

Controlling Multiple-Database Behavior

Understanding the `CROSS_DB_QUERY_FILTER` `SDE_dbtune` parameter for ArcSDE 8.2 geodatabases using Microsoft SQL Server 7 or 2000

A Microsoft SQL Server database is a container for objects that hold data, such as tables, and objects that manipulate data, such as triggers or stored procedures. Object security, transactions, and backup and recovery operations are all managed at the database level. The data stored in a database is usually related to a particular project or process.

An instance of SQL Server can support many databases. Before a user can access data stored within a particular database, they must be made a user in that database and be granted the appropriate object privileges. A connection to SQL Server must connect to a particular database, as no data can physically reside outside a database.

The architecture of an ArcSDE geodatabase is somewhat different. While an instance of SQL Server can support many databases, it can only support a single geodatabase. That geodatabase, however, can be made up of several SQL Server databases. The default behavior of ArcSDE for SQL Server has been to see feature classes and ArcSDE rasters inside a geodatabase, regardless of which database a user was physically connected to. With the release of ArcSDE 8.2, a new `SDE_dbtune` keyword allows you to modify this behavior.

The `CROSS_DB_QUERY_FILTER` Parameter

The `CROSS_DB_QUERY_FILTER` is a new parameter for the `DEFAULTS` `dbtune` keyword. This parameter has a value of 0 (zero) by default. Setting the value to 1 (one) will force ArcSDE connections to show only the data physically residing in the database they are connected to. This parameter applies to the entire ArcSDE geodatabase and anyone connecting to it.

Working With Multiple Databases—Default Behavior

In the example illustrated in Figure 1, the demo user owns data in the demo database and has permission to view data in the wilson database. When `CROSS_DB_QUERY_FILTER` is set to 0 (the ArcSDE default), the demo user can see feature classes and ArcSDE rasters in both the demo and wilson databases. However, the wilson user only has permission to see data that resides in the wilson database. What the demo user cannot see from his or her connection are tables, such as `wilson.wilson.CFCC`, which are visible only from connections made directly to the wilson database.

Working With Multiple Databases—`CROSS_DB_QUERY_FILTER = 1`

The following example illustrates the same two database connections when `CROSS_DB_QUERY_FILTER` has been set to 1. The `SDE_dbtune` parameter can be modified using the SQL statement shown in Figure 2. The ArcSDE server must be restarted for this new setting to take effect. After this parameter has been changed, the same connections are reopened as shown in Figure 3. Observe that now the demo user can only see data from the demo database, which is specified in his/her spatial database connection.

Deciding to Use `CROSS_DB_QUERY_FILTER`

Before you modify the default behavior in a multidatabase environment, it is important to consider how your users or applications access ArcSDE data. Modifying the default behavior can speed up layer list

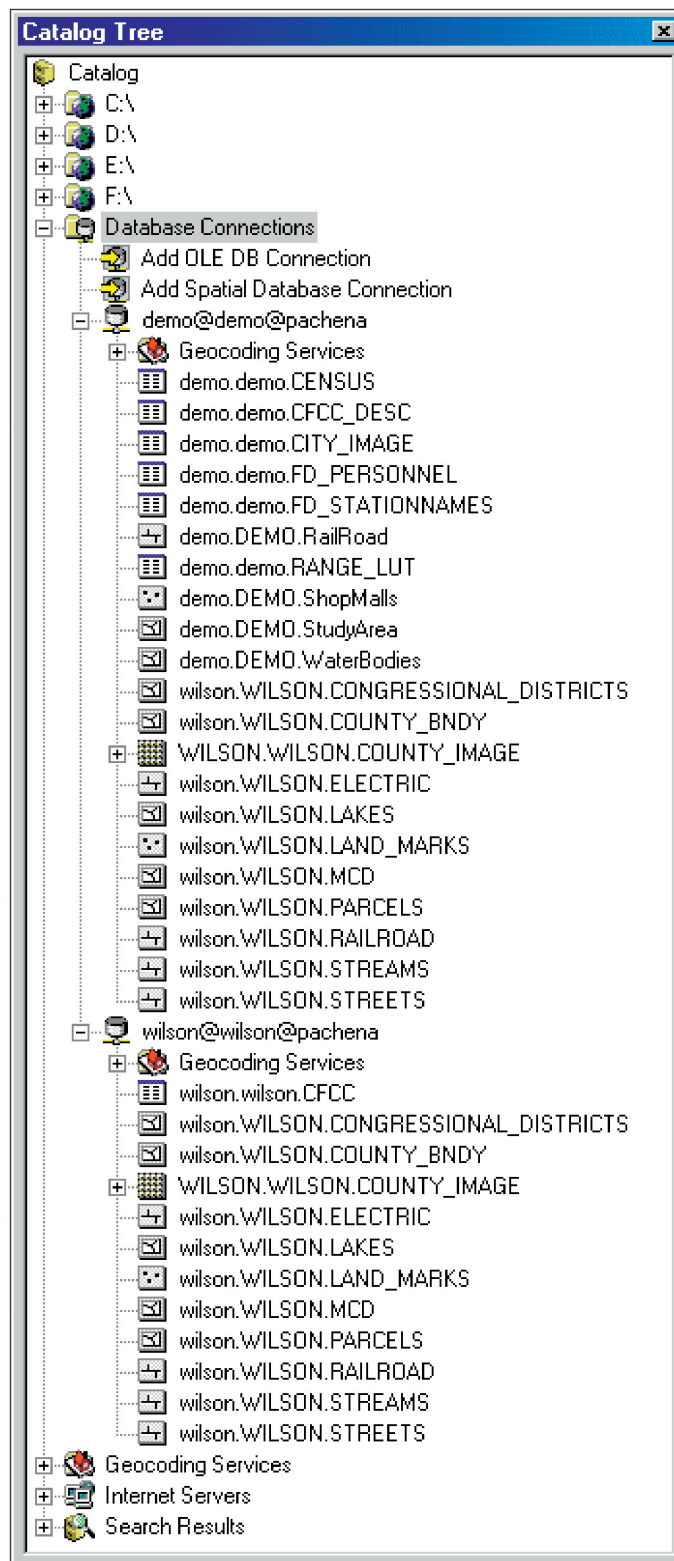


Figure 1: The demo user can view feature classes and rasters from the wilson database.

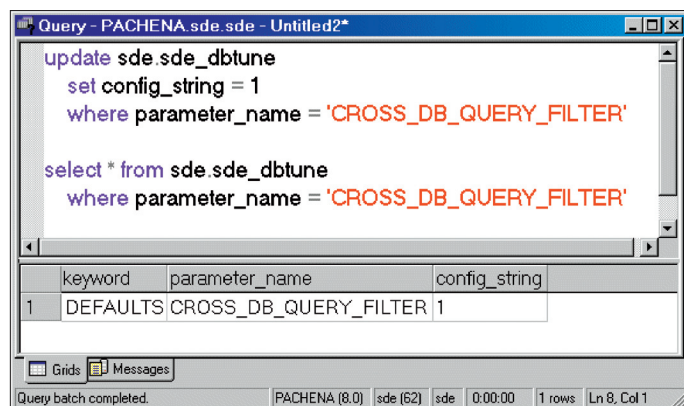


Figure 2: Modifying the SDE_dbtune parameter using a SQL statement.

times and reduce the number of feature classes that a user sees. If data is spread across multiple databases, but queried simultaneously by an application, such as an ArcIMS map service, using the default behavior can limit the number of separate gsrvr connections that are spawned.

If CROSS_DB_QUERY_FILTER is set to 1, each time data from a different database is queried a separate database connection must be created. In the examples referenced in this article, the wilson user had not been given permission to see any demo data. Consequently, regardless of the CROSS_DB_QUERY_FILTER setting, in all scenarios the wilson user was only able to see data residing in the wilson database. Effects similar to setting CROSS_DB_QUERY_FILTER to 1 can be achieved by controlling user privileges.

For more detailed information on the CROSS_DB_QUERY_FILTER parameter, see Chapter 2 of the *ArcSDE Configuration and Tuning Guide for Microsoft SQL Server*, located in the %sdehome%\documentation directory. [AU](#)

Moving to ArcGIS

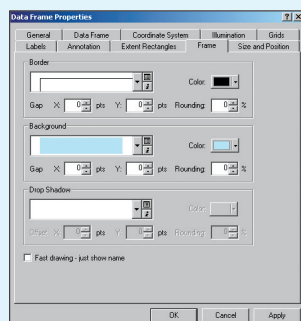
Modify a Map's Background Color

Changing the background color of a map from default white can be accomplished in just three mouse clicks.

1. Right-click anywhere on the map canvas and choose Properties from the context menu.

2. In the Data Frame Properties dialog box, click on the Frame tab.

3. In the Background section, click on the color swatch and choose a new color from the color palette.



4. Click OK to close the dialog box.

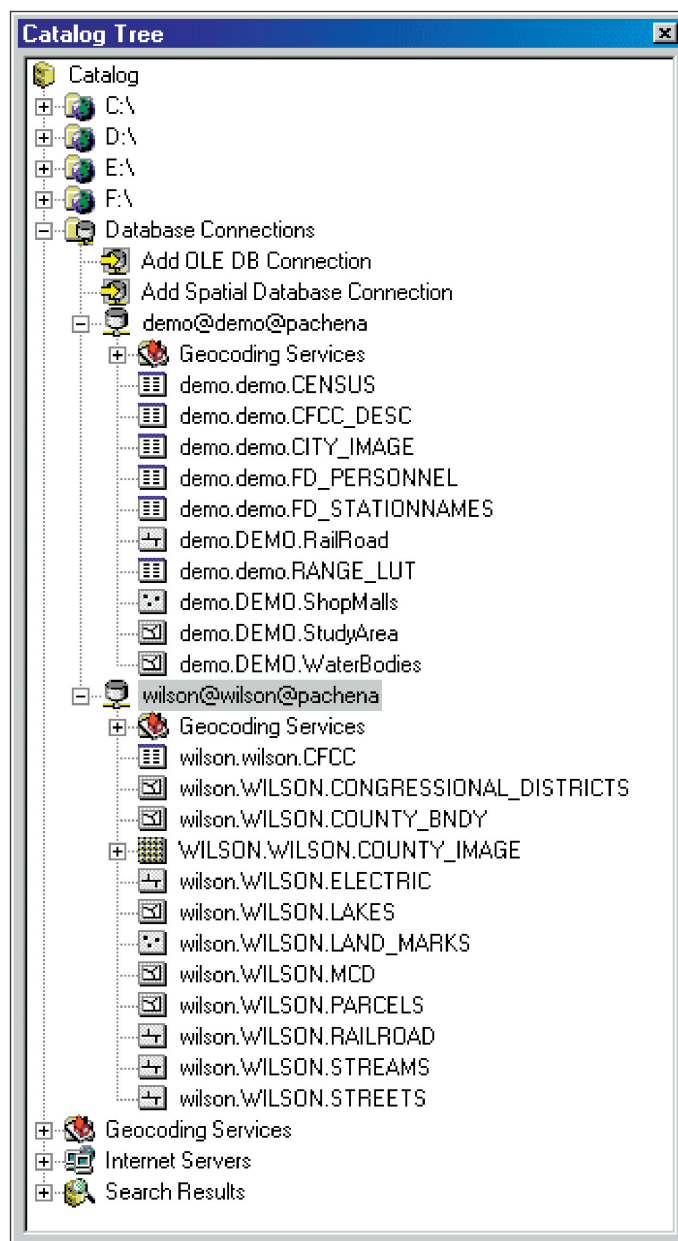


Figure 3: With CROSS_DB_QUERY_FILTER set to 1, the demo user now can see only the demo database data, as specified in his or her spatial database connection.