The ESRI Geodatabase Framework

MSc Marco Boeringa
2013

*** **DISCLAIMER** ***: This is **not** an ESRI document. It has been created by the author solely, and any errors herein are his only. The document contains a diagram that tries to represent the most significant parts of what the author calls "The ESRI Geodatabase Framework", with a focus on showing the different options to access spatial data in an enterprise database. As this framework is rather involved, and has evolved through time, the author makes no assumption of completeness nor correctness by representing this figure. There are, and have been, other components and options (E.g. CAD Client, ArcSDE C/JAVA API, MapObjects, ArcView 3.x etc...), and there may be issues in the way the figure represents reality. Feel free to contact the author if you spot an obvious error or have suggestions for improvement. The author takes no responsibility for problems or damage caused by errors in, or misinterpretation of, the contents of this document.

Contents

1 Introduction

Since I know from my own experience that getting familiar with all the different components of what I hereby would like to call "The ESRI Geodatabase Framework" can be a challenge, and have seen many questions, confusion, and some misconceptions abound on the ESRI Forum pages, I decided to create a diagram showing the relationships of the most prominent parts of the hard-, and especially software involved in setting up an ESRI Geodatabase in an enterprise database, and the components needed to actually access the data in ArcGIS, a custom webmapping application, or maybe even non-ESRI software like MicroStation or AutoCAD.

The task of creating this diagram has proven a challenge. How to represent so many different components and aspects of "The Framework" as I see it, in just one A4 page and yet maintain readability? Whether I succeeded or not, is up to you as the reader… Nonetheless, I sincerely hope this document with its diagram and accompanying explanation, will aid others struggling with the diverse concepts, in better understanding "The Framework", and grasp the relationships of all components as a whole instead of isolated elements. I also hope it will lead to, or at least encourage, a more consistent usage of the terms and names given by ESRI to the different components. Unfortunately, the wrong usage of terms and product names is common in many discussion threads, sometimes close to turning into a Babylonian Confusion, making the problem of understanding the "whole" an even bigger challenge than it already is.

I strongly recommend you to print the current document you are reading, or at least the included main diagram, in full color, as otherwise vital distinctions between connection lines and connection options will be lost.

Please note this document does not contain a full glossary and explanation of each component by itself. I hereby would like to refer you to ESRI's comprehensive Help system, that has many inroads to learning more about a specific software component, for further reading and details. In addition, there is a slightly older but still relevant article by ESRI employee Derek Law that gives better insight to key concepts of ArcSDE and geodatabases and goes deeper into some of the technical details:

Enterprise Geodatabase 101 - *A review of design and key features for GIS managers and database administrators*, Derek Law, ESRI - http://www.esri.com/news/arcuser/0408/entergdb_101.html

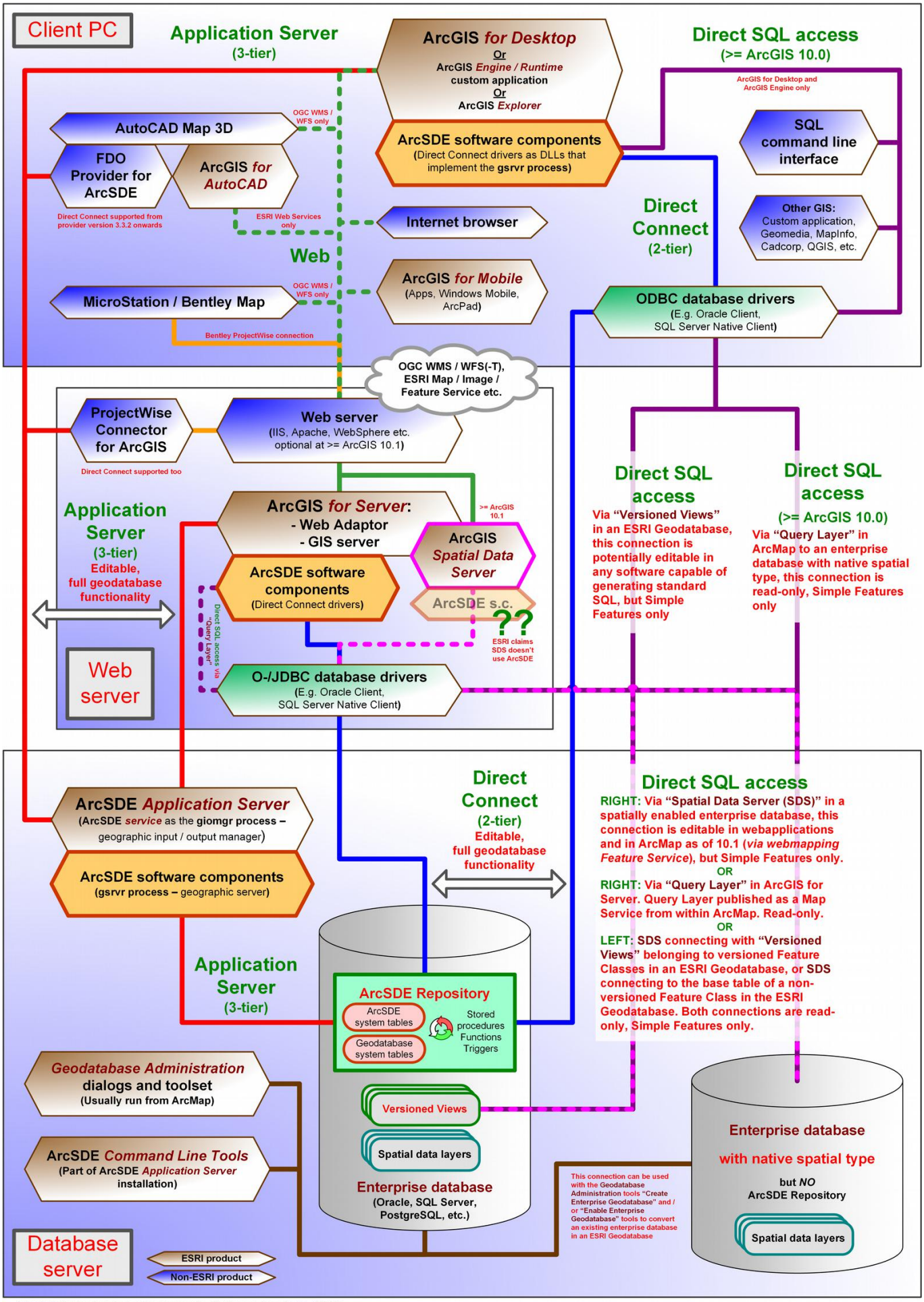
A final remark about this document: I have made every attempt to verify facts and features of products I mention, even if I haven't ever used them myself, to allow me to properly "fit" them inside "The Framework". This information was mainly gathered from the most recent technical documents, Online Help etc. as made available by the respective software developing company. However, this also means it reflects the most-up-to-date information I could find at the very point in time I write this. If you are running older versions of the mentioned software, some capabilities and even entire pieces of software, may be lacking or be non-options until a future upgrade. Keep this in mind while reading this. I have tried however, wherever relevant, to mention the lowest version of the software a certain capability was introduced. Lastly, there are third-party tools that complement or extend the ESRI Geodatabase Framework that are not included in the diagram. I make no assumption of completeness nor absolute correctness presenting it here.

MSc Marco Boeringa, May 2013

The Netherlands

Document version: V1.1

marco@boeringa.demon.nl



The ESRI Geodatabase Framework, © Marco Boeringa, 2013

2 How to read the diagram

In the diagram, three basic hardware components of a computer network are visible: A "Database server", "Web server" and a "Client PC", the last one probably being where you sit right now reading this. On each of these pieces of hardware, several software components are visible, most notably the ESRI products and an enterprise database.

The software components are linked by colored lines, that represent different types of connections, each of which are also marked by their names in green bold text. Four main types of connections are visible:

- Direct Connect (2-tier) connections
- Application Server (3-tier) connections
- "Direct SQL access" connections. Note this is unofficial terminology, but used in a couple of ESRI documents and webpages.
- Web / internet connections

Each of these connection types will be discussed later in this document.

Although there is a host of different software components visible, this doesn't mean you need them all… If, for example, you decide to stick with the ArcGIS 10.1 and beyond default "Direct Connect 2-tier" connection to your ESRI Geodatabase, then you don't need to install an ArcSDE *Application Server*. And the other way around, if you currently already run - or decide to run - an ArcSDE Application Server to access the geodatabase data through an "Application Server 3-tier" setup, then you don't have to install an appropriate "ODBC database driver" on each Client PC that needs access to your geodatabase. Similarly, if you decide to setup webmapping services and only need Simple Features access to vector data without advanced geodatabase behavior, than you can use the new 10.1 **ArcGIS *Spatial Data Server*** instead of installing the full **ArcGIS *for Server***. If instead, you actually *need* advanced geodatabase functionality in your webmapping application, than installing ArcGIS *for Server* is a must. So don't expect or think you need, or will see, all of these options in each organization. It is up to you to decide, based on your specific organizational needs, which parts of the "ESRI Geodatabase Framework" to install.

Please also note that, although separately visible in the figure, the **ArcSDE software components**[1] do ***not*** have their own installation on the "Client PC" and "Web Server", contrary to all other visible components.

The ArcSDE software components form an integral and vital part of the **ArcGIS *for Desktop*** and **ArcGIS *for Server*** installations and will be installed as soon as you install these.

In case of the "Database Server" as visible in the diagram, the situation is different. Installation of ArcSDE software components used to be mandatory at ArcGIS version 10.0 and below, as there was no other way of creating an **ArcSDE Repository** in an enterprise database, the step that essentially enables the database to become an ESRI ***Geo***database (more on this later in this document).

Now, at 10.1, installing ArcSDE software components on a database server is *an option*.

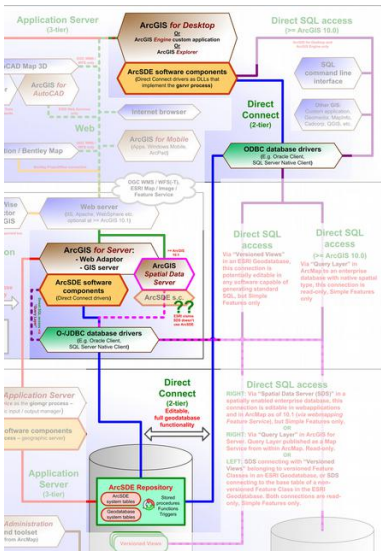


Fig 1: Direct Connect has become the default option for connecting to a geodatabase in ArcGIS 10.1

[1] Not official ESRI terminology

The reason for this is the enhancements to, and introduction of, geoprocessing tools in the new Geodatabase Administration dialogs and ArcToolbox toolset that can replace the traditional role of the tools included with an ArcSDE installation on a database server, in combination with the push of ESRI to Direct Connect (2-tier) connections. The primary new tools are "Create Enterprise Geodatabase" and "Enable Enterprise Geodatabase", both tools now capable of creating the ArcSDE Repository in any enterprise database accessible from your user account and corporate network.

For experienced users and geodatabase administrators, there may still be use for installing ArcSDE software components for troubleshooting purposes or enabling some specific non-default option. This is especially true in case one decides to install an ArcSDE Application Server. It is essential to realize that the ArcSDE installation is, and was, a two part installation, although modified / simplified in 10.1:

- Installation of an **ArcSDE *Application Server***. Usually installed on the database server, but maybe installed on a physically separate machine as well, although ESRI discourages this quite strongly due to potential high network contention. This option installs the basic command line tools and executables to define, start and run an ArcSDE *Application Server*.
- Installation of the **ArcSDE *Command Line Tools*** and supporting files. Command Line Tools can be installed locally, on your Client PC, or on the database server. The ArcSDE *Command Line Tools* extend the range of administrative tasks that can be applied against an ArcSDE *Application Server* and an enterprise geodatabase.

You are not required to install both of these options, but in general, once an ArcSDE Application Server is installed, it is also recommendable to install the full set of ArcSDE Command Line Tools.

2.1 Installation of the ArcSDE *Application Server*

As of version 10.1 of ArcGIS, the **ArcSDE *Application Server*** installation needs to be downloaded from Esri's Customer Care Portal. ESRI distributes most of these Customer Care Portal downloads as ISO DVD images, which means you can transfer them to DVD yourself using Windows' build in options, and subsequently use the DVD for installation, or use some form of ISO extractor. There is a separate installation of the ArcSDE *Application Server* for each supported DBMS included on the ArcGIS installation DVD for ArcSDE. This installation used to be called the "Post installation setup" prior to 10.1, as it did more than installing the ArcSDE *Application Server*. The "Post installation setup" also created the ArcSDE Repository. As this second task is now entirely taken over by the Geodatabase Administration toolset in ArcGIS 10.1 and beyond, there is just the installation of the ArcSDE *Application Server* itself left. If, however, you decide to use Direct Connect exclusively, than installing an ArcSDE *Application Server* is not necessary. You will still have the full capabilities of ArcSDE and ArcGIS using Direct Connect. *ESRI actively promotes Direct Connect as the preferred and default method of accessing an enterprise geodatabase!* More on this later on.

An introduction to setting up an ArcSDE Application Server at 10.1 is here: (blog article ESRI Australia)

Creating an ArcSDE Service on Windows using ArcGIS 10.1

2.2 Installation of the ArcSDE *Command Line Tools*

The installation of the **ArcSDE *Command Line Tools*** is included in the same DVD ISO image as for the ArcSDE *Application Server*. It will install the ArcSDE *Command Line Tools* and supporting files that allow diverse administrative tasks against a geodatabase. As said, the need for these tools has diminished in 10.1 with the introduction of the Geodatabase Administration dialogs and toolset in ArcGIS for Desktop.

From the 10.0 Help (this is no longer part of the 10.1 and beyond Help):

"Installing ArcSDE creates a directory (referred to as SDEHOME) that contains library files, administration command utilities, ArcSDE Administration Command Reference documentation, ArcGIS Server help, and

database-specific scripts. It also sets the SDEHOME variable in your system path."

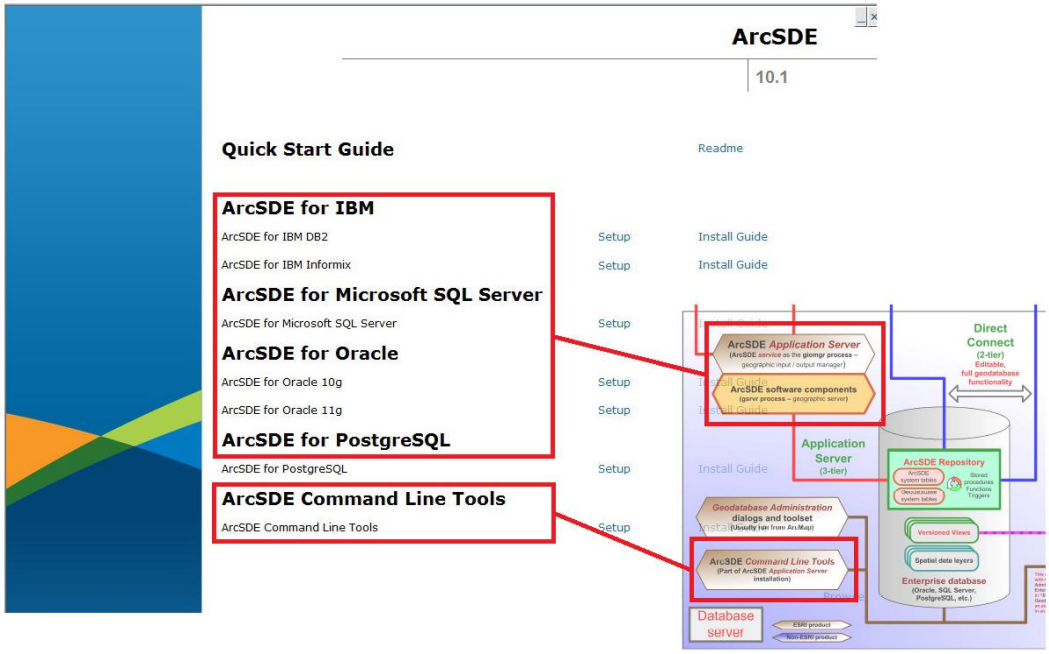


Fig 2: Installation of ArcSDE Application Server (Top 4 options, e.g. ArcSDE for PostGreSQL) and ArcSDE Command Line Tools

3 <u>What do the colored lines in the Geodatabase Framework diagram represent?</u>

The colored lines represent all of the connections between the software components making up "The ESRI Geodatabase Framework", and communication / data traffic across your organizational network (LAN – Local Area Network), except the green lines ending at the user's internet browser, which are internet (WAN - Wide Area Network) connections.

Connections basically run from top to bottom and the other way around. *It is important to note that traffic and connections do **NOT** jump colors.* So you can travel from "ArcGIS for Desktop" via the "ArcSDE software components" through a blue line to the "ODBC database drivers" on your Client PC, and from there via the blue line to the "ESRI Geodatabase", but you can't switch at the ODBC database driver to the purple line to end up at the "Enterprise database with native spatial type".

There are just two exceptions to this rule, and that is the switch **to**, or **from**, a green (stippled) line representing the traffic over the internet starting at the Web Server, and the specific case of the orange line running from MicroStation / Bentley Map to Bentley's "ProjectWise Connector for ArcGIS", a connection that from there on continues as a red line to the "ArcSDE Application Server" and finally the geodatabase.

In addition, *since basic traffic is from top to bottom*, you also can't travel across for example the purple line from the "ESRI Geodatabase" to the "Enterprise database with native spatial type", nor across the red line from "ArcGIS for Desktop" via the "ArcSDE Application Server" to the "ArcGIS for Server" component.

Please also note that the green depicted internet connection lines could potentially also end at the user's ArcGIS for Desktop software instead of the Internet browser, when a user accesses a WMS or WFS webmap service from within ArcMap or ArcCatalog. Or it might end at an AutoCAD or MicroStation client in a similar scenario. Both of these connection options are incorporated in the diagram.

So what do the colored lines represent?

Red lines represent so called "Application Server (3-tier)" connections to an **ESRI *Geodatabase***. All of this traffic from **ArcGIS *for Desktop*** or **ArcGIS *for Server*** passes through the **ArcSDE *Application Server***.

Blue lines represent so called "Direct Connect (2-tier)" connections to an ESRI Geodatabase, and always pass through an **ODBC driver** for the underlying respective enterprise database.

Both Application Server (3-tier) and Direct Connect (2-tier) give access to the full geodatabase functionality, including versioned editing, history management, Feature Datasets, Feature Classes, Annotation, Topologies, Parcel Fabrics, Geometric Networks, Network Datasets, etc.

Purple (stippled) lines represent two things:

1) the new ArcGIS 10.0 and beyond "Direct SQL access" [2] connections to a spatially enabled enterprise database *without* **ArcSDE Repository** using **Query Layers** created via the "Make Query Layer (Data Management)" tool. Such a "native spatial type" database could, for example, have been created by Bentley's MicroStation products, or Autodesk's AutoCAD product line, or any GIS webmapping application designed to directly edit a spatially enabled enterprise database. Geographic datasets can also be added / imported to the database via such a connection using ArcGIS, but the data is read-only - at least in current ArcGIS version 10.0 and 10.1 - in ArcMap. So you can't edit such a dataset in ArcMap.

2) Access from non-ESRI software or a SQL Command Line interface to **Versioned Views** in a geodatabase to view or edit the data using standard SQL (See "What are versioned views?" in the ArcGIS Help).

While viewing data via something like Query Layers or Versioned Views is not an issue, a word of caution is in order though regarding editing of geodatabases via SQL using generic SQL tools like Oracle SQL*Plus. ESRI specifically warns not to edit any datasets participating in geodatabase behavior, for an overview, see here:

"A quick tour of using SQL with enterprise geodatabases"

"What type of data can be edited using SQL?"

Pink stippled lines show the connections of the **ArcGIS *Spatial Data Server*** (new in ArcGIS 10.1 and beyond) to enterprise databases with native spatial type data, and ESRI Geodatabases. In both cases, the connection is Simple Features only, and is editable from within ArcGIS or a webapplication if an ESRI "Feature Service", or OGC (Open GIS Consortium) "Transactional WFS" service is used to serve out the data to the internet [3].

Green stippled lines represent internet connections over the WAN (Wide Area Network) or LAN (Local Area Network) to the internet browser, **ArcGIS *for Desktop***, **ArcGIS *for AutoCAD*** or CAD software directly, or any other (custom build) application capable of accessing OGC WMS, WFS or ESRI Map, Image and Feature Services.

Orange lines represent Bentley's "ProjectWise Connector for ArcGIS" connections from Bentley Map / MicroStation to an ESRI Geodatabase. This is a product developed by Bentley.

[2] Not official ESRI terminology, but informally called so in some documents
[3] Editing a WFS-T service is not yet supported from within ArcGIS for Desktop, ArcGIS needs an ESRI Feature Service for editing

4 <u>What is ArcSDE? The ArcSDE Repository and associated software processes</u>

One of the biggest confusions and misconceptions seems to arise from not understanding the role of ArcSDE, or **ArcSDE Technology** as ESRI likes to call it nowadays, as part of the "ESRI Geodatabase Framework" [4]. ArcSDE is often blindly associated with the software component marked as the **ArcSDE *Application Server*** (ArcSDE service) in the diagram, the component running as a service on the database server and used to create 3-tier connections to the geodatabase.

However, in reality, ArcSDE's major role is not so much as a simple service allowing 3-tier connections - "3-tier" because the ArcSDE Application Server sits in-between the client ArcGIS software and the DBMS - but as the *<u>vital</u>* and *<u>integrated</u>* software component of primarily ArcGIS *for Desktop*, ArcGIS *for Server,* the ArcSDE *Application* Server and the RDBMS, in allowing all of these to manage and use spatial data stored in a RDBMS.

ArcSDE Technology is implemented in the form of 5 - *or 4 in the case of Direct Connect* - components that closely work together:

- The **giomgr process**. This is the **ArcSDE *Application Server*** or **ArcSDE service.** It is an executable that runs on a server and acts as the geographic input / output manager accepting and relaying user connections to the database. *Please note installing and using an ArcSDE Application Server is **optional***. There is *no* equivalent of the giomgr process is in 2-tier Direct Connect connections to a database, it is specific for the ArcSDE Application Server.

- The **gsrvr process**. This is the geographic server managing the geographic data throughput from a specific client to the geodatabase. This process is used in both 3-tier ArcSDE Application Server, and 2-tier Direct Connect connections.

- The **ArcSDE system tables**. A set of tables that store crucial information and system metadata for the multi-user operation of the geodatabase and allowing versioned storage of spatial data.

- The ArcSDE database **Stored procedures**, **Functions** and **Triggers**. These implement programming and operational logic of ArcSDE at the database level.

- The programmatic objects and API's implementing and supplementing all this functionality in ESRI's ArcGIS product line. Think of ArcObjects, ArcGIS Runtime, the ArcSDE C/JAVA API etc.

The ArcSDE system tables, stored procedures, functions and triggers together form what ESRI calls the **ArcSDE Repository** and is visible in the diagram. In addition, the Geodatabase system tables are generally considered part of the ArcSDE Repository too, although these don't cater for ArcSDE's main function - the versioned editing environment - but for ArcGIS's advanced Geodatabase functionality, supporting complex data types like Network Datasets.

ArcSDE's main function is allowing for the multi-user versioned long transaction workflow, supporting multiple concurrent editors executing edits on the same datasets in workflows that may take days, like updating a large scale road map to reflect a new highway's infrastructure. Without ArcSDE, nothing of this would be possible in the ESRI product line. ArcSDE Technology is thus responsible for the operation of this type of spatial database storage environment in the ArcGIS product line, much like other GIS vendors (AutoDesk, Bentley) have their own components for managing CAD and GIS data in spatial databases (e.g. AutoCAD Map 3D).

In the diagram I have attempted to illustrate this crucial, but often misunderstood, position of ArcSDE Technology inside the Geodatabase Framework by adding the **ArcSDE software components** [5] as a separate component, attached to the respective software they form part of (e.g. ArcGIS *for Desktop*, ArcGIS

[4] Not official ESRI terminology
[5] Not official ESRI terminology

for Server). In reality, to better understand this, you may check out your "ArcGIS *for Desktop*" program folder. In a standard installation this is:

"**C:\Program Files (x86)\ArcGIS\Desktop10.0\Bin**"

If you look inside this folder, you will see DLL's like **"sde.dll"**, **"sdeora11gsrvr101.dll"**, and **"gsrvrora11g101.dll"**, all of which are in full use as soon as you attempt to access an ESRI Geodatabase. This is especially true for a 2-tier connection, where all processing takes place on your own client PC running ArcGIS for Desktop. Note there are separate DLLs for the different RDBMs (e.g. **"gsrvrora11g101.dll"** for Oracle and **"gsrvrsql101.dll"** for SQL Server) and the inclusion of the ArcGIS release number in the file names.

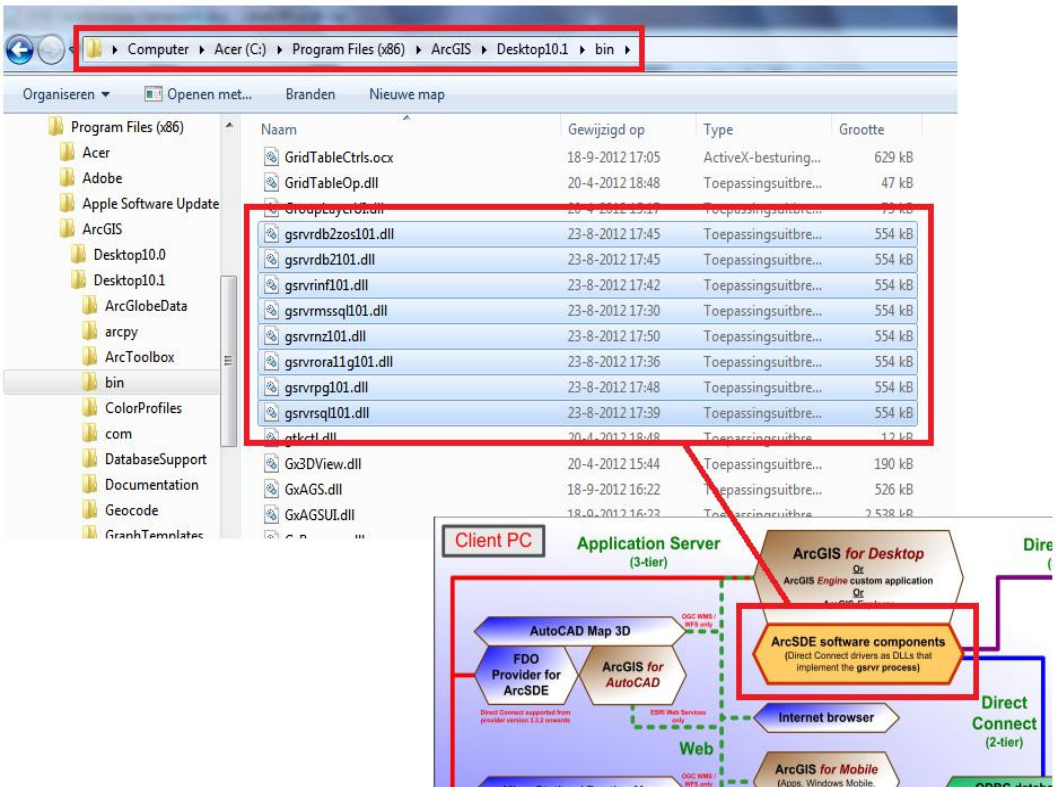


Fig 3: The ArcSDE Direct Connect drivers form an integral part of ArcGIS for Desktop and ArcGIS for Server.

Now you may be wondering how ArcSDE manages all this. This is where the **ArcSDE Repository**, **ArcSDE system tables**, and the **Geodatabase system tables** come into play. The ArcSDE Repository stores these two sets of tables that contain the logic and metadata of all spatial and non-spatial tables that form part of an enterprise geodatabase. The ArcSDE system tables mainly cater for ArcSDE's main tasks, that is, to facilitate the storage of basic geometries in a relational database, and support both short and long transaction versioned editing. The Geodatabase system tables on the other hand, primarily cater for the advanced geodatabase behavior of complex datatypes like Network Datasets, Parcel Fabrics, Topologies etc. Each time you "register" a spatial table with the enterprise geodatabase, it gets registered into these tables. From that moment onwards, it is considered part of the geodatabase and called a Feature Class, and can participate in any advanced geodatabase behavior and all the geodatabase tools ArcMap has on offer, for example executing network analysis against a Network Dataset. As the ArcGIS Help states:

"The system tables for a geodatabase enforce geodatabase behavior, store information about the geodatabase, and keep track of the data stored in the geodatabase."

As can be seen in the diagram, ***all*** data traffic to an ArcSDE Repository passes through ArcSDE software components, irrespective of a 2- or 3-tier connection being used. There is no way around this. This is not to say you couldn't potentially access this data otherwise, e.g. by means of a SQL command

line interface, but any SQL edits to these system tables carry the very real and serious risk of corrupting the ArcSDE Repository, and rendering the entire enterprise geodatabase useless.

4.1 The ArcSDE software components and the ArcSDE Application Server

Please note there is a vital distinction between the **ArcSDE software components**, and the **ArcSDE *Application Server***, often abbreviated to "ArcSDE *service*". The ArcSDE *Application Server* processes Client PC connection requests of users trying to access the geodatabase, and handles TCP/IP traffic between the ArcGIS client application (e.g. ArcMap) and the database server, while the ArcSDE software components handle the true SQL logic and communication with the enterprise database to support basic ESRI geodatabase functionality. The ArcSDE Application Server acts as kind of "broker", spawning (starting) processes on the server to help establish a connection to the geodatabase.

The ArcSDE Application Server's functionality is implemented via the **giomgr** [6] process / executable, while the ArcSDE software components' role is implemented via the **gsrvr** [7] process. Both are running on a server in case of a 3-tier connection. In a 2-tier situation, similar functionality is implemented via the ArcSDE Direct Connect drivers on the client PC, except that there is no giomgr process. This is well illustrated by the file names of the DLLs involved in the Direct Connection on the client PC. Go back a few paragraphs, and note I wrote about the DLLs stored in the ArcGIS for Desktop installation folder. Note the naming of one of the listed files:

"**gsrvr**ora11g101.dll"

It uses the same abbreviation as the process name of the geographic server executable that is the counterpart of it in a 3-tier Application Server connection...

As the ESRI Help page "What is an ArcSDE service?" states:

"***The giomgr process***

This is the ArcSDE service monitor. Each ArcSDE service has one giomgr process. This process listens for user application connection requests, spawns gsrvr processes, and cleans up disconnected user processes. The giomgr will not start if a valid server license has not been installed.

The gsrvr process

The giomgr process spawns a gsrvr process for each application connected to the ArcSDE service. The gsrvr process is dedicated to a single user/application connection. It communicates with the database on behalf of the connected application. The gsrvr process responds to the queries and edit requests the application sends to the database."

More detailed information regarding the giomgr and gsrvr processes and their function during the process of establishing a connection from a client PC to an ArcSDE Application Server and a geodatabase, can be found on the "A quick tour of ArcSDE services" Help page.

4.2 The ArcSDE Application Server and the Direct Connect drivers

It is important to realize that the distinction between the ArcSDE Application Server, implemented as executables running on a server, and the Direct Connect drivers, implemented as DLLs on the client PC, is actually a *very* small one. As Derek Law (ESRI) in Enterprise Geodatabase 101 writes:

"*Direct connection drivers are built from the same software code used to build the ArcSDE service. The difference is that direct connect drivers are built as dynamic-link library files and execute in the process space of the client application, whereas the ArcSDE service was built as an executable program that runs on the ArcSDE server."*

[6] geographic input/output manager
[7] geographic server

And as Vince Angelo (ESRI) wrote in an ESRI Forum thread:

*"Direct Connect *is* an ArcSDE connection. ... It's ArcSDE that provides the basis for versioned geodatabases. With Direct Connect, it's just a multi-threaded solution in the ArcGIS client instead of a multi-processing solution on the database server."*

And in another thread even more pronounced:

*"... Direct Connect *is* an ArcSDE server."*

This last sentence may leave you baffled and seem an oxymoron, but think about it one other time in the context of all that was written before, and it may not longer be such a contradiction. And as you may now also begin to understand, the often heard quote that:

"Direct Connect (2-tier) connections don't use ArcSDE."

is in fact a serious misconception. Any contact with a geodatabase requiring access to the ArcSDE Repository, will need and use ArcSDE Technology in some form.

5 The ancillary role of ArcObjects in managing geodatabases next to ArcSDE

That said, the last sentence needs refinement. ArcSDE Technology is just one part of the equation to allow the advanced geodatabase behavior and spatial datasets we use on a daily basis, for example when analyzing a Geometric Network to find the quickest or most economical route to deliver goods.

It is the combination with ArcObjects, as the foundation of ArcGIS for Desktop and ArcGIS for Server, which fills the gap. The role of ArcSDE Technology and ArcObjects in combination is illustrated well by the following two remarks in Derek Law's Enterprise Geodatabase 101:

"ArcSDE technology provides fundamental capabilities that include

- ***Access and storage of simple feature geometry in the RDBMS***
- *Support for native RDBMS spatial types (if available)*
- *Spatial data integrity*
- *Multiuser editing environment (i.e., versioning)*
- *Support for complex GIS workflows and long transactions"*

"The upper level of the application tier, ArcObjects, implements geodatabase application logic. This set of platform-independent software components is written in C++ and provides services to support GIS applications as thick clients on the desktop and thin clients on the server. This technology component is built into GIS clients (e.g., ArcGIS Desktop) and implements more complex object behavior and integrity constraints on simple features, such as points, lines, and polygons, stored in an RDBMS. ***In other words, ArcObjects implements behavior on the feature geometries.****"*

The highlighted sentences in both quotes reveal an important thing to note. To ArcSDE, features in a geodatabase are just geometric spatial features with some attributes, all stored in database records. It is the ArcObjects layer, and the associated feature and attribute data stored in the geodatabase system fields and tables associated with each Feature Class, that implements the truly advanced behavior of a geodatabase and allow you to work with complex data types like Network Datasets, Topologies and Parcel Fabrics. ESRI invested years of software development into these ArcSDE software components and the ArcObjects code to allow it all. A geodatabase thus is more than a few Simple Feature layers bundled together in an RDBMS, at least from the perspective of the user interacting with it through all of ArcMap's advanced tools for data management and analysis as realized in the ArcObjects component model.

6 ArcGIS for Server and the ArcSDE Application Server

Despite the possibly confusingly similar names, Arc**GIS** *for Server* and the Arc**SDE** *Application Server* are two completely different ESRI products. They serve different purposes, so don't mix them up!

As said, the ArcSDE *Application Server* is responsible for processing connection requests to the geodatabase, while ArcGIS *for Server* is a suite of software components for creating and managing GIS web-applications and services to be hosted on the internet. These include webmapping, geoprocessing, metadata services etc.

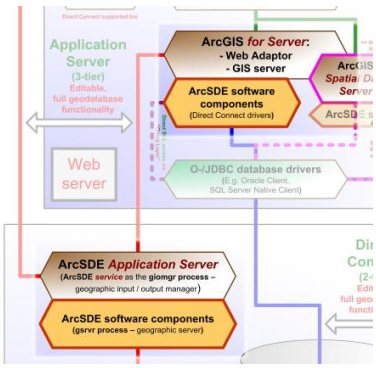


Fig 4: ***ArcGIS for Server*** *and the* ***ArcSDE Application Server*** *are two distinct products and should not be confused.*

The two components may work together though: ArcGIS *for Server* can use an ArcSDE *Application Server* to access spatial data in a 3-tier connection to a geodatabase (red lines), but is also capable of using a 2-tier Direct Connect connection (blue lines) for the same purpose.

7 Direct Connect (2-tier) versus Application Server (3-tier)

Another often discussed topic on user forums is the distinction between Direct Connect and Application Server connections to a geodatabase. From a functionality point of view, there is absolutely no difference between these connection options. Both options support the full geodatabase model and hence offer similar functionality from within the familiar ArcMap environment. From an IT and geodatabase administration point of view, there are some differences, as outlined in other parts of this document and below.

From ArcGIS 10.1 onwards, Direct Connect (2-tier) is the default option for connecting to an ESRI Geodatabase. As visible in the diagram, setting up Direct Connect requires suitable (ODBC) database drivers installed for the underlying enterprise database. This is achieved by installing the respective database client software, like Oracle Instant Client or SQL Server Native Client on the client PC running ArcGIS for Desktop. It is vitally important to install the correct database client version: since **ArcGIS *for Desktop*** is still 32-bit, you need to install the 32-bit version of the database driver, even on a 64-bit operating system like Window 7/8 64-bit. Please note that, as mentioned in the previous chapter, **ArcGIS *for Server*** can also use Direct Connect when suitable ODBC database drivers are installed on the web server. Since ArcGIS *for Server* is 64-bit, you need to install a 64-bit database driver on the web server in case you want to setup a Direct Connect connection from your ArcGIS for Server to the geodatabase. The database driver is used to submit SQL statements to the database and receive back the data.

ESRI says Direct Connect is better scalable compared to Application Server connections, as it takes part of the processing of the database server, and transfers it to the client PC, thus freeing up server resources. The main reason for the push to Direct Connect is therefore probably an increasing number of ESRI clients seeing a significant rise in the number of concurrent database users, requiring the better scalability of the 2-tier setup. In addition, the setup and management of an ArcSDE Application Server to allow 3-tier connections, puts an extra burden on the ArcSDE / geodatabase administrator, that some rather be without. That said, the need to install a database client / driver on each client PC to allow Direct Connect 2-tier connections, has to be considered as well, and may impact organizational IT policies.

Application Server connections do not require installed database drivers to work, but instead rely on TCP/IP traffic from the client PC straight to the ArcSDE Application Server installed on the database server [8]. To allow this traffic, you may need to make a small change to the Windows "services" file to open up the default TCP/IP port 5151 that ArcGIS for Desktop uses to communicate with the ArcSDE Application Server [9]. This potentially has to be done both on the client PC where ArcGIS for Desktop resides, and on the server where the ArcSDE Application Server runs. Details can be found here: "Editing service files on Windows". Please note that the change in TCP/IP number described on this page to the services file on the ArcSDE Application Server, and the entry of an edited "4000" TCP/IP number, is only required if you ***don't*** want to use the default port 5151 for some reason (e.g. already being used by another Application Server), otherwise you can accept the default 5151 TCP/IP port and simply remove the comment mark if it hasn't already been removed. If this same port is specified in the "Create ArcSDE Connection File" file tool, than modifying the Windows services file on the client PC where ArcGIS for Desktop resides is unnecessary too. As Vince Angelo (ESRI) wrote in an ESRI Forum thread:

"The services file on the application server must be modified because the service is started by name. It's good practice to distribute a modified services file among clients, but it's only necessary if those clients will also lookup by name (if you specify the port number, then it isn't necessary)."

[8] The ArcSDE Application Server is not required to be installed on the same server as the RDBMS. It can be installed on a separate server, but is not the recommended configuration. See the ESRI documentation for instructions.
[9] Default port is 5151, but is configurable on the server. For details, see: "The services.sde file" in the Help.

However, Direct Connect being default as of 10.1, you can still install an ArcSDE Application Server to connect to an ESRI Geodatabase at 10.1, and continue to use any existing Application Server if you have one already installed. You must take notice of some changes though compared to previous versions of ArcGIS. Application Server connection files can no longer be created using the standard Database Connection dialog at 10.1. You have to use the new "Create ArcSDE Connection File (Data Management)" instead, to create a connection file capable of connecting to an ArcSDE Application Server. This tool is not part of the default user interface of ArcGIS for Desktop. Use the **Search** dialog box to find and open it. Please note this tool only creates a connection file, it *doesn't* install or configure an ArcSDE Application Server. Installing an ArcSDE Application Server was discussed in chapter 2.1.

Depending on your specific client/server hardware and network configuration, and the contents of your geodatabase, there may still be reasons to run an ArcSDE Application Server in some situations. One compelling argument I encountered was by D.E.Wright on the GIS StackExchange Forum:

*"There are some very good reasons to use an ArcSDE Server Engine (**note by author:** Should be ArcSDE Application Server in official ESRI terminology), the first being the load. When you utilize a ArcSDE (**edit by author:** Application) Server Service you are taking the bulk of that data load off the database server and queuing it versus relying on just your local machine to store all that temp data.*

*One thing you will see especially with a MSSQL server when you make your initial database connection in a MXD is that ArcGIS does a 'SELECT *' (as seen in your query analyzer and logs on your DB Server) against that table/feature-class. Now, this can be a huge impact if you have very large datasets; the ArcSDE Service/Process helps in this by aiding in the request of the appropriate data scope.*

*Now as we have all gotten bigger machines, with more RAM it's much easier to just load everything into the current session and run with it (**note by author:** this is Direct Connect); but don't just discount the idea of using the service just because the ArcGIS docs say you 'don't need it' anymore, versus when you probably could/should use it."*

For those still struggling with the difference between a 3-tier ArcSDE Application Server connection and a 2-tier Direct Connect connection: think of the ArcSDE Application Server running on a remote server being "public transport", and the ArcSDE DLLs on your local computer as part of "ArcGIS for Desktop" as your "private car". Both share an "engine" - this is ArcSDE! - but they are independent and *both* get you from A to B: allow you to access an ESRI Geodatabase with all of its functionality.

It is up to you to decide if you want to travel by "public transport" or use your "private car". One mode of transport may be faster than the other, or the other way around, depending on the conditions in your local "area"...

There is also the option to run a mixed environment, with both 3-tier ArcSDE Application Server connections and 2-tier Direct Connect connections. The options are not mutually exclusive but instead complement each other. From a practical point of view though, the most likely situation to encounter such a set up is in an organization running legacy 9.x versions of ArcGIS next to new 10.x versions, where the ArcGIS 9.x or 10.x clients connect to a 9.x ArcSDE Application Server and associated geodatabases, whereas the 10.x clients use Direct Connect to connect to 10.x geodatabases.

8 Versioned Views

Even though the preferred, and most functional way, of accessing an enterprise geodatabase is by means of ArcSDE and ArcGIS for Desktop, data in so called Versioned Views [10] in a geodatabase, can be accessed directly via SQL. A Versioned View can be created on any dataset registered as versioned in the geodatabase, and, as of 10.1, simply registering a dataset as versioned will automatically create a Versioned View. Versioned Views carry the same name as the spatial data layers they are based on, just with an extra "**_VW**" suffix appended to the name.

ESRI designed this option primarily for editing through a SQL client or command line interface (e.g. SQL Server Management Studio or Oracle SQL *Plus), or to allow non-ESRI software to access and potentially edit geodatabase data (e.g. AutoCAD, Microstation, or a custom build application). In case of editing through a SQL client, a number of supporting ArcSDE database stored procedures can be used to create a new "named version", and explicitly start / stop an editing session. Creating a new named version to edit is also the recommended workflow. See the links in footnote 11 below for details about this.

There is a couple of caveats to take notice of though when using Versioned Views:

- Versioned View access is restricted to Simple Features only, so none of the advanced ESRI Geodatabase functionality like Feature Dataset / Feature Class hierarchy, Topologies, Geometric Networks etc. is available. Essentially, you are accessing the geodatabase as if it were any regular enterprise database with spatial data stored in it.

- Versioned Views need to use native spatial storage types, in order to be able to easily access the geometries from a SQL client or GIS/CAD package and view the spatial data. The ArcSDE default storage options for most databases satisfy this requirement (Geometry for SQL Server and ST_Geometry for other databases). In case of Binary storage (SDEBINARY or OGCWKB) SQL access to a Versioned View will be limited to attributes only, unless the SQL client is capable of directly reading and displaying the binary data.

- Unless you are editing the data through a SQL command line interface and explicitly *start* and *end* an editing session, and explicitly *create* and *set* a "current version", by running one of the **ArcSDE stored procedures** specially designed for this purpose, you can only access the DEFAULT version through a Versioned View [11]. QGIS for example, only sees the DEFAULT version when accessing a Versioned View.

- This also means long transaction editing on a specific named version is currently only supported through a SQL command line interface. All other editing applications (e.g. MicroStation, AutoCAD Map3D, QGIS) will perform short transaction edits against DEFAULT if they make use of Versioned Views. Keep in mind that MicroStation and AutoCAD have more advanced options for accessing an ESRI Geodatabase, that will be discussed in chapter 14.

- If new named versions are created using the ArcSDE "create_version" stored procedure in a SQL command line interface, you will need to reconcile, post and delete versions edited in the SQL interface by means of ArcGIS for Desktop. So version management is mainly through ArcGIS for Desktop.

- Directly editing against the DEFAULT on Archive enabled Feature Classes had an issue in ArcGIS version 9.3.1 and below, and was therefore discouraged. Instead, the recommended workflow of creating and editing a new named version should be applied. See this Knowledgebase article.

[10] Called "Multiversioned Views" in ArcGIS releases 10.0 and below
[11] See the Help pages "A quick tour of editing versioned data using SQL" and "Editing versioned data in SQL Server using SQL" for more information regarding the editing through SQL and usage of the ArcSDE stored procedures.

It is important to note that ArcGIS for Desktop can access Versioned Views as well via the **Make Query Layer** option, which will be elaborated in the next chapter. The use case for this is not very strong, as the data will be limited to non-editable Simple Features only, whereas accessing the same data through the ArcSDE Repository, allows the full geodatabase functionality including all of the advanced editing capabilities on complex dataset types like Geometric Networks etc. There is one exception though: if you are in need of read-only layers based on multiple table joins using complex SQL queries, that aren't easily realized within the geodatabase context, than using the Make Query Layer option in ArcMap to join Versioned Views of geodatabase layers with other database tables using standard ANSI/ISO SQL *can* make a lot of sense. The resulting dataset will be read-only.

As an alternative to using Versioned Views in non-ESRI software, it may be more convenient to access such data in a CAD package using the build-in - *if available in the CAD software* - capabilities to connect to an OGC WMS or (Transactional-)WFS webmapping service based on the same geodatabase data. Additionally, in case of AutoCAD, you can use an ESRI "Feature Service" created by "ArcGIS for Server" or "ArcGIS Spatial Data Server" in combination with ESRI's ArcGIS *for AutoCAD* software to edit data in an ESRI Geodatabase [12]. For details about editable webmapping services, see the following Help topic: Web editing with WFS services. More information about CAD / GIS integration can be found in the chapter "CAD integration in the ESRI Geodatabase Framework".

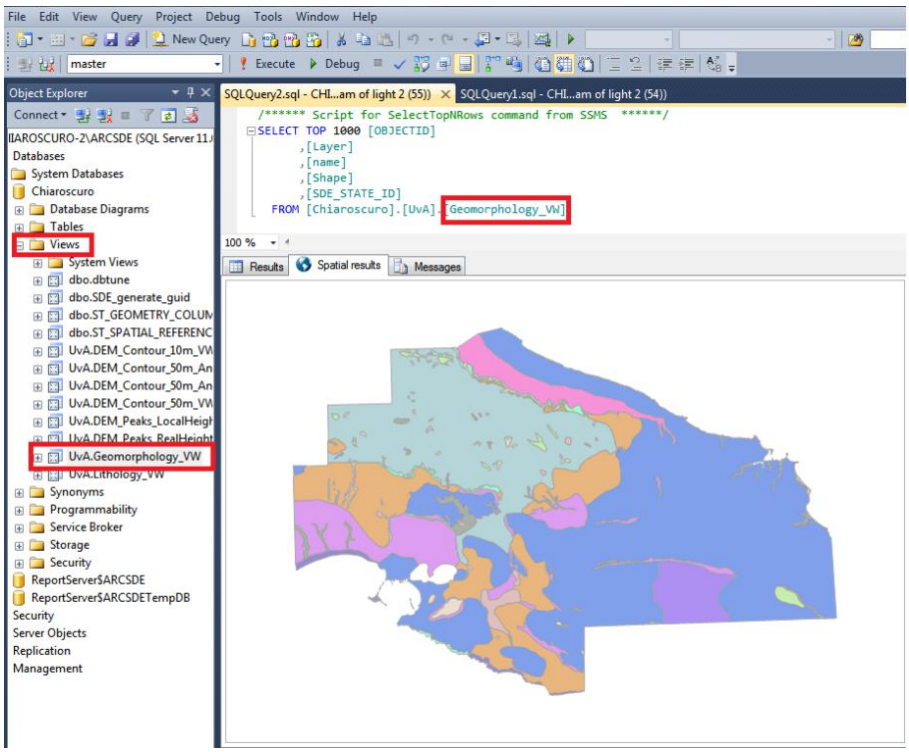


Fig 5: Versioned view of a geodatabase Feature Class data in SQL Server Management Studio, notice the " _VW" supplement to the Feature Class name.

As to the technical nature of Versioned Views, the ArcGIS Help says:

*"Versioned views incorporate **database views**, **stored procedures**, **triggers**, and **functions** to allow you to read or edit versioned data in a geodatabase table or feature class using Structured Query Language (SQL). When you query a versioned view, you can see the data in the base (business) table and the edits that are stored in the delta tables. **The triggers used by the versioned views update the delta tables when you edit the versioned view using SQL.**"*

The last highlighted sentence is also illustrated in the figure on the next page, that shows the SQL code of a Versioned View created by ArcGIS.

[12] ArcGIS for AutoCAD doesn't yet support true OGC WMS or WFS-T services

These automatically triggered functions ensure the integrity and health of the ArcSDE Repository and the geodatabase it forms part of. This is the mechanism associated with Versioned Views that allows edits to be made to geodatabase Feature Classes from applications that have no knowledge at all of the geodatabase structure.

For more details concerning editing of Versioned Views through a SQL client, see the following references:

- A quick tour of editing versioned data using SQL from the ArcGIS Help
- Working with the Geodatabase Using SQL (PDF). Older PDF from 2004 with many useful details. Despite its age still highly relevant, just take notice that "Multiversioned Views" is the name of "Versioned Views" prior to 10.1. Additionally, the specific behavior of Versioned Views regarding editing against the DEFAULT has changed, meaning editing against the DEFAULT is now supported without blocking concurrent edits in ArcGIS for Desktop.
- Working with the geodatabase effectively using SQL. PowerPoint presentation by Shannon Shields, Thomas Brown and Kevin Watt (ESRI) for the ESRI Developer Summit 2009.

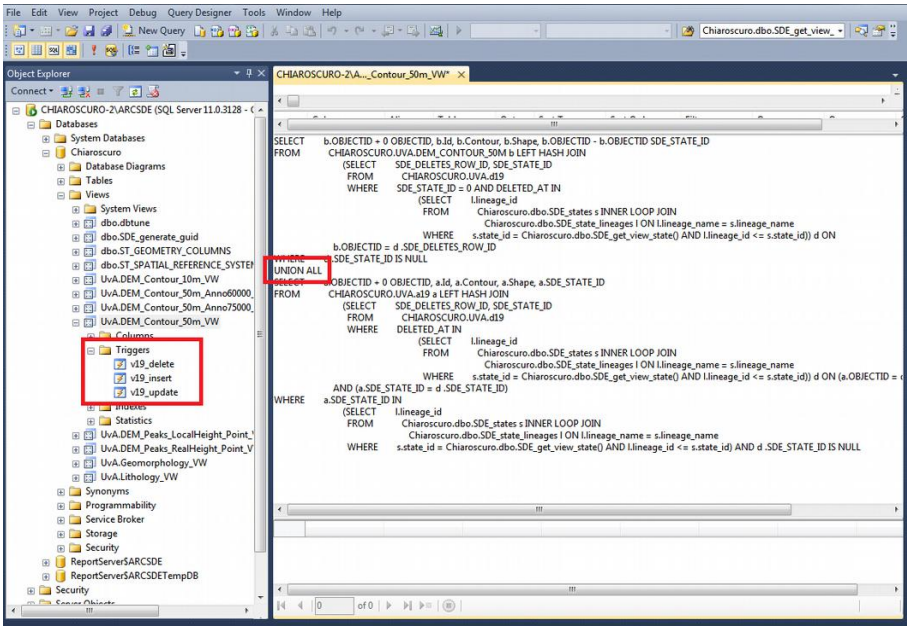
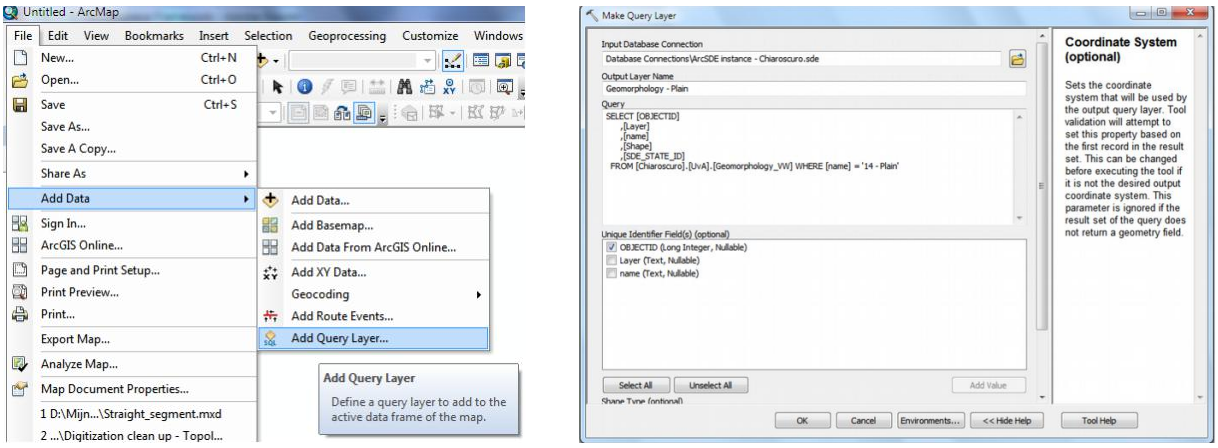


*Fig 6: Versioned View SQL code (right), note the Union All statement that ensures the base table and delta tables (**Adds** and **Deletes** tables) are combined. Also note the triggers (left) that are fired on insert , update, or delete of geometries.*

9 Direct SQL access

Direct SQL access [13], represented by the **purple lines** in the diagram, is a new option introduced first at 10.0. It encompasses the options to access spatial and non-spatial data in an enterprise database without an ArcSDE Repository by using arbitrarily defined SQL queries, similar to what is possible by means of SQL views inside an RDBMS. The resulting layers are called **Query Layers** and are read-only. ArcGIS for Desktop (ArcMap) offers a new **Add Query Layer** menu option and dialog, or the Make Query Layer (Data Management) tool for this purpose, that will create either a layer in the TOC of ArcMap in case of spatial data, or a stand-alone table in case of non-spatial data. The Direct SQL access option doesn't require an ArcSDE Repository, but it does require suitable ODBC database drivers installed. These drivers should be 32-bit, as ArcGIS still is 32-bit at version 10.1.

ArcGIS for Server can use Query Layers too. Once defined in ArcMap, create a Map Service of the map document for ArcGIS for Server to make the data available on the web.



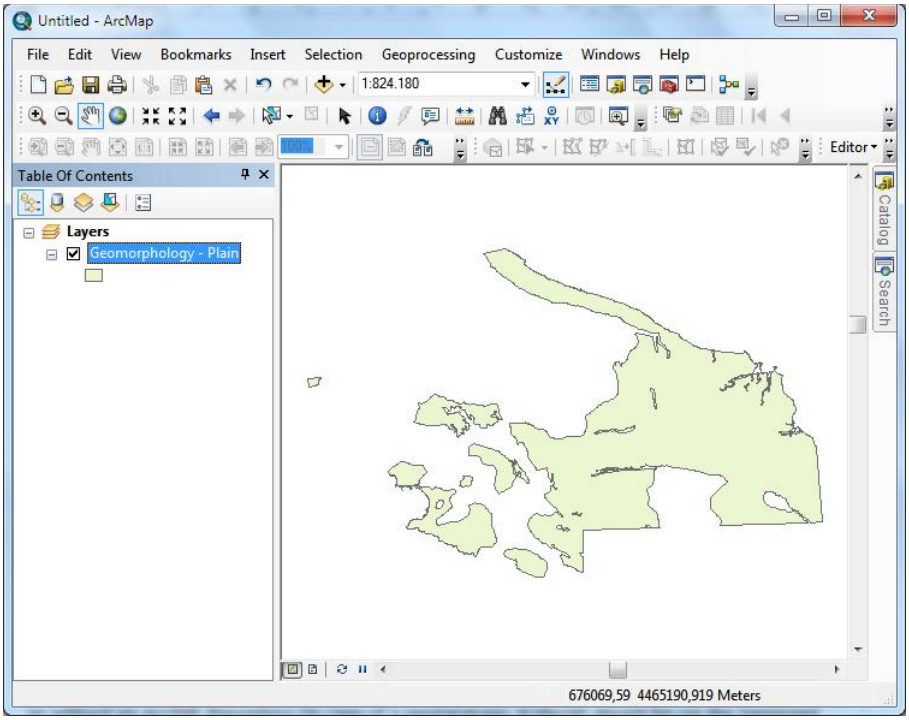


Fig 7: "Add Query Layer" menu option and "Make Query Layer" tool, and resulting dataset in ArcMap, a SQL selection of the 'Plain' type of the Geomorphology map shown in fig 5. This selection was made against a Versioned View in an ArcSDE geodatabase, but would more realistically have been made against an enterprise database without ArcSDE Repository, which is the primary use case for Direct SQL access and Query Layers.

[13] Not official ESRI terminology

You may have noticed there are a couple of purple connection lines in the diagram, that don't entirely correspond with ESRI's vision of Direct SQL access and Query Layers, e.g. the line running from the top right "SQL command line interface" to a Versioned View in the geodatabase. I have also classified the pink stippled Spatial Data Server (SDS) connection lines as "Direct SQL access". The main reason for this is that both of these options are virtually equivalent to what Query Layers in ArcGIS accomplish, that is, direct access to spatial data in an enterprise database without accessing and using the system data stored in the ArcSDE Repository[14].

An example of Direct SQL access to a Versioned View in an ArcSDE geodatabase, is shown below. It shows the Open Source **Quantum GIS** (QGIS) connecting to a Microsoft SQL Server stored geodatabase, accessing the Versioned View of a Feature Class, and making an edit to an existing geometry by adding an extra polygon part. Note that although this example is given as to what is potentially possible, one should be careful trying to create such interoperable workflows. There are some risks associated with this approach, since Object ID's and relationships with ArcSDE system tables need to be managed. Geodatabase Versioned Views take care of most of these issues via database triggers and stored procedures, as written in the previous chapter, but edits outside ArcGIS in untested or unsupported workflows might still result in corruption of the geodatabase if improperly used. In addition, ESRI warns not to edit any dataset participating in geodatabase behavior, so edits can essentially be made to Simple Feature layers only.

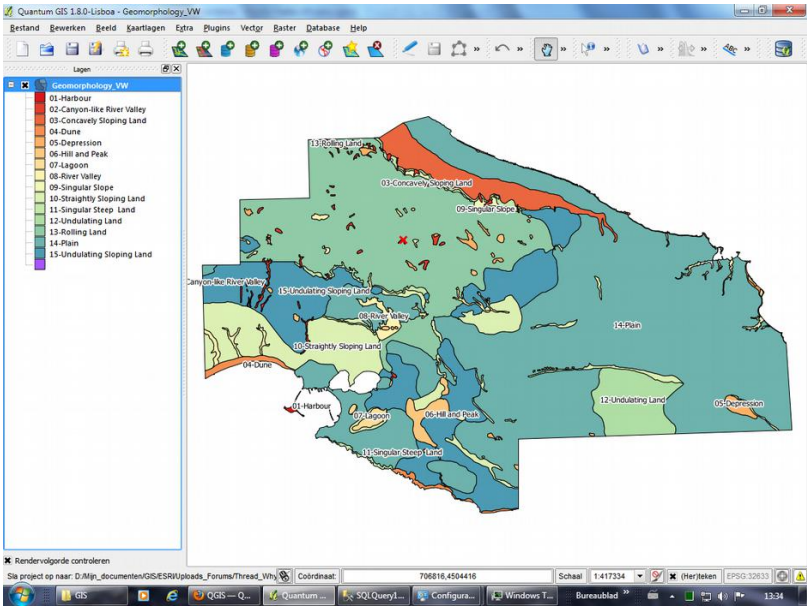


Fig 8: Quantum GIS (QGIS) accessing a Versioned View in SQL Server database.

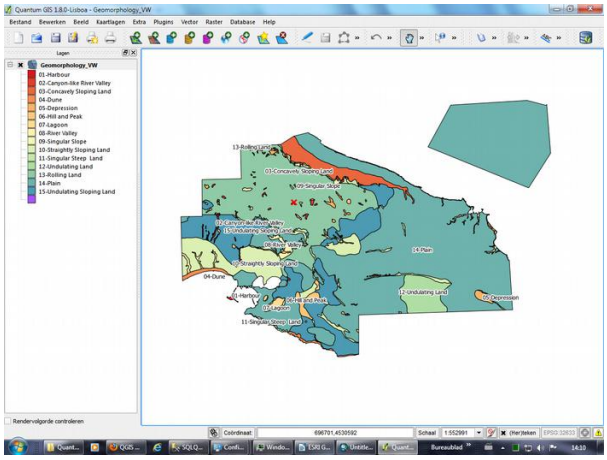


Fig 9: Versioned View layer edited in Quantum GIS (QGIS), polygon part added to an existing feature. Edit against Default version.

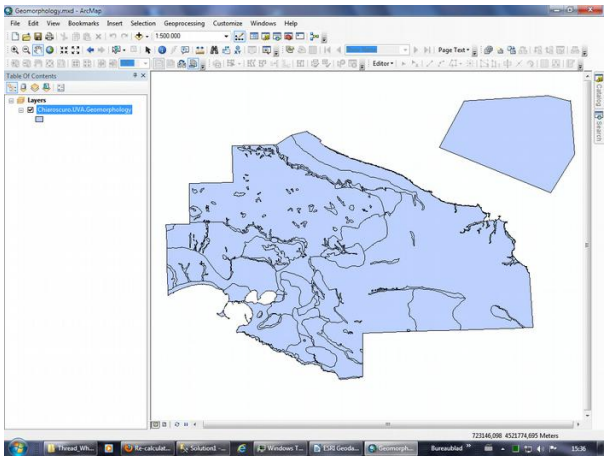


Fig 10: Same layer as figure 4, but now viewed in ArcMap, QGIS edit visible.

[14] In case of a geodatabase, Direct SQL access should always be via the Versioned Views to avoid corruption of the geodatabase.

For further reading regarding Query Layers: a small but nice introduction to Query Layers is given in the "Introduction to Query Layers" PDF based on a PowerPoint presentation by Thomas Brown (ESRI) at the 2011 Developer Summit. It seems likely ESRI will extend Query Layer functionality at some point in time to include the option of editing as well, although there are (logical) constraints concerning editable queries that will impact such potential future functionality. Not all SQL queries can be edited.

10 ArcGIS Spatial Data Server (SDS)

As of 10.1, ESRI has introduced a new product called **ArcGIS *Spatial Data Server*** (**SDS**). ArcGIS *Spatial Data Server*, according to ESRI, essentially acts as a kind of light-weight alternative for the combination of both "ArcSDE", and ArcGIS *for Server*, as it both supports accessing spatial data in a database, and serves webmapping services to the internet.

However, it should not be regarded as a full replacement, as there are many aspects and functionality of ArcGIS for Server lacking. E.g. it doesn't support the geodatabase's advanced functionality and data types, as it is limited to Simple Feature access only, and it doesn't support the long transaction versioned editing workflow like ArcGIS for Server does.

Using ArcGIS *Spatial Data Server*, you can access Versioned Views in an ESRI Geodatabase, and create ESRI Feature Services to edit them. The use case for this however, is not very strong as you will need to have an ArcSDE Repository enabled on your database to use it, in which case you might as well directly access the full geodatabase functionality using ArcGIS for Server.

More importantly though, *and this is primary reason ESRI developed SDS*, Spatial Data Server can access data and provide editing capabilities in a regular enterprise database *with* native spatial type but ***without*** an ArcSDE Repository. In the diagram this is illustrated by the pink stippled lines running from the ArcGIS Spatial Data Server, through the O-/JDBC drivers, to the lower right "Enterprise database with native spatial type". This option of SDS is restricted to Simple Features only though. Editing in this case also requires a Feature Service webmapping service to be created.

Whether serving data from an ESRI Geodatabase or a regular spatial enterprise database, editing could take place in ArcMap, a custom (web-)application, or any other GIS/CAD application supporting editing capabilities on Transactional Web Feature Services.

Currently, if editing through ArcMap, the editing takes place against a disconnected local copy of the Feature Service, stored in a local File Geodatabase created automatically when using the **Create Local Copy for Editing** menu option on the layer. The edits must than be synchronized back to the server once finished. The process for these steps is described in more detail here:

"Editing a feature service in ArcMap"

It may be a slight disappointment that no "direct" editing against the Feature Service is supported from ArcMap, but the disconnected nature also has its advantages in terms of robustness. Even for those mobile applications supporting "direct" editing of WFS-T type services, connection stability and ways to handle connection loss and recovery is a continuing concern. Clearly, you will always have a need to cache a small amount of data to be able to edit at all, even if it is just in volatile memory or "hidden" from the user in an Application Data folder. There may be more transparent options than the method chosen in ArcGIS for Desktop at this point in time, but there is some reason in this decision.

More information and limitations regarding ArcGIS Spatial Data Server can be found here:

"What is ArcGIS Spatial Data Server"

"A quick tour of authoring feature services for ArcGIS Spatial Data Server"

11 Will ArcGIS Spatial Data Server and Direct SQL access replace ArcSDE Technology?

A new question in the ESRI user community is whether ArcGIS Spatial Data Server and Direct SQL access (Query Layers) will replace ArcSDE... Ten years ago, when Oracle Spatial and OpenGIS Consortium standards were making their first serious headway into broad usage within the GIS community, people were asking similar questions regarding ArcSDE Technology. *"Oracle Spatial and the Open Source and OpenGIS standards make the proprietary ArcSDE obsolete"* was a common statement heard back then. ESRI however, instead did quite the opposite, and continued development to extend ArcSDE and ArcGIS to support the new "open" standards, in addition to significantly enhancing the geodatabase base model to support all kinds of new advanced data types and functionality. For those better informed, this could hardly have been a surprise. Even an Open Source, OpenGIS database, needs a software component managing GIS related (spatial) SQL transactions to and from the database to be accessible, and manage vital metadata related to the spatial database structure. And that was, and is, one of the primary roles of ArcSDE Technology in the ESRI product line.

Therefore, asking *"Is ESRI going to abandon ArcSDE Technology now ArcGIS Spatial Data Server and Direct SQL access options are available?"* is a bit similar to asking an innovative car manufacturer in the middle of designing its latest hybrid and electric car models whether *"Now we have E-Bikes and Segways, will you stop making cars?…"*

So yes, to me it sounds highly improbable that ESRI would decide to abandon or replace ArcSDE Technology any time soon. But apart from this rather personal opinion, let's delve into this a bit deeper from a technical point of view. There may be a surprise for you here coming up next...

A question that kept bothering me from the first time I heard about the new options, and read marketing and technical documentation about the options, is how ESRI implemented this functionality... Not having an ArcGIS Server / Spatial Data Server setup at hand, all I had to go by were the documents written by ESRI. Technical and Help documentation proved inconclusive in providing an answer, with ESRI merely stating for example (2011 ESRI International User Conference "Questions and Answers"):

"At ArcGIS 10, we introduced the concept of Query Layers. They enable ArcGIS users to have direct access to spatial types in a database, ***without the need for ArcSDE technology.****"*

This is a pretty bold and seemingly conclusive statement, suggesting ESRI somehow bypasses using ArcSDE Technology entirely for the new data access options. ***But:*** *is it true?!...* This last question kept nagging me, as my gut feeling told me it would be a waste of resources to rewrite proven and tested data access code from the ground up, unless there was a compelling reason to redevelop all data access functionality, like when ESRI decided to abandon Arc/Info workstation, allowing migration to more modern software development methods and platforms and opening up significant new capabilities. Such a reason did not seem to be here, with ArcSDE Technology and the associated API's already accessing databases in a pretty standard manner grounded in SQL, C++, C#, VB.Net and Java.

It seemed likely that SDS and Direct SQL access (Query Layers) actually shared a significant part of their code with ArcSDE Technology components, despite ESRI claiming it "does not use ArcSDE". In a sense, from a code base maintenance and economical point of view, it should be far more logical to have shared code to avoid technical and geometric inconsistencies and excess work. An alternative reason for using a separate code base might be the usage of Open Source components, developed in a loose collaborative way by expert users worldwide. Although ESRI has been "blamed" for years of creating "proprietary" "inaccessible" products and data, a quick look at the "Acknowledgments" PDF in the ArcGIS for Desktop installation folder, reveals ESRI's extensive usage of Open Source or third-party software components in ESRI products, especially for data access components (e.g. Raster readers). It spans a staggering 66 pages at 10.0...

To get a better feeling of what was going on, I decided to do a small but quite radical test, that is, to rename some of the DLLs in ArcGIS's "Bin" installation folder that I knew were directly involved in the ArcSDE Technology framework, and see what effect that would have on the ability of ArcGIS for Desktop to use Direct SQL access to access spatial data. I set up a layer using the **Make Query Layer** tool that accessed spatial layers in a native spatial type database stored in SQL Server 2012 Express, without enabling a geodatabase on this same test database. If ESRI's claims of not using ArcSDE Technology were true, than the drawing and access of these Query Layers should be unaffected by renaming the chosen DLLs.

The figure below shows the Query Layers added to ArcMap with no changes to filenames. The spatial layers are drawn without issues. Crucially highlighted is the DLL that I will rename in the next step, "**sdesqlsrvr101.dll**". It is one of the DLLs involved in 2-tier Direct Connect connections to the ArcSDE Repository and geodatabases in SQL Server databases.

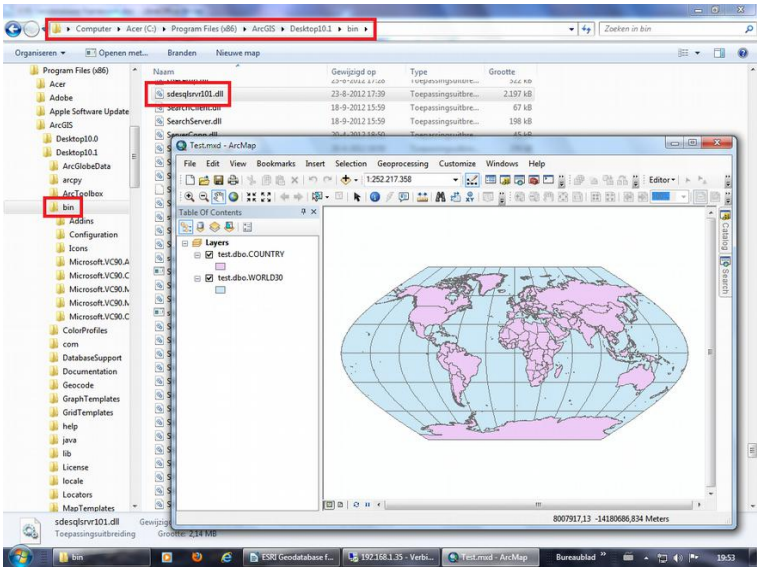


Fig 11: Database layers accessed through the Make Query Layer option. "sdesqlsrvr101.dll" file with it's original name in the "Bin" folder.

I was a bit anxious if ArcMap would start up at all, with one leg drawn from under it. Hitting the shortcut, it did! In the next figure, after renaming the highlighted DLL, you see the same datasets in the TOC, but with a question mark... The layers don't draw!

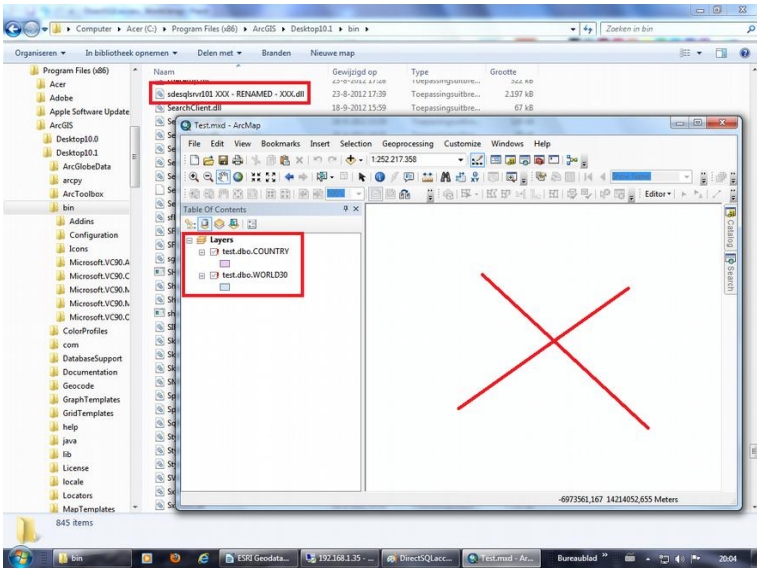


Fig 12: Native spatial type layers opened with Make Query Layer fail to draw when ArcSDE DLLs are inaccessible due to renaming.

Similar results are obtained if the "**gsrvrsql101.dll**" is renamed, these DLLs together implementing the "geographic server" functions so familiar from ArcSDE Application Servers and Direct Connect. This is pretty conclusive evidence that Query Layers, and with it most likely ArcGIS *Spatial Data Server*, are fully depended on ArcSDE Technology components for providing access to geometries in a database. For some, this may be a bit of a disappointment, to me, it is kind of comforting to know that ESRI used the data access components of ArcSDE Technology to realize these new options. It means Direct SQL access is virtually equivalent to 2-tier Direct Connect, except for the lack of dependency on the ArcSDE Repository / schema, and is solidly grounded in the well proven and tested ArcSDE Technology framework. The only difference with 2-tier Direct Connect, seems directly accessing spatial tables and extracting limited but crucial metadata from the features stored in the spatial fields, instead of reading system metadata from the ArcSDE Repository (e.g. spatial reference ID – SRID, layer extent, feature type etc.).

Trying to set the query by means of **Change Query** in the layer properties fails with a warning about inaccessibility of the database client, see below. Simply renaming the DLL back to its original name, and re-opening ArcMap, solves the issue and the layers are drawn again.

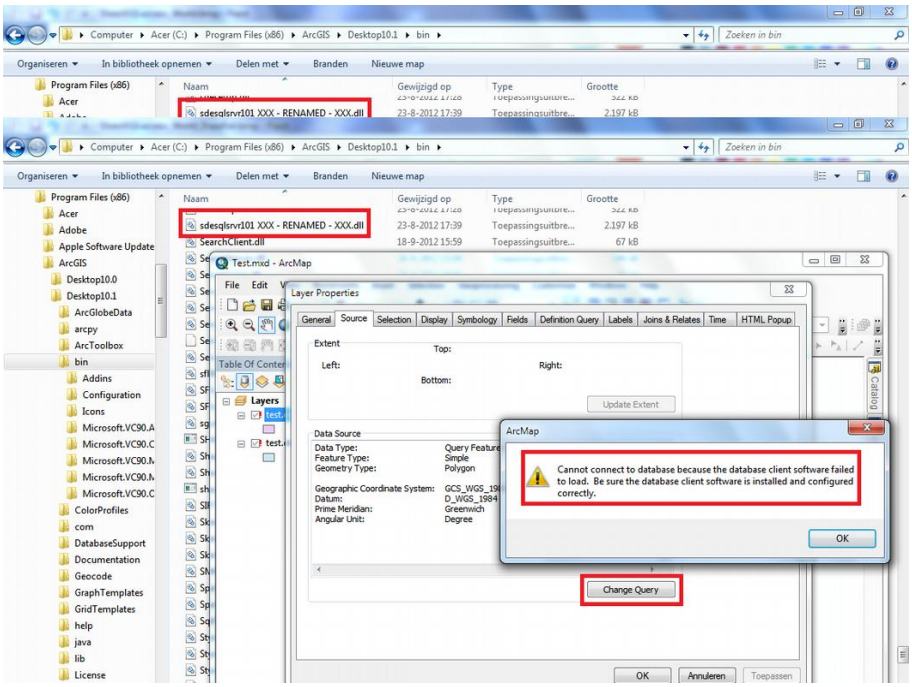


*Fig 13: Trying to set the base SQL query of a Query Layer using the **Change Query** option fails if the ArcSDE DLLs can't be accessed.*

As ArcGIS *Spatial Data Server* (SDS) is directly depended on Query Layers too, *even for geodatabase Feature Classes,* as illustrated by the fact that versioned Feature Classes from geodatabases are required to have associated Versioned Views in order to be useable in a SDS web service, this clearly indicates dependence of SDS on ArcSDE Technology too.

All in all, ESRI's claim of SDS not using ArcSDE needs refinement: ArcGIS Spatial Data Server - *and Query Layers* - don't need nor use an ArcSDE ***Repository***. But this isn't equivalent to "... *without the need for ArcSDE technology*", especially not so if components of the ArcSDE Technology framework are involved that have been around for well over a decade... and most likely will be the next decade.

12 The ArcSDE C/Java API

The ArcSDE C and Java APIs are Application Programming Interfaces targeted at developers wanting basic access to a geodatabase's geometries while retaining full awareness of the geodatabase's versioning model and allowing short and long transaction editing. It is a lower level access than ArcObjects, meaning any geodatabase logic implemented in ArcObjects is not available. Access to geometries is therefore more or less limited to Simple Features only, and no knowledge of Geometric Networks, Parcel Datasets etc. is available. While this may seem a serious limitation, the ArcSDE C and Java APIs have good found uses in specific fields and products, especially data conversion, where exchange of Simple Features often satisfies the basic need of data transfer between different systems. A good example is Autodesk's / OSGeo's "FDO Provider for ArcSDE" that will be discussed in chapter 14.

13 The ArcSDE *Command Line Tools* and the Geodatabase Administration toolset

The **ArcSDE *Command Line Tools*** are primarily used for trouble shooting and configuring (storage types) of an ESRI Geodatabase, especially modification of the DBTUNE and SERVER_CONFIG system tables. There are more tools for other geodatabase related management like data loading and user management. The ArcSDE Command Line Tools are a useful, but advanced and slightly difficult to use set of tools, mostly targeted at expert geodatabase managers or ArcSDE administrators. Being command line, usability for novice users is relatively low.

As a consequence of this, and a need by geodatabase administrators to easily and quickly perform day-to-day geodatabase management tasks and also have graphical feedback on things like state lineages and other interrelated geodatabase features, the demand for a graphical user interface (GUI) has been high. Finally, after years of requests by users, ESRI introduced a GUI at ArcGIS 10.1, in the form of the Geodatabase Administration dialogs and accompanying ArcToolbox toolset, meaning most day-to-day geodatabase management tasks can now be done using this dialog or the tools that form part of the new "Geodatabase Administration toolset (Data Management)" toolset. See the ArcGIS Help for detailed info on the ArcSDE Command Line Tools or Geodatabase Administration toolset.

14 CAD integration in the ESRI Geodatabase Framework

In the starting to become distant past, CAD and GIS used to be two entirely distinct fields in the world of geographic data management, with CAD focusing on construction and design, and GIS on more general small scale topographical data management and analysis. With the continuing push for closer collaboration between the fields, and integration of detailed large scale data in geodatabases, the question arises how to integrate CAD in the GIS workflow and the ESRI Geodatabase Framework.

Essentially, to answer the question above, there are three options:

- Data conversion of file based data sources. As this approach, although still in frequent use, is not part of "The Geodatabase Framework" as I see it, it will not be elaborated in this document.
- Integration via ArcSDE Technology on the CAD side. This may be a surprise for some, but yes, both Autodesk and Bentley use ArcSDE Technology in their products.
- Integration using spatial web services, which may be proprietary ESRI Map, Image or Feature services, or full OpenGIS WMS or transactional WFS-T, all types creatable by means of ArcGIS for Server.

Both Bentley, Autodesk and ESRI offer solutions for the latter two options, as visible in the diagram. Integration using ArcSDE Technology in CAD software, offers the advantage of a potential direct access to the source data in the geodatabase via the corporate LAN, and, depending on usage of just the ArcSDE C/Java API or ArcObjects, a lower or higher level of possible "geodatabase intelligence". Integration on the web services tier carries with it all the advantages of the independence of location offered by internet and WAN, allowing you to view, edit and post changes to a geodatabase from anywhere around the globe.

14.1 Integration via ArcSDE Technology on the CAD side

To start out with ArcSDE Technology on the CAD side, Autodesk offers an "FDO Provider for ArcSDE" as part of the AutoCAD Map 3D product family. This initially proprietary solution was made Open Source by Autodesk in 2006 and is now part of OSGeo's Open Source solutions for GIS. The FDO (Feature Data Objects) provider leverages the ArcSDE C / Java API's capabilities of accessing versioned datasets in a geodatabase, and therefore supports the long transaction versioned editing workflow, nicely integrating with concurrent editing in ArcGIS. It allows feature and attribute editing of data stored in a geodatabase, including the use of geodatabase defined attribute domains, using AutoCAD's standard editing tools. However, since it is based on the ArcSDE C API, it also means it is limited to Simple Features access only, so editing of any datasets involved in advanced geodatabase behavior, like Geometric Networks, is strongly discouraged. In addition, since it is based on a subset of the ArcSDE API, it doesn't support subtypes and annotation.

Bentley has the "ProjectWise Connector for ArcGIS" as part of the **Bentley Map** (MicroStation) and **ProjectWise** product family. This connector, contrary to the OSGeo's solution, is based on ArcObjects, and requires ArcGIS for Desktop installed on the same machine where the ProjectWise Connector for ArcGIS is installed. Since it is based on ArcObjects, it can potentially leverage the full geodatabase access capabilities ArcObjects has on offer, and as the documentation shows, it offers a wider integration with ESRI Geodatabase functionality, supporting Feature Dataset / Feature Class hierarchy, versioning, domains, subtypes and annotation. However, despite being based on ArcObjects and interfacing with software components of ArcGIS for Desktop, the current version is not entirely capable of directly using and editing an ESRI Geodatabase. Bentley uses an Extract, Modify, and Post process where the geodatabase spatial features are temporarily converted to Bentley's XML Feature Modeling (XFM) spatial format. This format *does* maintain all vital geodatabase characteristics in combination with the ProjectWise Connector, and this is how things like versioning, domains and subtypes are being supported. Once posted back to the geodatabase, modified features should thus be indistinguishable of

features created of modified in ArcGIS for Desktop, and fully integrate with the source Feature Class.

Comparing Autodesk's to Bentley's solution, one other aspect must be noted. Autodesk's solution is client PC based, and connections run more or less straight from the client PC to the database server. Bentley's ProjectWise Connector though, is a server based solution, and needs a rather complicated server based installation requiring multiple Bentley ProjectWise software components and possibly Bentley's **Geospatial Server**. In addition, a SQL Server installation for ProjectWise metadata storage, and Microsoft's IIS web server for running vital ProjectWise services are needed. In essence, ProjectWise and Geospatial Server fullfill a similar role for Bentley's products, as ArcGIS for Server does for the ESRI products. More information can be found here:

- Installing the ProjectWise Connector for ArcGIS
- Configuring the Connector

Both Autodesk's and Bentley's solutions require an ArcSDE connection available. 3-tier Application Server connections seem to be preferred, but from version 3.3.2 on, OSGeo's FDO Provider for ArcSDE supports 2-tier Direct Connect as well. Setting the provider up correctly for this though, requires some work and may prove an initial challenge, and may require overwriting the Autodesk provided FDO Provider with OSGeo's one, see these links for some more information regarding this topic:

Using Direct Connect with FDO Provider for ArcSDE

Top 5 List for Troubleshooting SDE Connections

Bentley's ProjectWise Connector for ArcGIS, being based on ArcObject, supports Direct Connect 2-tiers connections in a similar manner as ArcGIS for Desktop, and has the same requirements for setting up the connection. Some more information from Bentley is here, and the ArcGIS for Desktop Help should be of use too when setting up a Direct Connect connection.

14.2 Integration via web services – ArcGIS for AutoCAD

As for integration via spatial web services, ESRI has the **ArcGIS *for AutoCAD*** product on offer for AutoCAD users. Where in previous versions, the solution just offered basic read-only Image Services of available geodatabase datasets, the latest incarnations offer editable ESRI Feature Services as well. Using ArcGIS *for Server*, the Feature Service allows editing of an ESRI Geodatabase, including support for versioning, domains and subtypes, from within AutoCAD. A special property of ArcGIS for AutoCAD is that it bypasses ESRI's REST API that only allows access to Simple Features, instead using the SOAP API to allow ArcGIS for AutoCAD to exchange and edit complex data types like true curves.

Using ESRI's new ArcGIS *Spatial Data Server*, databases with native spatial type, but without an ArcSDE Repository, can be edited as well. Please note ArcGIS *Spatial Data Server*, at least at the current 10.1 release, does NOT support editing of an ESRI Geodatabase, so you will need the ArcGIS *for Server* product for that, as mentioned before. See "What is ArcGIS Spatial Data Server?" in the ArcGIS Help.

14.3 Integration via web services – OGC WMS and WFS services

Both Bentley Map and AutoCAD Map 3D are capable of accessing OGC WMS and WFS services in a read-only mode. So no transactional editing is supported on WFS services (WFS-T). Editing of an ESRI Geodatabase through web services is therefore currently only available for AutoCAD Map 3D users using ESRI's ArcGIS *for AutoCAD*, unless third-party plugins I do not know of offer this support.

AutoCAD Map 3D uses OSGeo's "FDO Provider for WFS" for the purpose of accessing WFS [15]. Actually, the FDO Provider for WFS, according to the AutoCAD Map 3D 2013 Online Help, *does* support basic editing operations on WFS services, but it seems Autodesk has yet to implement this same functionality in their own product. See the following excerpt from the Help:

"Using FDO Provider for WFS data manipulation operations, you can do the following:

Query features based on spatial and non-spatial constraints.

***Create** new feature instances.*

***Delete** feature instances.*

***Update** feature instances.*

Lock feature instances."

15 ArcGIS for Mobile

ArcGIS for Mobile constitutes a number of different applications and SDKs aimed at leveraging GIS capabilities in mobile environments, like mobile users via smart phones, iPads, etc., but also more technical uses in for example total stations for terrestrial surveying purposes. Most mobile editing applications are either based on editable web services (ESRI Feature Service, OGC WFS-T), as visible in the diagram, or disconnected editing using for example data stored in a file geodatabase, edits of which can be synchronized back to the enterprise geodatabase when back in office. Combinations of the two options are also possible.

The primary new development platform for this is ArcGIS Runtime, offering support for development on Windows, Linux, iOS (OS X expected for 10.2), Android, Windows Phone and Mobile. Geodatabase access is currently attribute and Simple Features only for File, Workgroup and Enterprise Geodatabases.

Two nice blog items about ArcGIS Runtime by ESRI Australia:

The New ArcGIS Runtime

A Bit More on ArcGIS Runtime (for Windows and Linux)

An alternative older development platform by ESRI is ArcPad and ArcPad Studio.

[15] Not included in the diagram, it is represented by the stippled green internet line from the Web server to the AutoCAD Map 3D product