

# ArcSDE Plugin

Module Maintainer:	<a href="#">Gabriel Roldán</a>
Status:	★★★★★
Email Help:	Geotools-gt2-users@lists.sourceforge.net
Volunteer:	geotools-devel@lists.sourceforge.net
Plugin:	DataStore

Plug-in provides access to an ArcSDE server for both vector and raster data.

- [Module Status](#)
  - [Gold Star Quality Assurance Check](#)
  - [Recent Development \(as of May 2009\)](#)
  - [IP Review](#)
  - [Usage](#)
  - [Outstanding Issues](#)
- [Development](#)
  - [Design and Research Pages](#)
    - [ArcSDE Installation Notes on Windows](#)
  - [Building a Working ArcSDE using the jsde jars](#)
    - [So what do I have to type again?](#)
    - [Running the Tests Against your Own Server](#)

## Module Status

- ✓ The arcsde module is stable for data reading for both vector and raster data.
- ✓ The arcsde module is stable for transactional vector data modification. Does not support raster data modification.
- ✓ The arcsde module is currently a part of the nightly build
- ✗ The test suite is not being run as a part of the nightly build as there is no build profile which would connect it to a live SDE instance to perform tests.

## Gold Star Quality Assurance Check

There are five 'support stars' to earn. In order to get the ArcSDE module to supported status, we need at least four, and eventually five.

status	description
★	IP check, see <a href="#">review.apt</a>
★	Releasable - has no blocking bugs in jira

★	Used in anger - Used by GeoServer or uDig on large real world datasets – used by MassGIS to serve the statewide landuse dataset.
★	Optimized - has been tuned to meet hard performance requirements (example shapefile)
★	Supported - user docs, module maintainer watches user list, answers email etc.. Javadocs could use a review.

## Recent Development (as of May 2009)

For the [2.6.x](#) branch (trunk) arcsde raster support has been completely refurbished in order to support all the ArcSDE raster pixel depths, and any number of bands (with the exception of 4-bit rasters and 1-bit ones with more than one band), as well as colormapped rasters (8 or 16-bit rasters with a color palette).

The vector support was also enhanced to perform really well under high concurrency.

All the changes in the 2.6.x series are being backported to the stable [2.5.x](#) branch.

For the [2.5.x](#) branch arcsde has been almost completely rewritten, it no longer extends AbstractDatatore and the code is **much** easier to follow. Internally it protects all SeConnection use behind a Queue (rather than a Lock) as a way of addressing concurrent usage in a fair manner.

Also, it has been extended to support the read and edition of versioned tables, limited to edit the default version.

For the [2.4.x](#) branch arcsde is working on returning to supported status; we need to provide a profile for nightly builds to run. ArcSDE Raster Support is also available on 2.4.x as well. You can find out about it at the [ArcSDE Raster Support](#) page.

For the [2.3.x](#) branch, there is an initial implementation of ArcSDE Raster Support. You can find out about it [here](#).

For the [2.2.x](#) branch the arcsde module has returned to the build process with a stub jar to replace ESRI dependencies.

## IP Review

Please refer to:

- <http://svn.osgeo.org/geotools/trunk/modules/plugin/arcsde/datastore/src/site/apt/review.apt>

## Usage

For usage instructions please consult the [ArcSDE Plugin](#) page on the GeoTools User Guide.

## Outstanding Issues

**Error rendering macro 'jiraissues' : Unable to determine if sort should be enabled.**

## Development

ArcSDE used to have two great module maintainers looking after it: Saul Farber and Gabriel Roldan. As per GeoTools 2.6.x Saul dropped from being a co-maintainer but still watches the mailing lists and provides great feedback. Let's thank him for the great work during the years in implementing the basis of the current ArcSDE Raster support and general module maintainance help

We have a couple short term goals:

- We have run into problems when using the ArcSDEDatastore from uDig; when performing an Edit (ie writing) a desktop client is quite willing to be drawing **at the same time**.
- When using a low number of connections it is quite easy to leave the DataStore without enough SeConnections to answer basic questions such as describing a FeatureType is.

## Design and Research Pages

### [ArcSDE Installation Notes on Windows](#)

## Building a Working ArcSDE using the jsde jars

To connect to ArcSDE from java code, ESRI publishes a java library which is split across 2, 3 or 4 jars (depending on the version you use). These "java-to-ArcSDE connector jars" are here called the 'jsde' jars, so as to separate them from our own ArcSDE module.

The ArcSDE module relies on the jsde jars for all its direct communication to ArcSDE, as indeed any java-based ArcSDE client application does. However, due to concerns about redistributing ESRI's jsde jars, geotools wanted to avoid having **everyone** who built gt2 download the jsde jars from the geotools maven site. This was accomplished by the use of a 'dummy' api, which is an exact mimic of the jsde api, only it doesn't actually do anything. It's simply classes which allow compilation of the ArcSDE module, without actually having or downloading the jsde jar files.

If you want to actually connect to ArcSDE using the ArcSDE module, you'll need to download the jsde jars and instruct maven to actually use them when building the ArcSDE module.

### **You have to build your module against the jsde jars if you want to use it to really access ArcSDE**

There have been reports of problems if you:

1. Build the ArcSDE module against the dummy api
2. Copy the ArcSDE module into their application, leaving the dummy api behind
3. Copy the actual jsde jars into their application
4. Run the dummy-built module against the real jsde jars and try to access ArcSDE and fetch data

Mostly these problems have to do with static final constants which have different values and get inlined by the compiler. There is a jira issue to track this, so if you need to do the above, [check it out](#)

The jsde jars are actually very easily available. You can download the 9.1 jars [here](#) Just unroll the tar, zip and then tar files that you get, and extract the jsde\_sdk.jar and jsde\_pde.jar files.

Here's how to build the ArcSDE module with the real jsde jars:

geotools version	sde version	instructions
------------------	-------------	--------------

2.2.x	any	<p>Modify the module pom.xml to include dependencies on com.esri:jsde_sdk:9.0,com.esri:jsde_jpe:9.0 and com.esri:jsde_concurrent:9.0. All these dependencies are commented out in the pom.xml, you just need to uncomment them and rebuild.</p>
2.3.x	any	<p>Modify the module pom.xml to include dependencies on com.esri:jsde_sdk:9.0,com.esri:jsde_jpe:9.0 and com.esri:jsde_concurrent:9.0. All these dependencies are commented out in the pom.xml, you just need to uncomment them and rebuild.</p>
2.4.x 2.5.x 2.6.x	9.0	<p>add the parameter <b>-DhaveSDEJars=true</b> to your maven build command. Something like this will work just fine:</p> <div data-bbox="1092 999 1442 1266" style="border: 1px dashed blue; padding: 10px; margin: 10px 0;"> <pre>\$ mvn -DhaveSDEJars=true e -Dmaven.test.skip clean install</pre> </div> <p>Note that the jsde version 9.0 jars are publically available at <a href="http://maven.geotools.fr">maven.geotools.fr</a>, so your maven build will just automatically download these and build everything correctly.</p>
2.4.x 2.5.x 2.6.x	9.1 9.2 9.3	<p>Add both the parameter <b>-DhaveSDEJars=true</b> and <b>-Dsde.version=&lt;version&gt;</b> to your maven build command. Something like this will work:</p>

```
$ mvn
-DhaveSDEJars=true
-Dsde.version=9.2
-Dmaven.test.skip
clean install
```

Note that no versions other than jsde 9.0 are available via a maven repository, so you'll need to inject:

- jsde\_sdk
- jsde\_jpe\_sdk

Into your local maven repository yourself. Something like this should do the trick (once you've downloaded/tracked down the jars you wish to inject):

```
mvn
install:install
-file
-DgroupId=com.e
sri
-DartifactId=js
de_sdk
-Dversion=9.3
-Dfile=jsde_sdk
.jar
-Dpackaging=jar
-DgeneratePom=t
rue
mvn
install:install
-file
-DgroupId=com.e
sri
-DartifactId=jp
re_sdk
-Dversion=9.3
-Dfile=jpe_sdk.
jar
-Dpackaging=jar
-DgeneratePom=t
rue
```

ESRI includes a couple more jars which we will ignore at this time:

- concurrent - this is now covered by Java 5
- icu4j - this is available from ibm and has already been published to our maven repository

## So what do I have to type again?

To install and use the latest 9.3 jars (or whatever server you have):

1. Install the ArcSDE SDK
2. Choose the option to install "arcsde developers sdk"
3. It will copy the jars into C:\ArcGIS\ArcSD lib by default
4. Open up the command line and navigate to where the jars are

5. Type the following maven command:

```
mvn install:install-file -DgroupId=com.esri -DartifactId=jsde_sdk  
-Dversion=9.3  
-Dfile=jsde_sdk.jar -Dpackaging=jar -DgeneratePom=true  
mvn install:install-file -DgroupId=com.esri -DartifactId=jpre_sdk  
-Dversion=9.3  
-Dfile=jpe_sdk.jar -Dpackaging=jar -DgeneratePom=true
```

6. To set up your eclipse development environment for testing - go to the root of geotools and type:

```
mvn eclipse:eclipse -Parcsde -Dsde.version=9.3
```

7. You can also build on the command line:

```
mvn clean install -Parcsde -Dsde.version=9.3
```

## Running the Tests Against your Own Server

TBA