

Best Practices for Storing the ArcGIS® Data Reviewer Workspace in an Enterprise Geodatabase for Oracle

An Esri® White Paper
November 2012



Copyright © 2012 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, ArcGIS, ArcSDE, ArcCatalog, ArcMap, esri.com, and @esri.com are trademarks, service marks, or registered marks of Esri in the United States, the European Community, or certain other jurisdictions. Other companies and products or services mentioned herein may be trademarks, service marks, or registered marks of their respective mark owners.

Best Practices for Storing the ArcGIS Data Reviewer Workspace in an Enterprise Geodatabase for Oracle

An Esri White Paper

Contents	Page
Introduction.....	1
ArcSDE DBTUNE.....	1
Disk Configuration.....	2
Reducing Disk I/O Contention.....	2
Transparent Data Encryption	3
Step 1: Store Data Files.....	5
Step 2: Create the REV User	6
Step 3: Modify DBTUNE	6
ArcGIS Data Reviewer	6
Step 4: Configure Oracle Parameters.....	7
Step 5: Configure ArcSDE Parameters.....	7
Step 6: Create the Reviewer Workspace.....	7
ArcGIS Data Reviewer	8
Step 7: Set the Feature Dataset and Tables as Versioned.....	9
Step 8: Verify Storage.....	9
Step 9: Grant Permissions and Roles	9
Tips for Granting Permissions	10

Contents	Page
Step 10: Set Log File Table Options.....	10
Create Log File Tablespaces.....	10
Change DBTUNE Log File Parameters.....	10
Create Log File Tables.....	10
Step 11: Create ArcSDE Users	11
Editor User	11
Viewer User	11
Replication	11
Conclusion	11

Best Practices for Storing the ArcGIS Data Reviewer Workspace in an Enterprise Geodatabase for Oracle

Introduction ArcGIS® Data Reviewer is a data quality control management application that is supplied as an extension for ArcGIS for Desktop and ArcGIS for Server. It requires that you define a workspace to store the anomalies found during the data review process. ArcGIS Data Reviewer 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 to help database administrators establish the ArcGIS Data Reviewer workspace in an enterprise geodatabase for Oracle. The enterprise geodatabase uses ArcSDE® technology as the gateway between geographic information system (GIS) clients and Oracle.

ArcSDE DBTUNE DBTUNE storage parameters let you control how ArcSDE technology creates objects within an Oracle database. They allow you to determine things such 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 let you specify one of the available storage formats for the geometry of a spatial column.

The DBTUNE storage parameters are stored in the DBTUNE table, which, along with all other metadata tables, is created during the setup phase that follows the installation of ArcSDE. 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 in separate data files across different locations on disk. This will reduce disk contention and improve database input/output (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 position tablespace data files based on their usage pattern. For a multiversioned, highly active editing geodatabase, database files of the 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. The best strategies for disk and data organization involve spreading your data across multiple disks. With data spread across multiple disks, more spindles will actively search for it. This can increase disk read time and decrease disk contention. However, having too many disks can slow down a query. There are two main ways of achieving striping: tablespaces and redundant array of independent disks (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.

A suggested Oracle optimal configuration is as follows:

- Disk 0—Oracle/Application 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

Reducing 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 tablespaces on different disks, you can also minimize the 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

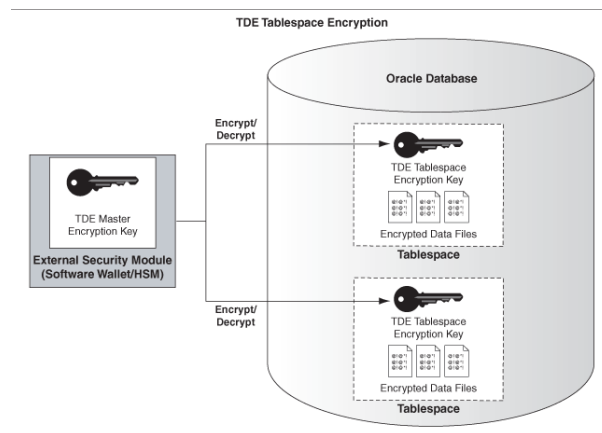
Transparent Data Encryption

Transparent data encryption (TDE) enables you to encrypt sensitive data—such as credit card numbers—stored in tables and tablespaces. Encrypted data is transparently decrypted for a database user or application that has access to data. TDE helps protect data stored on media in the event that the storage media or data file gets stolen.

Oracle Database uses authentication, authorization, and auditing mechanisms to secure data in the database but not in the operating system data files where data is stored. To protect these data files, Oracle Database provides TDE. To prevent unauthorized decryption, TDE stores the encryption keys in a security module outside the database.

These are some benefits of using transparent data encryption:

- Security administrators can be sure that sensitive data is safe in case the storage media or data file gets stolen.
- Implementing TDE helps you address security-related regulatory compliance issues.
- You do not need to create triggers or views to decrypt data for the authorized user or application. Data from tables is transparently decrypted for the database user and application.
- Database users and applications need not be aware of the fact that the data they are accessing is stored in encrypted form, since data is transparently decrypted.
- Applications need not be modified to handle encrypted data. Data encryption and decryption are managed by the database.
- Key management operations are automated. The user or application does not need to manage encryption keys.



TDE Tablespace Encryption

See Oracle documentation on how to configure TDE tablespace encryption:
http://download.oracle.com/docs/cd/E11882_01/network.112/e10746/asotrans.htm#ASOAG9579

```
*-----*
--Configure TDE Oracle Enterprise Manager - OEM
*-----*/

--Create the wallet folder
mkdir C:\oracle\admin\wallets

OEM > login as sys / sysdba

OEM > Server > Transparent Data Encryption

Advanced Options > Change Location

Host Credentials
Username: avworld\dbs_ora
Password: xxxxxxxx

Configuration Method: File System

Encryption Wallet Directory: C:\oracle\admin\wallets

OK

Create Wallet > Local Auto-Open Wallet > Create

Host Credentials
Username: avworld\dbs_ora
Password: xxxxxxxx

Wallet Password: walletadmin

Continue

--Backup the wallet folder
cd C:\oracle\admin

zip -r wallets wallets

/*-----*
--Configure TDE Manually
*-----*/

--Create the wallet folder
mkdir C:\oracle\admin\wallets

--Add wallet location to sqlnet.ora

ENCRYPTION_WALLET_LOCATION =
(SOURCE =
(METHOD = FILE)
(METHOD_DATA =
(DIRECTORY = C:\oracle\admin\wallets\${ORACLE_SID}
)
)
)

Note: The default encryption wallet location is $ORACLE_BASE/admin/<global_db_name>/wallet.
If you want to let Oracle manage a wallet in the default location then there is no need to
set the ENCRYPTION_WALLET_LOCATION parameter in sqlnet.ora.

--Generate a master key

alter system set encryption key identified by "walletadmin";

--See the status of the wallet

select * from v$encryption_wallet;

--Make the wallet auto login

set ORACLE_SID=revdb

orapki wallet create -wallet C:\oracle\admin\wallets -auto_login -pwd walletadmin

--Backup the wallet folder

cd C:\oracle\admin

zip -r wallets wallets
```


Step 1: Store Data Files

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

```

TABLESPACE      ArcSDE_PARAMETER
REV_BDATA       Business table
REV_BINDEXT     Business table index
REV_FDATA       Feature table
REV_FINDEX      Feature table index
REV_SDATA       Spatial Index table
REV_SINDEX      Spatial Index table index
REV_ADAT        Adds table (versioned)
REV_AINDEX      Adds table index
REV_DDAT        Deletes table (versioned)
REV_DINDEX      Deletes table index

ALTER SYSTEM SET WALLET OPEN IDENTIFIED BY "revadmin";

CREATE SMALLFILE TABLESPACE "REV_BDATA" ENCRYPTION default storage (ENCRYPT)
DATAFILE 'D:\oracle\ORADATA\REVDB\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_BINDEXT" ENCRYPTION default storage (ENCRYPT)
DATAFILE 'D:\oracle\ORADATA\REVDB\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" ENCRYPTION default storage (ENCRYPT)
DATAFILE 'D:\oracle\ORADATA\REVDB\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" ENCRYPTION default storage (ENCRYPT)
DATAFILE 'D:\oracle\ORADATA\REVDB\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" ENCRYPTION default storage (ENCRYPT)
DATAFILE 'D:\oracle\ORADATA\REVDB\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" ENCRYPTION default storage (ENCRYPT)
DATAFILE 'D:\oracle\ORADATA\REVDB\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_ADAT" ENCRYPTION default storage (ENCRYPT)
DATAFILE 'D:\oracle\ORADATA\REVDB\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" ENCRYPTION default storage (ENCRYPT)
DATAFILE 'D:\oracle\ORADATA\REVDB\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_DDAT" ENCRYPTION default storage (ENCRYPT)
DATAFILE 'D:\oracle\ORADATA\REVDB\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" ENCRYPTION default storage (ENCRYPT)
DATAFILE 'D:\oracle\ORADATA\REVDB\rev_Dindex01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K SEGMENT SPACE MANAGEMENT AUTO;

```

By setting the data files' initial size to 10 MB, there is no delay in the creation of the tablespaces; to avoid fragmentation, you can resize the data files.

```

ALTER DATABASE DATAFILE 'D:\oracle\ORADATA\REVDB\rev_Bdata01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\oracle\ORADATA\REVDB\rev_Bindex01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\oracle\ORADATA\REVDB\rev_Fdata01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\oracle\ORADATA\REVDB\rev_Findex01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\oracle\ORADATA\REVDB\rev_Sdata01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\oracle\ORADATA\REVDB\rev_Sindex01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\oracle\ORADATA\REVDB\rev_Adata01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\oracle\ORADATA\REVDB\rev_Aindex01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\oracle\ORADATA\REVDB\rev_Ddata01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\oracle\ORADATA\REVDB\rev_Dindex01.dbf' RESIZE 400M;

```

Step 2: Create the 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_BINDEXT"
  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.

```
sdedbtune -o export -f dbtune_exp.sde -u sde -p sde -i sde:oracle11g:revdb
```

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

ArcGIS Data Reviewer

Create a new DBTUNE keyword following the steps below:

Copy the `##DEFAULTS` keyword and paste it at the end of the file named *dbtune_rev.sde*.

Rename it `##REVIEWER` and change the tablespace name for the appropriate parameters.

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

If your database only stores the Reviewer repository, you can edit `##DEFAULTS`; otherwise, create a new configuration keyword as described above.

Import the modified *dbtune_rev.sde* file.

```
sdedbtune -o import -f dbtune_rev.sde -u sde -p sde -i sde:oracle11g:revdb
```

Step 4: Configure Oracle Parameters

It is recommended that you use the following parameter values when creating an Oracle database:

Oracle Parameters for Reviewer

Parameter Name	Value
Configure with Database Enterprise Manager	Enabled
Automatic Memory Management	Enabled
OPEN_CURSORS	10000
SESSION_CACHED_CURSORS	50–150
DEFERRED_SEGMENT_CREATION	False
RESOURCE_LIMIT	True
ALTER PROFILE "DEFAULT" LIMIT IDLE_TIME 60 PASSWORD_LIFE_TIME UNLIMITED PASSWORD_GRACE_TIME UNLIMITED;	Set Default profile for ArcSDE and data owner users.
RECYCLEBIN	Off

Step 5: Configure ArcSDE Parameters

You will need to configure the MAXBLOBSIZE and TCPKEEPALIVE parameters for the ArcSDE geodatabase used as the Reviewer workspace. The MAXBLOBSIZE value is -1 by default. However, if you are using Oracle or another enterprise DBMS, make sure that this value is set to -1 and the TCPKEEPALIVE value is set to 1. This command should be used from the command prompt of a machine where ArcSDE is installed.

```
sdeconfig -o alter -v MAXBLOBSIZE=-1 -i <service> -u sde -p <sde_password>  
sdeconfig -o alter -v TCPKEEPALIVE=1 -i <service> -u sde -p <sde_password>
```

For more information, see the ArcSDE Administration Command Reference.

Step 6: Create the Reviewer Workspace

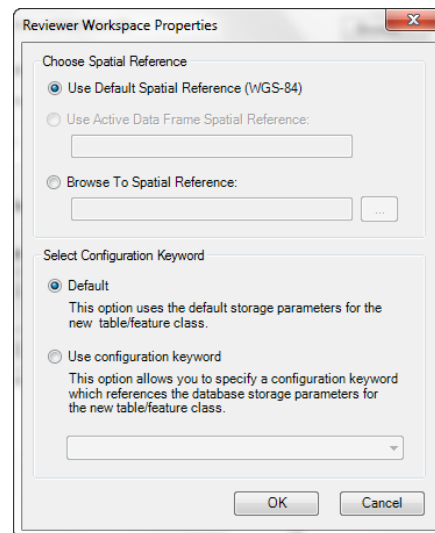
Create a database connection in ArcCatalog™ with the REV user; this will be the Reviewer workspace location.

ArcGIS Data Reviewer

In ArcMap™, click the Reviewer Session Manager button on the 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 Reviewer Workspace Properties dialog box appears.



Reviewer Workspace Properties

- Choose the custom spatial reference.

When you click Browse To Spatial Reference, the New Spatial Reference wizard appears. Choose the spatial reference you want to use with the Reviewer feature dataset.

- Choose the configuration keyword.

You can use the default configuration keyword or select the ##REVIEWER configuration keyword you created in step 3.

- 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.

Step 7: 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)
- RevBatchRunTable table
- RevCheckRunTable table
- RevTableConfig table
- RevTableLocation table
- RevTableMain table

Step 8: Verify Storage

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

```
sqlplus rev/rev@revdb
--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, use ALTER TABLE and ALTER INDEX 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), see Oracle MetaLink Doc ID: 130814.1, "How to Move LOB Data to Another Tablespace."

Step 9: Grant 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 "RLREVEDITOR";
CREATE ROLE "RLREVEDITOR" NOT IDENTIFIED;
DROP ROLE "RLREVVIEWER";
CREATE ROLE "RLREVVIEWER" NOT IDENTIFIED;
select 'grant select on ' ||owner|| '.' || table_name || ' to RLREVVIEWER;'
from sys.dba_tables where lower(owner) = 'rev' order by table_name;
select 'grant select,insert,update,delete on ' ||owner|| '.' || table_name || ' to RLREVEDITOR;'
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 REVVIEWER role to ArcSDE viewer users.

Tips for Granting Permissions

Grant a user only select, update, insert, and delete privileges to the RevAdminCustomFields and RevAdminDescriptions tables if the user has privileges to modify the database schema and add customized error descriptions.

Step 10: Set Log File Table Options

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 have more than 100 records 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
DATAFILE 'D:\oracle\ORADATA\REVDB\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
DATAFILE 'D:\oracle\ORADATA\REVDB\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.

```
sdedbtune -o export -f dbtune_logfile.sde -u sde -p sde -i sde:oracle11g:revdb
```

- 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.

```
sdedbtune -o export -f dbtune_logfile.sde -u sde -p sde -i sde:oracle11g:mcs
```

Create Log File Tables

- Grant QUOTA on SDELOGFILE and SDELOGFILEIDX permissions to the ArcSDE 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 ArcSDE editor/viewer user.
- In ArcMap, select more than 100 features; this automatically creates the log file tables.
- Remove CREATE TABLE permissions as appropriate.

Learn more about ArcSDE log file tables at resources.arcgis.com/en/help/main/10.1/index.html#//002n000001400000.

Step 11: Create ArcSDE Users

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

Editor User

```
CREATE USER REV_EDITOR PROFILE "DEFAULT"
  IDENTIFIED BY editor
  DEFAULT TABLESPACE "USERS"
  TEMPORARY TABLESPACE "TEMP" ACCOUNT UNLOCK;
GRANT "CONNECT" TO "REV_EDITOR";
GRANT CREATE TABLE TO "REV_EDITOR";
GRANT "RLREVEDITOR" TO "REV_EDITOR";
/*-- RLREVEDITOR role has SELECT, INSERT, UPDATE and DELETE permission on the REV data
(featureclasses, tables, etc.)
ALTER USER REV_EDITOR QUOTA UNLIMITED ON "SDELOGFILE";
ALTER USER REV_EDITOR QUOTA UNLIMITED ON "SDELOGFILEIDX";
```

Viewer User

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

Replication

You can use geodatabase replication to replicate the Reviewer workspace (one-way, two-way replication), but you can create new Reviewer sessions only in the parent geodatabase or only in the child geodatabase.

Conclusion

You can reduce disk contention and improve database I/O by storing the ArcGIS Data Reviewer workspace in different locations on disk. However, this practice alone does not guarantee optimal database performance, and additional tuning tasks may be needed.

Learn more about the recommended tuning tasks:
resources.arcgis.com/en/help/main/10.1/index.html#/Minimize_disk_I_O_contention_in_Oracle/002n0000002500000/

Learn more about ArcGIS Data Reviewer:
visit esri.com/software/arcgis/extensions/arcgis-data-reviewer/index.html or e-mail datareviewer@esri.com

Access blogs, forums, downloads, and more from the ArcGIS Resource Center:
resources.arcgis.com/en/communities/data-reviewer/

You can access other resources at ArcGIS 10.1 for Desktop Help:
resources.arcgis.com/en/help/main/10.1/#/Welcome_to_the_ArcGIS_Professional_Help_Library/00qn0000001p000000/ and Esri Support (support.esri.com)



Esri inspires and enables people to positively impact their future through a deeper, geographic understanding of the changing world around them.

Governments, industry leaders, academics, and nongovernmental organizations trust us to connect them with the analytic knowledge they need to make the critical decisions that shape the planet. For more than 40 years, Esri has cultivated collaborative relationships with partners who share our commitment to solving earth's most pressing challenges with geographic expertise and rational resolve. Today, we believe that geography is at the heart of a more resilient and sustainable future. Creating responsible products and solutions drives our passion for improving quality of life everywhere.



Contact Esri

380 New York Street
Redlands, California 92373-8100 USA

1 800 447 9778
T 909 793 2853
F 909 793 5953
info@esri.com
esri.com

Offices worldwide
esri.com/locations