

Administering ArcSDE 8 Logfiles

By Daniel Lee and Greg Cunningham, ESRI Educational Services

ArcSDE uses logfiles to maintain temporary and persistent sets of selected records. When a selection is made in a client application, such as ArcMap, records are written to logfile tables for later use by the application. However, due to the heavy use of these files by ArcSDE, performance problems occasionally arise.

This article describes when and how logfile tables are created and some of the difficulties associated with them and outlines some methods for mitigating these problems. Although the discussion of logfile tables in this article pertains specifically to Microsoft SQL Server and Oracle, the logfile architecture is also applicable to the other relational database management systems (DBMSs) ArcSDE 8 supports—DB2 and Informix.

When Are Logfile Tables Created?

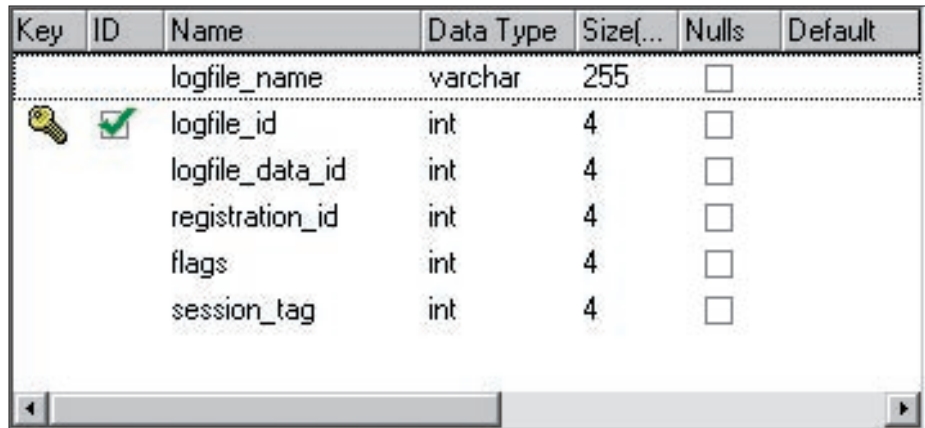
Every ArcSDE user, even a read-only user, requires two logfile tables—SDE_LOGFILES and SDE_LOGFILE_DATA. These tables are initially created when a user connects from an ArcSDE client for the first time, or are re-created, if they have been deleted, the next time a user connects.

The SDE_LOGFILES table stores information about each logfile that is created. The first two columns, logfile_name and logfile_id, are worth mentioning. The logfile_name uniquely identifies the name of the logfile, and the logfile_id links the SDE_LOGFILES table with the SDE_LOGFILE_DATA table. In the SDE_LOGFILE_DATA table, the sde_row_id column stores feature identifiers for the selected set, and the logfile_id column identifies when the selection was made.

Logfile Permissions

Because the logfile tables are created when a user connects with an ArcSDE client for the first time, each user requires DBMS permissions to create these tables. Permissions will differ depending on the underlying DBMS. For SQL Server users, ArcSDE stores and manages logfile tables in the SDE database (also called the ArcSDE repository). As a result, each user will need CREATE TABLE permissions to the ArcSDE database. Once the tables have been created, the CREATE TABLE permission should be revoked to prevent the user from writing to the SDE database.

An Oracle user will need the CREATE TABLE privilege in order to create logfile tables. These tables will be stored in the user's schema. The CREATE SESSION privilege is also required to connect to an Oracle database. Once the tables are created—when the user



Key	ID	Name	Data Type	Size(...)	Nulls	Default
		logfile_name	varchar	255	<input type="checkbox"/>	
	<input checked="" type="checkbox"/>	logfile_id	int	4	<input type="checkbox"/>	
		logfile_data_id	int	4	<input type="checkbox"/>	
		registration_id	int	4	<input type="checkbox"/>	
		flags	int	4	<input type="checkbox"/>	
		session_tag	int	4	<input type="checkbox"/>	

Figure 1: ArcSDE logfiles

connects—the CREATE TABLE system privilege can be revoked to prevent the user from creating any additional tables.

Difficulties With Logfile Tables

Potential problems with logfiles can come from several sources—from multiple individuals sharing the same user logfile; from very large logfiles, which can adversely affect performance; and from delete operations on very large logfile tables, which are expensive and can be problematic.

Problems With Contention

When several users log in as the same DBMS user, they share the same logfile. Not only does this cause logfiles to grow to an exceptionally large size, but it also introduces contention. Contention occurs because the logfile tables are used to store selection IDs from client applications. If multiple users log into ArcSDE as the same user, contention may arise if selection IDs are duplicated in the SDE_LOGFILE_DATA table. This can create unpredictable results and/or performance problems.

Large Files

The SDE_LOGFILE_DATA table holds selected sets of records for each user login, while SDE_LOGFILES stores the name of the logfile and the logfile ID. The entry in the SDE_LOGFILES table is deleted from the log when the selection is cleared. However, the delete from the SDE_LOGFILE_DATA is deferred until the user disconnects. This means that the SDE_LOGFILE_DATA table will continue to grow while a user is logged in and/or making selections in a client application. For example, ArcIMS users who stay connected for a long time or ArcMap users who perform large selections can generate extremely large SDE_

LOGFILE_DATA tables. As the table grows, it can also increase the size of the data files within the SDE database, in the case of a SQL Server, or in an Oracle user's tablespace.

Whenever a data file grows, delays will be experienced and performance impacted. Larger SDE_LOGFILE_DATA tables also cause longer queries when retrieving selected records. When a layer with selected features is rendered by ArcMap, two queries are submitted to ArcSDE. The first query redraws the layer, and the second query draws the selected features on top using different symbology.

Delete Operations

Deletions from the SDE_LOGFILE_DATA table only take place when users disconnect. However, if the SDE_LOGFILE_DATA table has grown extremely large (e.g., hundreds of thousands of records), a delete operation from this table can consume both time and considerable CPU and DBMS resources and greatly impact overall DBMS performance. In severe cases, the ArcSDE client may seem to disconnect, but the underlying DBMS, in the case of Oracle, does not release the gsrvr process (i.e., ArcSDE connection process). When this happens, the DBMS consumes as many CPU resources as possible and the gsrvr process must be killed manually and the user's SDE_LOGFILE_DATA table truncated.

The symptoms that indicate the SDE_LOGFILE_DATA table has grown too large depend on the underlying DBMS. In SQL Server, all user logfile tables are written to the SDE database. If selections are relatively small (e.g., several hundred), SQL Server is able to delete all the records in the SDE_LOGFILE_DATA. However, deleting a large number of records may take several hours. This inefficiency is due to the type of delete



Key	ID	Name	Data Type	Size(...)	Nulls	Default
		logfile_data_id	int	4	<input type="checkbox"/>	
		sde_row_id	int	4	<input type="checkbox"/>	

Figure 2: SDE_LOGFILE_DATA

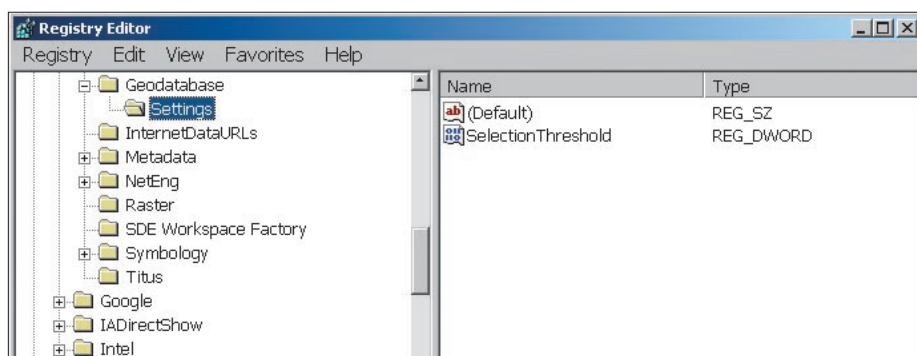


Figure 3: SelectionThreshold registration value

operation performed on the SDE_LOGFILE_DATA table. Rather than a truncate, a query is submitted for each record for the delete operation. If the same user logs into ArcSDE before the delete operation is complete and makes more selections, the SDE database will eventually grow to capacity (assuming the SDE database has a growth restriction applied). Once the database is full, selections can no longer be made. The inability to make further selections indicates that the SDE_LOGFILE_DATA table has grown too large.

Oracle database administrators (DBAs) will notice that, upon disconnect, Oracle will attempt to consume as many resources as possible in order to perform a delete of all the records in the logfile table. If the delete operation is not completed before the same user connects to ArcSDE, the gsrvt will resubmit the query to delete all the records. The already overconsumed CPU will now have to process another set of queries, which further impacts resources. The inability for the user to reconnect due to a lock on the logfile table indicates the SDE_LOGFILE_DATA table has grown too large.

Work-Arounds for ArcSDE 8 Logfiles

The following suggestions will help avoid logfile problems associated with multiple individuals sharing the same user logfile, very large logfiles, and delete operations on large logfiles.

Create Separate DBMS Login

Perhaps the most common problem with logfiles results from many users sharing the same logfile. Not only does this cause contention as previously described but also causes the SDE_LOGFILE_DATA table to grow excessively. Rather than having many users employing a single DBMS username/password, each user should have a separate DBMS login ensuring a separate logfile.

Changing the SelectionThreshold for ArcGIS Clients

By default, ArcGIS clients, such as ArcMap, will cache up to 100 selected records on the client machine. If selections exceed 100 features, ArcSDE will write the records to the SDE_LOGFILE_DATA table. If large

Continued on page 54

Administering ArcSDE 8 Logfiles

Continued from page 53

selections are frequently made (e.g., selections greater than 1,000), the SelectionThreshold can be changed to a higher value.

However, this results in more processing on the client end and can have performance impacts if the client machine lacks sufficient available RAM. If the SelectionThreshold is set to a value higher than the number of records that can be cached on the client machine, the application will need to write the selected set to the SDE logfiles. The client, which was trying to store the IDs in memory, now has to write the IDs to the DBMS. Consequently, the user may experience a delay while the client application tries to cache the data and then write to the DBMS.

The threshold value will be highly dependent on the number of features routinely being selected and the amount of available memory on client machines. More information on editing the SelectionThreshold can be found in the knowledge base article "HowTo: Increase performance displaying selected sets in ArcGIS sub-selected features from ArcSDE feature classes." To access it, go to the ESRI Support Center (support.esri.com) and search on 22668. This article recommends modifications to the Windows registry. *Note that editing registry settings can have adverse effects and should only be done when absolutely necessary.* Before modifying registry settings, make a backup of the registry.

Create an ON DELETE Trigger on SDE_LOGFILES

When a user clears a selection within a client application, such as ArcMap, the records from the SDE_LOGFILE tables are removed. However, the records in the SDE_LOGFILE_DATA table will remain until the user disconnects. Upon disconnection, the gsrvr will submit a query to delete all the records in the SDE_LOGFILE_DATA table. Alternatively, a trigger can be created on SDE_LOGFILES to delete entries in SDE_LOGFILE_DATA when the selected set is cleared. This will help reduce the size of the SDE_LOGFILE_DATA table.

For Oracle: Create a Global Temporary Table

Another method for reducing the size of the SDE_LOGFILE_DATA table for Oracle users is to create a temporary logfile table. In other words, rather than using the SDE_LOGFILE_DATA table, which is a permanent table, Oracle can create temporary tables to hold session-private data that exists only for the duration

of a transaction or session. The CREATE GLOBAL TEMPORARY TABLE statement creates a temporary table that can be transaction specific or session specific. For transaction specific temporary tables, data exists only for the duration of the transaction, while for session specific temporary tables, data exists for the duration of the session. With temporary tables, a TRUNCATE statement is issued on the table rather than a delete operation as is the case with an SDE_LOGFILE_DATA table.

Drop/Truncate Logfile Tables

If the SDE_LOGFILE_DATA table becomes exceptionally large, the table can simply be dropped or truncated by the DBA. Once the user disconnects, the DBA can drop the table using the DBMS drop table command or truncate the table using the ArcSDE command *sdelog -o clean*. If the DBMS drop table command is used, the SDE_LOGFILES and SDE_LOGFILE_DATA tables will be re-created the next time the user connects to

ArcSDE. ArcSDE command *sdelog -o clean* will simply truncate the logfile.

What's Ahead in ArcSDE 9?

The logfile issues currently experienced with ArcSDE 8 will be addressed with the release of ArcSDE 9. Instead of creating individual logfile tables, ArcSDE 9 will implement a pool of logfiles to store selected features. Session logfiles are dedicated to a single gsrvr rather than an individual ArcSDE user. Session logfiles dramatically reduce contention for logfiles because a single connection is using the logfile rather than each user sharing a logfile. Since only one connection is using the logfile, logfile tables will not grow as large. Finally, rather than deleting the rows in the SDE_LOGFILE_DATA table once the connection is terminated, selected features that are cleared from ArcGIS will have the corresponding ArcSDE logfile file entries removed immediately.



Let ArcMap Convert Map Units

ArcMap's editing tools will convert distance units for you as you edit features. If you are editing data that was recorded in distance units that differ from the units specified by the Data Frame's coordinate system, you can avoid converting the nonconforming measurements. Simply type in the measurement and the abbreviation for that distance unit. The editing tools in ArcMap will correctly convert the distance into the unit used by the specified coordinate system. Table 1 provides a list of these abbreviations.

Unit	Abbreviation
meter	m
centimeter	cm
kilometer	km
foot	ft
inch	in
yard	yd
mile	mi
nautical mile	nm
chain	ch
link	lk
rod	rd
U.S. survey foot	ftus
U.S. survey chain	chus
U.S. survey link	lkus
U.S. survey rod	rdus
U.S. survey mile	mius

Table 1: Distance unit abbreviations