m ConfigMakefiles]`
- Help: `--h[elp]`
- Action: `--a[ction]` `buildinstall`(default), `build`, `install`, `uninstall`, `clean`
- ConfigMakefiles: `--m[akefile]` `configure`(default), `noconfigure`

The default action is to configure the makefiles, do the make and install the software in the distribution directory. The `--m noconfigure` option can be used to skip the configuring of the makefiles.

The following files are installed to `/usr/local/fdo-3.2.0/lib`:

- `libxalan-c.so.17.0`
- `libxalanMsg.so.17.0`
- `libxerces-c.so.25.0`
- `libgdal.so.1.9.0`

## Optionally Run the build\_parse.sh Script

The FDO build uses several source (`.cpp` and `.h`) files generated from `.y` files by the Bison and Sed utilities. These `.y` files define a grammar for the FDO expression and filter language. If you change the `.y` files, you must run the `build_parse.sh` script to regenerate the source files.

## Run the build\_linux.sh Script

In a terminal window enter `cd /home/OpenSource` and then enter `build --h`. The build help text is contained in the following list:

- `build_linux.sh [--h] [--c BuildType] [--a Action] [--w WithModule] [--d BuildDocs] [--m ConfigMakefiles]`
- Help: `--h[elp]`
- BuildType: `--c[onfig] release(default), debug`
- Action: `--a[ction] buildinstall(default), build, install, uninstall, clean`
- ConfigMakefiles: `--m[akefile] configure (default), noconfigure`
- WithModule: `--w[ith] all (default), fdocore, fdo, utilities, providers, shp, sdf, wfs, wms, arcsde, rdbms, gdal`
- BuildDocs: `--d[ocs] skip(default), build`

---

**NOTE** The `--c debug` option causes the unit test executables to be built. The `--m noconfigure` option causes the step of configuring the makefiles to be skipped. The `--w fdocore` option is the same as `--w fdo` and `--w utilities` options combined.

---

The `--w fdo` option together with the `--a buildinstall` option causes `libFDO-3.2.0.so` and `libFDO.la` to be written to `/usr/local/fdo-3.2.0/lib`. Xml schema files are written to `/usr/local/fdo-3.2.0/Docs/XmlSchema`, Four header files, `FdoCommon.h`, `FdoGeometry.h`, `Fdo.h`, and `FdoMessage.h`, and three directories, `Common`, `Fdo`, and `Geometry`, are written to `/usr/local/fdo-3.2.0/include`. `FDOMessage.cat` to be written to `/usr/local/fdo-3.2.0/nls`. If you add the `--d build` option, you will get the API documentation generated and written to `/usr/local/fdo-3.2.0/docs/HTML/FDO_API`.

The `--w utilities` option together with the `--a buildinstall` option causes `libFdoOws-3.2.0.so` and `libFdoOws.la` to be written to `/usr/local/fdo-3.2.0/lib`. There is no documentation generated for utilities.

The `--w arcsde` option together with the `--a buildinstall` option causes `libArcSDEProvider-3.2.0.so` and `libArcSDEProvider.la` to be written to `/usr/local/fdo-3.2.0/lib`. The `ArcSDE` directory is created in `/usr/local/fdo-3.2.0/include`. There is no documentation generated for `ArcSDE`.

The `--w gdal` option together with the `--a buildinstall` option causes `libGRFPOverrides-3.2.0.so`, `libGRFPOverrides.la`, `libGRFPProvider-3.2.0.so`, and

libGRFPPProvider.la to be written to /usr/local/fdo-3.2.0/lib. The GdalFile directory is created in /usr/local/fdo-3.2.0/include. There is no documentation generated for GDAL.

The --w sdf option together with the --a buildinstall option causes libSDFProvider-3.2.0.so and libSDFProvider.la to be written to /usr/local/fdo-3.2.0/lib. The SDF directory is created in /usr/local/fdo-3.2.0/include. There is no documentation generated for SDF.

The --w shp option together with the --a buildinstall option causes libSHPOverrides-3.2.0.so, libSHPOverrides.la, libSHPPProvider-3.2.0.so, and libSHPPProvider.la to be written to /usr/local/fdo-3.2.0/lib. The SHP directory is created in /usr/local/fdo-3.2.0/include. There is no documentation generated for SHP.

The --w wfs option together with the --a buildinstall option causes libWFSPProvider-3.2.0.so, and libWFSPProvider.la to be written to /usr/local/fdo-3.2.0/lib. There are no header files written to /usr/local/fdo-3.2.0/include. There is no documentation generated for WFS.

The --w wms option together with the --a buildinstall option causes libWMSOverrides-3.2.0.so, libWMSOverrides.la, libWMSPProvider-3.2.0.so, and libWMSPProvider.la to be written to /usr/local/fdo-3.2.0/lib. The WMS directory is created in /usr/local/fdo-3.2.0/include. There is no documentation generated for WMS.

The --w rdbms option together with the --a buildinstall option causes libFdoMySQL-3.2.0.so, libFdoMySQL.la, libFdoODBC-3.2.0.so, libFdoODBC.la, libSchemaMgr\_OV-3.2.0.so, and libSchemaMgr\_OV.la to be written to /usr/local/fdo-3.2.0/lib. The directory Rdbms is created in /usr/local/fdo-3.2.0/include. There is no documentation generated for MySQL or ODBC.

Configure and remove the intermediate files produced during the last build using the command `./build_linux.sh -a clean`

Configure and uninstall the current distribution, leaving the documentation intact, using the command `./build_linux.sh -a uninstall`

Configure, build and install the debug binaries using the command `./build_linux.sh -c debug`

Configure and build the release binaries component by component using the following commands.

- `./build_linux.sh --a build --w fdo`
- `./build_linux.sh --a build --w utilities`



```
■ ./build_linux.sh --a build --w arcsde
■ ./build_linux.sh --a build --w sdf
■ ./build_linux.sh --a build --w shp
■ ./build_linux.sh --a build --w wfs
■ ./build_linux.sh --a build --w wms
■ ./build_linux.sh --a build --w rdbms
■ ./build_linux.sh --a build --w gdal
```

Build and install the API documentation and install the release binaries using the following commands.

```
■ ./build_linux.sh --a install --w fdo --d build
■ ./build_linux.sh --a install --w providers --d build
```

Remove intermediate files from the build directories and uninstall the binaries and include files from the output directories using the following commands.

```
■ ./build_linux.sh --a clean --w fdo
■ ./build_linux.sh --a clean --w utilities
■ ./build_linux.sh --a clean --w providers
■ ./build_linux.sh --a uninstall --w fdo
■ ./build_linux.sh --a uninstall --w providers
■ ./build_linux.sh --a uninstall --w utilities
```

---

**NOTE** The uninstall commands do not delete the API documentation in the docs directory.

---

Configure, build, and install the debug binaries using the following commands.

```
■ ./build_linux.sh --c debug --m noconfigure --w fdo
■ ./build_linux.sh --c debug --m noconfigure --w utilities
■ ./build_linux.sh --c debug --w providers
```

# Unit Tests

## Introduction

Unit test executables for the Fdo core components and for the ArcSDE, GDAL, MySQL, ODBC, SDE, SHP, and WMS providers are generated during the build of the debug version.

## Run the FDO Unit Tests

Do the following in a terminal window:

```
1 cd /home/OpenSource/Fdo/UnitTest
2 ./UnitTest
```

## Run the ArcSDE Unit Tests

You must configure the ArcSDE server before you can run the unit tests. This procedure is described in the topic [Configure the ArcSDE Server](#) (page 18).

You must add \$SDEHOME/lib to the LD\_LIBRARY\_PATH environment variable so that the UnitTest binary can find the ArcSDE client libraries.

The parameters in the following commands are more fully explained in the Windows build topic [Run the ArcSDE Unit Tests](#) (page 20).

```
1 cd /home/OpenSource/Providers/ArcSDE/Src/UnitTest

2 ./UnitTest server=<dnsHostnameOrIPAddr> port_multi=<port>/tcp
 port_single=<port>/tcp sdepwd=<sdeUserPassword>
 dataset=<datasetName> rdbms=ORACLE

3 ./UnitTest server=<dnsHostnameOrIPAddr>
 port_multi=<multiPort>/tcp port_single=<singlePort>/tcp
 sdepwd=<sdeUserPassword> dataset=<datasetName> rdbms=SQLSERVER

4 ./UnitTest ThreadingTests server=<dnsHostnameOrIPAddr>
 port_multi=<port>/tcp port_single=<port>/tcp
 sdepwd=<sdeUserPassword> dataset=<datasetName> rdbms=ORACLE
```

```
5 ./UnitTest ThreadingTests server=<dnsHostnameOrIPAddr>
port_multi=<multiPort>/tcp port_single=<singlePort>/tcp
sdepwd=<sdeUserPassword> dataset=<datasetName> rdbms=SQLSERVER
```

## Run the GDAL Unit Tests

Do the following in a terminal window:

```
1 cd /home/OpenSource/Providers/GDAL/Src/UnitTest
2 ./UnitTest
```

## Run the MySQL Unit Tests

Do the following in a terminal window:

```
1 cd /home/OpenSource/Providers/GenericRdbms/Src/UnitTest
2 cp MySqlInit.txt MySqlInitEx.txt
```

---

```
3 NOTE The contents of this file is
"provider=MySQL;service=mysqlserver;username=root;password=xxxx;clean=true;"
```

---

```
4 Edit MySqlInitEx.txt. Replace "service=mysqlserver" with
"service=<hostname>" where <hostname> is the symbolic name of the
IP address of the machine hosting the MySQL server. Replace
"password=xxxx" with "password=<rootPassword>".
5 Dbg/UnitTestMySQL.exe initfiletest=MySqlInitEx.txt
```

## Run the ODBC Unit Tests

There are two sets of tests. One set for Oracle, and one set for MySQL. Do the following in a terminal window:

```
1 cd /home/OpenSource/Providers/GenericRdbms/Src/UnitTest
2 cp OdbcInit.txt OdbcInitEx.txt
```

---

**3 NOTE** The contents of this file is

```
provider=Odbc;clean=false;
serviceOracle=oraserver;usernameOracle=xxxx;passwordOracle=xxxx;DSNOracle=xxxx;enableOracleSetup=false;
serviceMySql=mysqlserver;usernameMySql=root;password=xxxx;DSNMySql=xxxx;
serviceSqlServer=sqlserver;usernameSqlServer=xxxx;passwordSqlServer=xxxx;DSNSqlServer=xxxx;
```

---

- 4** Edit OdbcInitEx.txt. Replace “serviceOracle=oraserver” with “service=<netServiceName>” where <netServiceName> is the name of the entry in the \$ORACLE\_HOME/network/admin/tnsnames.ora file. Replace “xxxx” in “usernameOracle=xxxx” with the name of a user that you created in the Oracle database that has been assigned the F\_USER\_ROLE role. Replace “password=xxxx” with “password=<userPassword>”. Replace the “xxxx” in “DSNOracle=xxxx” with the Data Source Name in the /etc/odbc.ini file created during the install of the Easysoft ODBC Oracle driver. By default, that value is ORACLE.

- 5** `unix> ./UnitTestOdbc initfiletest=OdbcInitEx.txt OdbcOracleTests`

## Run the SDF Unit Tests

Do the following in a cmd.exe window:

- 1** `cd /home/OpenSource/Providers/SDF/Src/UnitTest`
- 2** `./UnitTest`

## Run the SHP Unit Tests

Do the following in a cmd.exe window:

- 1** `cd /home/OpenSource/Providers/SHP/Src/UnitTest`
- 2** `./UnitTest`

## Run the WMS Unit Tests

Do the following in a cmd.exe window:

- 1 `cd /home/OpenSource/Providers/WMS/Src/UnitTest`
- 2 `./UnitTest`



# Index

