



Best Practices for Storing the GIS Data ReViewer Workspace in an Enterprise Geodatabase for Oracle®

Copyright © 2009 ESRI
All rights reserved.
Printed in the United States of America.

The information contained in this document is the exclusive property of ESRI. This work is protected under United States copyright law and other international copyright treaties and conventions. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or by any information storage or retrieval system, except as expressly permitted in writing by ESRI. All requests should be sent to Attention: Contracts and Legal Services Manager, ESRI, 380 New York Street, Redlands, CA 92373-8100 USA.

The information contained in this document is subject to change without notice.

ESRI, the ESRI globe logo, ArcSDE, ArcCatalog, ArcMap, ArcGIS, PLTS, www.esri.com, and @esri.com are trademarks, registered trademarks, or service marks of ESRI in the United States, the European Community, or certain other jurisdictions. Other companies and products mentioned herein may be trademarks or registered trademarks of their respective trademark owners.

Best Practices for Storing the GIS Data ReViewer Workspace in an Enterprise Geodatabase for Oracle

An ESRI White Paper

Contents	Page
Introduction.....	1
SDE DBTUNE.....	1
Disk Configuration.....	2
Reduce Disk I/O Contention.....	2
Step 1: Data Files	3
Step 2: REV User.....	3
Step 3: Modify DBTUNE.....	4
GIS Data ReViewer 9.3	4
GIS Data ReViewer 9.3.1	4
Step 4: ReViewer Workspace	5
GIS Data ReViewer 9.3	5
GIS Data ReViewer 9.3.1	6
Step 5: Set the Feature Dataset and Tables as Versioned	6
Step 6: Verify Storage.....	6
Step 7: Restore DBTUNE.....	6
GIS Data ReViewer 9.3	6
GIS Data ReViewer 9.3.1	7

Contents	Page
Step 8: Permissions and Roles	7
Tips for Granting Permissions	7
Step 9: Log File Tables	7
Create Log File Tablespaces	8
Change DBTUNE Log File Parameters.....	8
Create Log File Tables.....	8
Step 10: ArcSDE User	8
Editor User	8
Viewer User	9
Conclusion	9

Best Practices for Storing the GIS Data ReViewer Workspace in an Enterprise Geodatabase for Oracle

Introduction

GIS Data ReViewer, a data quality control management application, requires that you have a workspace defined when using a review session to store the anomalies found during your data review process. It creates a new feature dataset and tables in the geodatabase that you have identified as your ReViewer workspace. The feature dataset includes point, line, and polygon feature classes, which contain error geometries for the features that have been written to the ReViewer table. The ReViewer workspace can be stored in any existing geodatabase, including the production geodatabase, or a separate file or personal geodatabase created specifically for the data review process. When stored in an enterprise geodatabase, the ReViewer workspace supports versioning. This white paper is intended for database administrators and will help them establish the GIS Data ReViewer workspace in an enterprise geodatabase for Oracle[®]. The enterprise geodatabase uses ArcSDE[®] technology as the gateway between GIS clients and Oracle.

SDE DBTUNE

DBTUNE storage parameters allow you to control how ArcSDE technology creates objects within an Oracle database. They allow you to determine such things as how to allocate space to a table or index and which tablespace a table or index is to be created in, as well as other Oracle-specific storage attributes. They also allow you to specify one of the available storage formats for the geometry of a spatial column.

The DBTUNE storage parameters are stored in the DBTUNE table. The DBTUNE table, along with all other metadata tables, is created during the setup phase that follows the installation of ArcSDE technology. ArcSDE installation creates a dbtune file under the etc directory from which the DBTUNE table is populated. If no dbtune file is present during setup, the DBTUNE table will be populated with default values.

If a large number of database connections are accessing the same files in the same location on disk, database performance will be slower because the connections are competing with one another for the same resources. To reduce this competition, you can store database files in different locations on disk.

Thus, DBTUNE can be modified to store the ReViewer feature dataset and tables into separate data files across different locations on disk. This will lead to reduced disk contention and improved database I/O.

Standard GIS storage recommendations favor keeping index and log files separate from vector and tabular business tables. For performance reasons, it is better to position the business, feature, and spatial index tables separately and to position tablespace data files based on their usage pattern. For a multiversioned, highly active editing geodatabase, database files of VERSIONS tablespace may be separated and dispersed across available disks to avoid I/O contention.

Disk Configuration

Large-production enterprise geodatabase systems should employ a hardware striping solution. Your best disk and data organization strategies involve spreading your data across multiple disks. With data spread across multiple disks, more spindles actively search for it. This can increase disk read time and decrease disk contention. However, too many disks can slow down a query. There are two main ways of achieving striping: tablespaces and RAID. You can also combine the two—create tablespaces within disk arrays. You can employ data segregation strategies; keeping tables from indexes or certain types of tables from other tables will improve performance and alleviate administrative burdens.

Suggested Oracle optimal configuration is as follows:

- DISK 0—Oracle/App Software DISK 0
- DISK 1—SYSTEM, Control File 1
- DISK 2—RBS, TEMP, Control File 2
- DISK 3—REDO 1,2,3, Export Files
- DISK 4—Feature Data Tables
- DISK 5—Spatial Index Data Tables
- DISK 6—Attribute Data/Business Tables
- DISK 7—Oracle Indexes

Reduce Disk I/O Contention

As a rule, you should create database files as large as possible, based on the maximum amount of data you estimate the database will contain, to accommodate future growth. By creating large files, you can avoid file fragmentation and get better database performance. In many cases, you can let data files grow automatically; just be sure to limit autoextend by specifying a maximum growth size that leaves some hard disk space available. By putting different tablespaces on different disks, you can also help eliminate physical fragmentation of your files as they grow.

Below is a suggested design to reduce disk I/O contention:

File Type	Database Activity	Move File to Disk With
Redo log	Frequent edits	Relatively low I/O
Redo log	Few or no edits	Moderate I/O
Undo log files	Frequent edits	Low I/O but separate from redo log files
System data	Frequent edits	Moderate I/O
Temporary tablespace	Few edits	High I/O

Step 1: Data Files

Create new tablespaces to store the ReViewer feature dataset and tables.

```
TABLESPACE    SDE_PARAMETER
REV_BDATA     Business table
```

```

REV_BINDEIX    Business table index
REV_FDATA     Feature table
REV_FINDEX    Feature table index
REV_SDATA     Spatial Index table
REV_SINDEX    Spatial Index table index
REV_ADATA     Adds table (versioned)
REV_AINDEX    Adds table index
REV_DDATA     Deletes table (versioned)
REV_DINDEX    Deletes table index
    
```

```

CREATE SMALLFILE TABLESPACE "REV_BDATA"
DATA FILE 'D:\oracle\ORADATA\REV\rev_Bdata01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K SEGMENT SPACE MANAGEMENT AUTO;

CREATE SMALLFILE TABLESPACE "REV_BINDEIX"
DATA FILE 'D:\oracle\ORADATA\REV\rev_Bindex01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K SEGMENT SPACE MANAGEMENT AUTO;

CREATE SMALLFILE TABLESPACE "REV_FDATA"
DATA FILE 'D:\oracle\ORADATA\REV\rev_FDATA01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K SEGMENT SPACE MANAGEMENT AUTO;

CREATE SMALLFILE TABLESPACE "REV_FINDEX"
DATA FILE 'D:\oracle\ORADATA\REV\rev_FINDEX01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K SEGMENT SPACE MANAGEMENT AUTO;

CREATE SMALLFILE TABLESPACE "REV_SDATA"
DATA FILE 'D:\oracle\ORADATA\REV\rev_Sdata01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K SEGMENT SPACE MANAGEMENT AUTO;

CREATE SMALLFILE TABLESPACE "REV_SINDEX"
DATA FILE 'D:\oracle\ORADATA\REV\rev_Sindex01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K SEGMENT SPACE MANAGEMENT AUTO;

CREATE SMALLFILE TABLESPACE "REV_ADATA"
DATA FILE 'D:\oracle\ORADATA\REV\rev_Adata01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K SEGMENT SPACE MANAGEMENT AUTO;

CREATE SMALLFILE TABLESPACE "REV_AINDEX"
DATAFILE 'D:\oracle\ORADATA\REV\rev_Aindex01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K SEGMENT SPACE MANAGEMENT AUTO;

CREATE SMALLFILE TABLESPACE "REV_DDATA"
DATAFILE 'D:\oracle\ORADATA\REV\rev_Ddata01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K SEGMENT SPACE MANAGEMENT AUTO;

CREATE SMALLFILE TABLESPACE "REV_DINDEX"
DATAFILE 'D:\oracle\ORADATA\REV\rev_Dindex01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K SEGMENT SPACE MANAGEMENT AUTO;
    
```

Step 2: REV User

Create a new database user to store the ReViewer feature dataset and tables and grant the appropriate permissions.

```

CREATE USER REV PROFILE DEFAULT IDENTIFIED BY rev
DEFAULT TABLESPACE "REV_BDATA"
TEMPORARY TABLESPACE "TEMP"
QUOTA UNLIMITED ON "REV_BDATA"
QUOTA UNLIMITED ON "REV_BINDEIX"
QUOTA UNLIMITED ON "REV_FDATA"
QUOTA UNLIMITED ON "REV_FINDEX"
QUOTA UNLIMITED ON "REV_SDATA"
QUOTA UNLIMITED ON "REV_SINDEX"
QUOTA UNLIMITED ON "REV_ADATA"
QUOTA UNLIMITED ON "REV_AINDEX"
QUOTA UNLIMITED ON "REV_DDATA"
QUOTA UNLIMITED ON "REV_DINDEX"
ACCOUNT UNLOCK;

GRANT CREATE SESSION TO REV;
GRANT CREATE SEQUENCE TO REV;
GRANT CREATE TRIGGER TO REV;
GRANT CREATE VIEW TO REV;
GRANT CREATE TABLE TO REV;
    
```

Step 3: Modify DBTUNE

Export the dbtune file before making any modification.

```
sdedbntune -o export -f dbtune_exp.sde -u sde -p sde -i 5151
```

Copy *dbtune_exp.sde* to *dbtune_rev.sde*.

**GIS Data
ReViewer 9.3**

Modify the *dbtune_rev.sde* `##DEFAULTS` configuration keyword.

dbtune_rev.sde

```
##DEFAULTS
A_INDEX_RASTER          "PCTFREE 0 INITRANS 4 TABLESPACE REV_AINDEX NOLOGGING"
A_INDEX_XML             "PCTFREE 0 INITRANS 4 TABLESPACE REV_AINDEX NOLOGGING"
D_INDEX_STATE_ROWID    "PCTFREE 0 INITRANS 4 TABLESPACE REV_DINDEX STORAGE (INITIAL 409600) NOLOGGING "
D_INDEX_DELETED_AT     "PCTFREE 0 INITRANS 4 TABLESPACE REV_DINDEX STORAGE (INITIAL 409600) NOLOGGING "
B_STORAGE              "PCTFREE 0 INITRANS 4 TABLESPACE REV_BDATA STORAGE (INITIAL 409600) "
B_INDEX_XML            "PCTFREE 0 INITRANS 4 TABLESPACE REV_BINDEX NOLOGGING"
B_INDEX_USER           "PCTFREE 0 INITRANS 4 TABLESPACE REV_BINDEX STORAGE (INITIAL 409600) NOLOGGING "
B_INDEX_TO_DATE        "PCTFREE 0 INITRANS 4 TABLESPACE REV_BINDEX NOLOGGING"
B_INDEX_SHAPE          "PCTFREE 0 INITRANS 4 TABLESPACE REV_BINDEX NOLOGGING"
B_INDEX_ROWID          "PCTFREE 0 INITRANS 4 TABLESPACE REV_BINDEX STORAGE (INITIAL 409600) NOLOGGING "
B_INDEX_RASTER         "PCTFREE 0 INITRANS 4 TABLESPACE REV_BINDEX STORAGE (INITIAL 409600) NOLOGGING "
GEOMETRY_STORAGE      "ST_GEOMETRY"
F_STORAGE              "PCTFREE 0 INITRANS 4 TABLESPACE REV_FDATA "
F_INDEX_LEN            "PCTFREE 0 INITRANS 4 TABLESPACE REV_FINDEX NOLOGGING"
F_INDEX_FID            "PCTFREE 0 INITRANS 4 TABLESPACE REV_FINDEX NOLOGGING"
F_INDEX_AREA           "PCTFREE 0 INITRANS 4 TABLESPACE REV_FINDEX NOLOGGING"
D_STORAGE              "PCTFREE 0 INITRANS 4 TABLESPACE REV_DDATA STORAGE (INITIAL 409600) "
ST_GEOM_LOB_STORAGE   " STORE AS (TABLESPACE REV_FDATA ENABLE STORAGE IN ROW CHUNK 8K RETENTION
CACHE INDEX (TABLESPACE REV_FINDEX) ) "
S_STORAGE              "PCTFREE 0 INITRANS 4 TABLESPACE REV_SDATA STORAGE (INITIAL 409600) "
S_INDEX_SP_FID        "PCTFREE 0 INITRANS 4 TABLESPACE REV_SINDEX NOLOGGING"
S_INDEX_ALL            "PCTFREE 0 INITRANS 4 TABLESPACE REV_SINDEX STORAGE (INITIAL 409600) NOLOGGING "
A_STORAGE              "PCTFREE 0 INITRANS 4 TABLESPACE REV_ADATA STORAGE (INITIAL 409600) "
A_INDEX_USER           "PCTFREE 0 INITRANS 4 TABLESPACE REV_AINDEX STORAGE (INITIAL 409600) NOLOGGING "
A_INDEX_ROWID          "PCTFREE 0 INITRANS 4 TABLESPACE REV_AINDEX STORAGE (INITIAL 409600) NOLOGGING "
A_INDEX_SHAPE          "PCTFREE 0 INITRANS 4 TABLESPACE REV_AINDEX STORAGE (INITIAL 409600) NOLOGGING "
A_INDEX_STATEID       "PCTFREE 0 INITRANS 4 TABLESPACE REV_AINDEX STORAGE (INITIAL 409600) NOLOGGING "
END
```

Import the modified *dbtune_rev.sde* file.

```
sdedbtune -o import -f dbtune_rev.sde -u sde -p sde -i 5151
```

**GIS Data
ReViewer 9.3.1**

From the GIS Data ReViewer 9.3.1 release forward, you can select the DBTUNE configuration keyword on the *ReViewer Workspace Properties* dialog box.

Instead of modifying `##DEFAULTS`, you can simply create a new DBTUNE keyword following the steps below:

- Copy the `##DEFAULTS` keyword and paste it at the end of the *dbtune_rev.sde* file.
- Rename it `##REVIEWER` and change the tablespace name for the appropriate parameters.

```
##REVIEWER
A_INDEX_RASTER          "PCTFREE 0 INITRANS 4 TABLESPACE REV_AINDEX NOLOGGING"
...
A_INDEX_STATEID        "PCTFREE 0 INITRANS 4 TABLESPACE REV_AINDEX STORAGE (INITIAL 409600) NOLOGGING "
END
```

- Import the modified *dbtune_rev.sde* file.

**Step 4: ReViewer
Workspace**

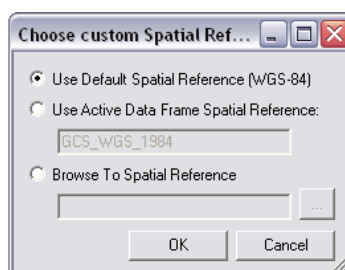
Create a database connection in ArcCatalog™ with the REV user; this will be the ReViewer workspace location.

GIS Data ReViewer 9.3

In ArcMap™, click the ReViewer Session Manager button on the PLTS™ GIS Data ReViewer toolbar:

- Browse to the ReViewer workspace location.
- Navigate to the database connection created in ArcCatalog.
- Click New to start a new ReViewer session.

The *Choose custom Spatial Reference* dialog box appears.



Choose custom Spatial Reference

- Choose the custom spatial reference.

When you click *Browse To Spatial Reference*, the New Spatial Reference wizard appears, which allows you to choose the spatial reference you want to use with the ReViewer feature dataset.

- Click OK.

This automatically creates the ReViewer feature dataset and tables.

- The ID and Name fields are automatically populated in the Session area.
- If necessary, type a custom name for the current ReViewer session in the Name text box.

By default, the name matches the ID.

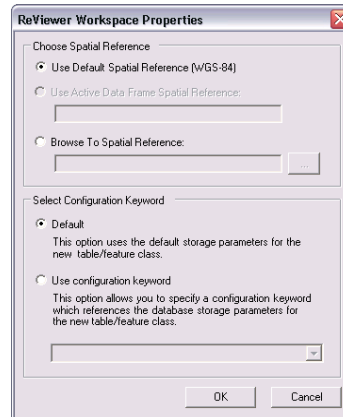
- If you are working with an enterprise database, click the ReViewer Dataset Version drop-down arrow and choose the version to be used.
- Click Start Session.

The button appears to be selected, and its name changes to End Session.

- Click Close.

GIS Data ReViewer 9.3.1

When you click New to start a new session in GIS Data ReViewer 9.3.1, the *ReViewer Workspace Properties* dialog box appears and you can select the DBTUNE configuration keyword. Select the ##REVIEWER configuration keyword created in step 3.



ReViewer Workspace Properties

Step 5: Set the Feature Dataset and Tables as Versioned

After the new session is created, the REV user must register the following as versioned in ArcCatalog:

- RevDataset (feature dataset)
- RevTableMain table
- RevTableLocation table

Step 6: Verify Storage

Run the SQL queries below to verify that the ReViewer workspace was created under the correct tablespaces:

```
sqlplus rev/rev@rev

--TABLES--
SELECT TABLE_NAME, TABLESPACE_NAME, STATUS FROM USER_TABLES ORDER BY 1;
--INDEXES--
SELECT INDEX_NAME, TABLE_NAME, TABLESPACE_NAME, STATUS FROM USER_INDEXES ORDER BY 2,1;
--LOBS--
SELECT * FROM USER_LOBS WHERE SEGMENT_NAME LIKE 'SYS_LOB%' ORDER BY TABLE_NAME;
```

If any tables or indexes are stored in the wrong tablespace, then ALTER TABLE and ALTER INDEX can be used to change the tablespace. See the SQL syntax.

```
ALTER TABLE <table_name> MOVE TABLESPACE <tablespace_name>;
ALTER INDEX <index_name> REBUILD TABLESPACE <tablespace_name>;
```

If moving large objects (LOB), please read Oracle MetaLink Doc ID: 130814.1, "How to move LOB Data to Another Tablespace."

Step 7: Restore DBTUNE

Once the ReViewer feature dataset and tables are created, you can restore the original DBTUNE parameters.

GIS Data ReViewer 9.3

Import the original dbtune file.

```
Sdedbtune -o import -f dbtune_exp.sde -u sde -p sde -i 5151 -D rev
```

***GIS Data
ReViewer 9.3.1***

Skip this step if you have created the `##REVIEWER` DBTUNE configuration keyword.

**Step 8: Permissions
and Roles**

Grant permissions to the REV tables through Oracle roles using the script below:

```
SET SERVEROUTPUT ON;

spool Roles_rev.sql;
DROP ROLE "REVEDITOR";
CREATE ROLE "REVEDITOR" NOT IDENTIFIED;
DROP ROLE "REVIEWER";
CREATE ROLE "REVIEWER" NOT IDENTIFIED;
select 'grant select on ' || owner || '.' || table_name || ' to REVIEWER;'
from sys.dba_tables where lower(owner) = 'rev' order by table_name;
select 'grant select,insert,update,delete on ' || owner || '.' || table_name || ' to REVEDITOR;'
from sys.dba_tables where lower(owner) = 'rev' order by table_name;
spool off;
SET SERVEROUTPUT ON;
/
@Roles_rev.sql;
/
```

Then, simply grant the REVEDITOR role to ArcSDE editor users and the REVIEWER role to ArcSDE viewer users.

***Tips for Granting
Permissions***

Grant user only select, update, insert, and delete privileges to the RevAdminCustomFields table if the user has privileges to modify the database schema.

Grant user only select, update, insert, and delete privileges to the RevAdminDescriptions table if you want the user to be able to add customized error descriptions. Note that all other users accessing the ReViewer workspace will see the customized error descriptions.

**Step 9: Log File
Tables**

Enterprise geodatabases use log file tables to maintain lists of selected records. Records are written to log file tables for later use by the application whenever a selection of a specific size is made, a reconciliation or post on a versioned database is performed, or a disconnected editing checkout is done in a client application. The log file tables store the ObjectIDs of the selected features so they can be redisplayed. This allows faster analysis and processing of information.

In ArcGIS®, by default, log file tables are used if the selection set contains 100 or more records. This selection threshold of 100 features is set in the registry. It can be changed; however, ESRI does not recommend doing so. There is no proven performance reason for changing it, and doing so could even cause performance problems. Thus, log file tables store feature selections in ArcMap that are greater than 100 for each connected ArcSDE editor/viewer user. Hence, it is recommended to store the log file tables in a separate tablespace; this can be achieved with the DBTUNE table.

Log file options are set using specific parameters in the SERVER_CONFIG and DBTUNE tables. Parameters in these tables are altered using the sdeconfig and sdedbtune commands, respectively.

Create Log File Tablespaces

```
CREATE SMALLFILE TABLESPACE SDELOGFILE
DATA FILE 'D:\oracle\ORADATA\REV\SDE\sdelogfile01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 800M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 512K SEGMENT SPACE MANAGEMENT AUTO;
```

```
CREATE SMALLFILE TABLESPACE SDELOGFILEIDX
DATA FILE 'D:\oracle\ORADATA\REV\SDE\sdelogfileidx01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M LOGGING
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 512K SEGMENT SPACE MANAGEMENT AUTO;
```

Change DBTUNE Log File Parameters

- Export the DBTUNE table.

```
sdebtune -o export -f dbtune_logfile.sde -u sde -p sde -i 5151
```

- Modify the dbtune_logfile.sde ##LOGFILE_DEFAULTS configuration keyword.

```
##LOGFILE_DEFAULTS
LD_INDEX_DATA_ID "PCTFREE 0 INITRANS 4 TABLESPACE SDELOGFILEIDX NOLOGGING "
LF_INDEXES "PCTFREE 0 INITRANS 4 TABLESPACE SDELOGFILEIDX NOLOGGING "
LF_STORAGE "PCTFREE 0 INITRANS 4 TABLESPACE SDELOGFILE"
SESSION_INDEX "PCTFREE 0 INITRANS 4 TABLESPACE SDELOGFILEIDX NOLOGGING "
SESSION_TEMP_TABLE 0
SESSION_STORAGE "PCTFREE 0 INITRANS 4 TABLESPACE SDELOGFILE"
LD_STORAGE "PCTFREE 0 INITRANS 4 TABLESPACE SDELOGFILE"
LD_INDEX_ROWID "PCTFREE 0 INITRANS 4 TABLESPACE SDELOGFILEIDX NOLOGGING "
END
```

- Import the modified dbtune_logfile.sde.

```
sdebtune -o import -f dbtune_logfile.sde -u sde -p sde -i 5151
```

Create Log File Tables

- Grant QUOTA on SDELOGFILE and SDELOGFILEIDX to the SDE editor/viewer user.

```
ALTER USER <user_name> QUOTA UNLIMITED ON "SDELOGFILE";
ALTER USER <user_name> QUOTA UNLIMITED ON "SDELOGFILEIDX";
```

- Grant CREATE TABLE permission for the SDE editor/viewer user.
- In ArcMap, select more than 100 features; this automatically creates the log file tables.
- Remove CREATE TABLE permissions as appropriate.

You can find out more about ArcSDE log file tables by visiting ArcGIS Desktop 9.3 Help, under Geodatabases and ArcSDE > Administering ArcSDE geodatabases > Configuring an ArcSDE geodatabase/Log file configuration options.

Step 10: ArcSDE User

The example below shows how to create an editor and viewer ArcSDE user:

Editor User

```
CREATE USER GIS_EDITOR PROFILE "DEFAULT"
IDENTIFIED BY editor
DEFAULT TABLESPACE "USERS"
TEMPORARY TABLESPACE "TEMP" ACCOUNT UNLOCK;

GRANT "CONNECT" TO "GIS_EDITOR";
GRANT CREATE TABLE TO "GIS_EDITOR";
GRANT "GISEDITOR" TO "GIS_EDITOR";
/*-- GISEDITOR role has SELECT, INSERT, UPDATE and DELETE permission on the GIS data (featureclasses, tables, etc.)
GRANT "REVEDITOR" TO "GIS_EDITOR";
ALTER USER GIS_EDITOR QUOTA UNLIMITED ON "SDELOGFILE";
ALTER USER GIS_EDITOR QUOTA UNLIMITED ON "SDELOGFILEIDX";
```

Viewer User

```
CREATE USER GIS_VIEWER PROFILE "DEFAULT"  
IDENTIFIED BY viewer  
DEFAULT TABLESPACE "USERS"  
TEMPORARY TABLESPACE "TEMP" ACCOUNT UNLOCK;  
GRANT "CONNECT" TO "GIS_VIEWER";  
GRANT CREATE TABLE TO "GIS_VIEWER";  
GRANT "GISVIEWER " TO "GIS_VIEWER";  
/*-- GISVIEWER role has SELECT permission on the GIS data (featureclasses, tables, etc.)  
GRANT "REVVIEWER" TO "GIS_VIEWER";  
ALTER USER GIS_VIEWER QUOTA UNLIMITED ON "SDELOGFILE";  
ALTER USER GIS_VIEWER QUOTA UNLIMITED ON "SDELOGFILEIDX";
```

Conclusion

You can reduce disk contention and improve database I/O by storing the GIS Data ReViewer workspace in different locations on disk. However, this practice alone does not guarantee optimal database performance, and additional tuning tasks may need to be taken into consideration.

You can find out more about the recommended tuning tasks in ArcGIS Desktop 9.3 Help, under Geodatabases and ArcSDE > Administering ArcSDE geodatabases > Tuning an ArcSDE geodatabase.

See ArcGIS Desktop 9.3 Help at www.esri.com/webhelp93 or visit www.esri.com/support.

For more information on GIS Data ReViewer, visit www.esri.com/datareviewer or e-mail datareviewer@esri.com.



About ESRI

For four decades, ESRI has been helping people make better decisions through management and analysis of geographic information. Our culturally diverse staff work with our business partners and hundreds of thousands of people who use GIS to make a difference in our world.

A full-service GIS company, ESRI offers support for implementing GIS technology from the desktop to enterprise-wide servers, online services, and mobile devices. GIS solutions are flexible and customizable to meet the needs of all our users.

Our Focus

At ESRI, we focus on promoting the value of GIS and its applications throughout the world and pay close attention to our users' needs. Our software development and services respond to our customers with products that are easy to use, flexible, and integrated. Our technology is multidisciplinary, productive, and valuable to our users.

We have a strong commitment to educating our customers through ESRI's various training programs. ESRI is a socially conscious business and invests heavily in issues regarding education, conservation, sustainable development, and humanitarian affairs.

Contact ESRI

1-800-GIS-XPRT (1-800-447-9778)

Phone: 909-793-2853

Fax: 909-793-5953

info@esri.com

www.esri.com

Offices worldwide

www.esri.com/locations



ESRI
380 New York Street
Redlands, California
92373-8100 USA