Best Practices for Storing the Esri® Production Mapping Workspace in an Enterprise Geodatabase for Oracle

An Esri® Technical Paper
September 2013

380 New York Street
Redlands, California 92373-8100 USA

800 447 9778
info@esri.com

esri.com

Best Practices for Storing the Esri Production Mapping Workspace in an Enterprise Geodatabase for Oracle

An Esri Technical Paper

Contents **Page**

J10066

Contents **Page**

Best Practices for Storing the Esri Production Mapping Workspace in an Enterprise Geodatabase for Oracle

Introduction

Esri® Production Mapping, an extension to ArcGIS® 10.1 for Desktop, streamlines your geographic information system (GIS) data and map production. It provides tools that facilitate data creation, maintenance, and validation as well as tools for producing high-quality cartographic products. Each organization has workflows that are unique to the type of data it collects and the products it delivers. These workflows can be generalized into a basic production workflow consisting of the steps used to create the geodatabase, capture or load the initial data, perform edits to the data, ensure data validity and accuracy, and produce digital or hard-copy output. Production Mapping can streamline each step while remaining flexible enough to adapt to your organization's business rules and workflows. This technical paper offers best practices for database administrators when establishing the Production Mapping workspace in an enterprise geodatabase for Oracle. The enterprise geodatabase uses ArcSDE® technology as the gateway between GIS clients and Oracle.

The Geodatabase

The geodatabase is central to any production workflow. Your geodatabase design is determined by the data that will be captured and edited, the logic rules used for validating data, and the types of output to be created.

When designing a geodatabase, certain types of validation are built in, such as the ability to limit the valid attribute values for a field by using a domain. You can also model the geometric relationships of features through topologies or networks. Production Mapping supports these forms of validation and allows you to define additional business rules using ArcGIS Data Reviewer checks stored in a batch job. The batch jobs can be run when you update the attributes of a feature or template, like domains, or as a postprocess such as validating the topology.

The product library in Production Mapping provides a framework for managing business rules, data, and map documents in a secure, centralized location. By leveraging the rules stored in the product library, data editing tools are tailored to ensure that features conform to your data collection standards. During cartographic production, the product library can act as a document management system for your map documents and data, allowing you to check in and check out files and restore historical versions. The cartographic tools in Production Mapping can also be used to ensure consistent and repeatable symbology as well as provide a number of dynamic surround elements.

See the white paper *Best Practices for Storing the Product Library Workspace in an Enterprise Geodatabase for Oracle*.

Production Database

A production database contains the data you are using for production tasks such as creating and updating features. Depending on the data model you are using, data in a production database can be used to create a digital or hard-copy map/chart or a specific type of data. The data in a production database usually corresponds with a data model and product class in the product library. This technical paper will help you establish the production database in an enterprise geodatabase for Oracle.

Capture

The purpose of data capture is to consume existing data or create new data in your geodatabase. Data can be captured in many different ways, such as extracting new data from imagery, gathering information in the field with a mobile device, or converting existing data.

Production Mapping provides data loading automation tools that allow you to define the mapping ahead of time between source data, such as shapefiles or coverages, and your geodatabase to ensure consistency when converting large amounts of data. Feature Manager allows you to quickly and easily create new features by using feature templates and construction tools within a centralized editing environment that leverages your enhanced validation rules.

Edit

The editing stage of the production workflow involves adjusting existing features and adding new features to ensure that your data is up-to-date and accurate. This can involve updating data to match a new source or modifying attributes of data that was collected for a different purpose.

When editing data within Production Mapping, feature attribution is managed through Feature Manager, which allows you to update fields while ensuring the attributes are valid in accordance with your validation rules as well as update attributes in batches and create new features. Production Mapping also provides a number of editing tools for batch geometry updates, as well as tools for specific types of data or industries, including tools for linear referencing, utilities, contour lines, and z-enabled data.

Review

Data review is important to ensure that the data being created is accurate and fit for its intended purpose before it's delivered or used for making a map product. The data review or quality control stage of the production workflow often involves three phases: finding issues with the data, either through manual or automated methods; fixing issues or marking them as exceptions; and verifying that issues are resolved.

The Data Reviewer component of Production Mapping provides the ability to track records of all issues found in the Reviewer table, which maintains the status of the record as well as a link between the record and the feature with the issue. Data Reviewer offers the ability to automate data validation through configurable checks that can be scheduled through a Windows service to run at specified times, or run through Python scripts. Automated checks may not be able to find all data problems. Data Reviewer also has tools to help with manual or visual quality control, including tools to flag missing features, and the ability to create a sample set of data for in-depth validation.

Note: Data Reviewer is available with Production Mapping or as a stand-alone extension.

See the white paper *Best Practices for Storing the ArcGIS Data Reviewer Workspace in an Enterprise Geodatabase for Oracle*.

Create Output

Typically, the final stage of a production workflow is the creation of the output that will be delivered. This stage may be repeated if delivery is contingent on approval or if the product requires regular updates. Many types of output may be produced: hard-copy maps or data exported to a certain format or served over the web.

Many of the hard-copy products produced—from one-off maps to map books to highly detailed charts—require version tracking. Production Mapping provides check-in and checkout capabilities for map documents as well as the ability to track history and roll back to a previous version.

Data visualization and symbology are important when producing a hard-copy product or serving data over the web. Production Mapping provides the tools for consistent, repeatable, rule-based symbology, where you define what symbol or representation should be applied to features based on their attribute combination. Production Mapping also provides a number of custom surround elements, such as the graphic table element, which allows you to create a table or legend that automatically updates based on the data being displayed.

Workflow Management

When managing production, it is important to be able to allocate resources and track the status of the project. Being able to ensure that work is being done consistently and steps are not being skipped is also essential. Production Mapping allows you to tie all the components of data capture, editing, validation, and cartography together in high-level workflows with ArcGIS Workflow Manager and in detailed workflows with Task Assistant Manager.

Note: ArcGIS Workflow Manager is available with Production Mapping and as a stand-alone extension.

Workflow Manager allows you to create a job (unit of work), assign or reassign the job to a user, and track the overall status of the project. Each job includes a predefined workflow that you build to represent your processes. When assigned a job, you execute the steps in the workflow to launch the appropriate tools, send e-mail notifications, or ask questions to determine the path the job should take.

Task Assistant Manager allows you to define workflows in ArcMap™ that guide you through various tasks. Task Assistant workflows can be used to provide step-by-step instructions for complex tasks to minimize confusion for new users, or simply as a reference. Clicking a task in a workflow can execute either a tool in ArcMap or geoprocessing tools, set up your environment by specifying layer or snapping properties, or provide a description of what needs to be done.

See the white paper *Best Practices for Storing the ArcGIS Workflow Manager Workspace in an Enterprise Geodatabase for Oracle*.

Production Mapping Solutions

Production Mapping can be used to build your own solutions by creating a product library to meet the standards of an industry or business. However, Esri has built three commercial off-the-shelf solutions for the defense mapping, maritime, and aviation industries that utilize and expand on Production Mapping functionality.

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. The DBTUNE table, along with all other metadata tables, is created in the database when the Create Enterprise Geodatabase or Enable Enterprise Geodatabase tool is executed.

When a large number of database connections access the same files in the same location on disk, database performance is 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 Production feature dataset and tables in separate data files across different locations on disk. This will lead to reduced disk contention and improved 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 to 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 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 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.

Suggested Oracle optimal configuration is as follows:

- Disk 0—Oracle/App Software
- 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 that are 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 minimize 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

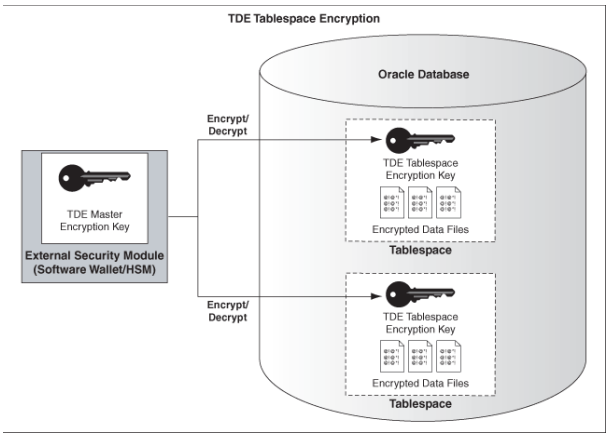
Transparent data encryption (TDE) enables you to encrypt sensitive data, such as credit card numbers, stored in tables and tablespaces. Encrypted data is transparently decrypted for a database user or application that has access to data. TDE helps protect data stored on media in the event that the storage media or data file is 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 transparent data encryption. TDE encrypts sensitive data stored in data files. To prevent unauthorized decryption, TDE stores the encryption keys in a security module outside the database.

Benefits of using TDE include the following:

- As a security administrator, you can be sure that sensitive data is safe in case the storage media or data file is 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 because 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.



Tablespace TDE

See the Oracle documentation on how to configure tablespace TDE:

http://docs.oracle.com/cd/E11882_01/network.112/e10746/asotrans.htm

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

--Create the wallet folder

mkdir D:\oracle\admin\wallets

OEM > login as sys / sysdba

OEM > Server > Transparent Data Encryption

Advanced Options > Change Location

     Host Credentials
     Username: avworld\dbs_ora
     Password: xxxxxxx

     Configuration Method: File System

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

     OK

Create Wallet > Local Auto-Open Wallet > Create

     Host Credentials
     Username: avworld\dbs_ora
     Password: xxxxxxx

     Wallet Password:  walletadmin

     Continue

--Backup the wallet folder

cd D:\oracle\admin

zip -r wallets wallets
```

J10066

```
/*-----------------------------------------------------------------------
--Configure TDE Manually
-----------------------------------------------------------------------*/
--Create the wallet folder

mkdir D:\oracle\admin\wallets

--Add wallet location to sqlnet.ora

ENCRYPTION_WALLET_LOCATION =
  (SOURCE =
    (METHOD = FILE)
    (METHOD_DATA =
      (DIRECTORY = D:\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=MCS

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

--Backup the wallet folder

cd D:\oracle\admin

zip -r wallets wallets
```

Using Data Compression

Table Compression: Overview

The Oracle database was the pioneer in terms of compression technology for databases with the introduction of table compression for bulk load operations in Oracle9*i*. By using this feature, you could compress data when performing bulk load using operations such as direct loads or Create Table As Select (CTAS). However, until now, compression was not available for regular data manipulation operations such as INSERT, UPDATE, and DELETE. Oracle Database 11*g* extends the compression technology to support these operations as well. Consequently, compression in Oracle Database 11*g* can be used for all kinds of workloads—online transaction processing (OLTP) or data warehousing. It is important to mention that table compression enhancements introduced in Oracle database 11*g* are not just incremental changes. An enormous amount of work has gone into making sure that the new compression technology has negligible impact on updates because any noticeable write time penalty due to compression will not be acceptable in an OLTP environment. As a result, compression technology in Oracle Database 11*g* is very efficient and could reduce the space consumption by 50–75 percent. So, your write performance does not degrade, and your read performance or queries improve. This is because, unlike desktop-based compression techniques where you have to wait for data to be uncompressed, Oracle technology reads the compressed data (fewer fetches needed) directly and does not require any uncompress operation.

Note: Compression technology is completely application transparent. This means that you can use this technology with any application such as ArcGIS.

Using Table Compression

- Requires database compatibility level at 11.1 or greater
- Use the COMPRESS keyword:
  - COMPRESS [FOR {ALL | DIRECT_LOAD} OPERATIONS]
  - FOR DIRECT_LOAD is the default: Refers to bulk load operations from prior releases
  - FOR ALL OPERATIONS: OLTP + direct loads
- Enable compression for new tables:
  - CREATE TABLE t1 COMPRESS FOR ALL OPERATIONS;
- Enable compression on existing table:
  - ALTER TABLE t2 COMPRESS FOR ALL OPERATIONS;
  - Does not trigger compression on existing rows

To use the new compression algorithm, you must flag your table with the COMPRESS FOR ALL OPERATIONS clause. You can do so at table creation or after creation. This is illustrated in the examples given in the slide.

If you use the COMPRESS clause without specifying any FOR option or if you use the COMPRESS FOR DIRECT_LOAD OPERATIONS clause, you will fall back to the old compression mechanism that was available in earlier releases.

You can also enable compression at the partition or tablespace level. For example, you can use the DEFAULT storage clause of the CREATE TABLESPACE command to optionally specify a COMPRESS FOR clause.

Note: You can view compression flags for your tables using the COMPRESS and COMPRESS_FOR columns in views such as DBA_TABLES and DBA_TAB_PARTITIONS.

Compression and TDE

Customers using TDE tablespace encryption get the full benefit of compression (Standard and Advanced Compression, as well as Exadata Hybrid Columnar Compression [EHCC]), because compression is applied before the data blocks are encrypted. Customers using TDE column encryption will get the full benefit of compression only on table columns that are not encrypted. Individual table columns that are encrypted using TDE column encryption will have a much lower level of compression because the encryption takes place in the SQL layer before the Advanced Compression process.

Oracle 11*g*: Transparent Data Encryption (TDE) Frequently Asked Questions

Step 1: Create Data Files

Create new tablespaces to store the production data.

```
TABLESPACE        DATAFILE_NAME     REMARKS
PM_BDATA          PM_Bdata          Business table
PM_BINDEX         PM_Bindex         Business table index
PM_FDATA          PM_Fdata          ST_Geometry Lob storage
PM_FINDEX         PM_Findex         ST_Geometry Lob index
PM_SDATA          PM_Sdata          Spatial Index Tables
PM_SINDEX         PM_Sindex         Spatial Index Features
PM_ADATA          PM_Adata          Adds table (versioned)
PM_AINDEX         PM_Aindex         Adds table index
PM_DDATA          PM_Ddata          Delete table (versioned)
PM_DINDEX         PM_Dindex         Delete table index
PM_RASTER         PM_Raster         Raster Datasets
PM_RINDEX         PM_Rindex         Raster Indexes
PM_RBLK           PM_Rblk           Raster BLK table
PM_RBLKIDX        PM_Rblkidx        Raster BLK table Indexes
PM_XMLDOC         PM_Xmldoc         XML Documents
PM_XMLIDX         PM_Xmlidx         XML Documents Indexes
PM_TOPO_BDATA     PM_Topo_Bdata     Topology Business table
PM_TOPO_BINDEX    PM_Topo_Bindex    Topology Business table index
PM_TOPO_FDATA     PM_Topo_Fdata     Topology ST_Geometry Lob storage
PM_TOPO_FINDEX    PM_Topo_Findex    Topology ST_Geometry Lob index
PM_TOPO_SDATA     PM_Topo_Sdata     Topology Spatial Index Tables
PM_TOPO_SINDEX    PM_Topo_Sindex    Topology Spatial Index Features
PM_TERRAIN_BDATA  PM_Terrain_Bdata  Terrain Business table
PM_TERRAIN_BINDEX PM_Terrain_Bindex Terrain Business table index
PM_TERRAIN_FDATA  PM_Terrain_Fdata  Terrain ST_Geometry Lob storage
PM_TERRAIN_FINDEX PM_Terrain_Findex Terrain ST_Geometry Lob index
PM_TERRAIN_SDATA  PM_Terrain_Sdata  Terrain Spatial Index Tables
PM_TERRAIN_SINDEX PM_Terrain_Sindex Terrain Spatial Index Features
PM_NET_BDATA      PM_Net_Bdata      Network Business table
PM_NET_BINDEX     PM_Net_Bindex     Network Business table index
PM_NET_FDATA      PM_Net_Fdata      Network ST_Geometry Lob storage
PM_NET_FINDEX     PM_Net_Findex     Network ST_Geometry Lob index
PM_NET_SDATA      PM_Net_Sdata      Network Spatial Index Tables
PM_NET_SINDEX     PM_Net_Sindex     Network Spatial Index Features

CREATE SMALLFILE TABLESPACE "PM_BDATA"
DATAFILE 'D:\oradata\MCS\PM\PM_BDATA01.dbf'SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';

CREATE SMALLFILE TABLESPACE "PM_BINDEX"
DATAFILE 'D:\oradata\MCS\PM\PM_BINDEX01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';

CREATE SMALLFILE TABLESPACE "PM_FDATA"
DATAFILE 'D:\oradata\MCS\PM\PM_FDATA01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';

CREATE SMALLFILE TABLESPACE "PM_FINDEX"
DATAFILE 'D:\oradata\MCS\PM\PM_FINDEX01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';

CREATE SMALLFILE TABLESPACE "PM_SDATA"
DATAFILE 'D:\oradata\MCS\PM\PM_SDATA01.dbf'SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';

CREATE SMALLFILE TABLESPACE "PM_SINDEX"
DATAFILE 'D:\oradata\MCS\PM\PM_SINDEX01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';

CREATE SMALLFILE TABLESPACE "PM_ADATA"
DATAFILE 'D:\oradata\MCS\PM\PM_ADATA01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';

CREATE SMALLFILE TABLESPACE "PM_AINDEX"
DATAFILE 'D:\oradata\MCS\PM\PM_AINDEX01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';
```

```
CREATE SMALLFILE TABLESPACE "PM_DDATA"
DATAFILE 'D:\oradata\MCS\PM\PM_DDATA01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';

CREATE SMALLFILE TABLESPACE "PM_DINDEX"
DATAFILE 'D:\oradata\MCS\PM\PM_DINDEX01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';

CREATE SMALLFILE TABLESPACE "PM_XMLDOC"
DATAFILE 'D:\oradata\MCS\PM\PM_XMLDOC01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';

CREATE SMALLFILE TABLESPACE "PM_XMLIDX"
DATAFILE 'D:\oradata\MCS\PM\PM_XMLIDX01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';

CREATE SMALLFILE TABLESPACE "PM_TOPO_BDATA"
DATAFILE 'D:\oradata\MCS\PM\PM_TOPO_BDATA01.dbf'SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';

CREATE SMALLFILE TABLESPACE "PM_TOPO_BINDEX"
DATAFILE 'D:\oradata\MCS\PM\PM_TOPO_BINDEX01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';

CREATE SMALLFILE TABLESPACE "PM_TOPO_FDATA"
DATAFILE 'D:\oradata\MCS\PM\PM_TOPO_FDATA01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';

CREATE SMALLFILE TABLESPACE "PM_TOPO_FINDEX"
DATAFILE 'D:\oradata\MCS\PM\PM_TOPO_FINDEX01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';

CREATE SMALLFILE TABLESPACE "PM_TOPO_SDATA"
DATAFILE 'D:\oradata\MCS\PM\PM_TOPO_SDATA01.dbf'SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';

CREATE SMALLFILE TABLESPACE "PM_TOPO_SINDEX"
DATAFILE 'D:\oradata\MCS\PM\PM_TOPO_SINDEX01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';

CREATE SMALLFILE TABLESPACE "PM_TERRAIN_BDATA"
DATAFILE 'D:\oradata\MCS\PM\PM_TERRAIN_BDATA01.dbf'SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';

CREATE SMALLFILE TABLESPACE "PM_TERRAIN_BINDEX"
DATAFILE 'D:\oradata\MCS\PM\PM_TERRAIN_BINDEX01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';

CREATE SMALLFILE TABLESPACE "PM_TERRAIN_FDATA"
DATAFILE 'D:\oradata\MCS\PM\PM_TERRAIN_FDATA01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';

CREATE SMALLFILE TABLESPACE "PM_TERRAIN_FINDEX"
DATAFILE 'D:\oradata\MCS\PM\PM_TERRAIN_FINDEX01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';
```

J10066

```
CREATE SMALLFILE TABLESPACE "PM_TERRAIN_SDATA"
DATAFILE 'D:\oradata\MCS\PM\PM_TERRAIN_SDATA01.dbf'SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';

CREATE SMALLFILE TABLESPACE "PM_TERRAIN_SINDEX"
DATAFILE 'D:\oradata\MCS\PM\PM_TERRAIN_SINDEX01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';

CREATE SMALLFILE TABLESPACE "PM_NET_BDATA"
DATAFILE 'D:\oradata\MCS\PM\PM_NET_BDATA01.dbf'SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';

CREATE SMALLFILE TABLESPACE "PM_NET_BINDEX"
DATAFILE 'D:\oradata\MCS\PM\PM_NET_BINDEX01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';

CREATE SMALLFILE TABLESPACE "PM_NET_FDATA"
DATAFILE 'D:\oradata\MCS\PM\PM_NET_FDATA01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';

CREATE SMALLFILE TABLESPACE "PM_NET_FINDEX"
DATAFILE 'D:\oradata\MCS\PM\PM_NET_FINDEX01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';

CREATE SMALLFILE TABLESPACE "PM_NET_SDATA"
DATAFILE 'D:\oradata\MCS\PM\PM_NET_SDATA01.dbf'SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';

CREATE SMALLFILE TABLESPACE "PM_NET_SINDEX"
DATAFILE 'D:\oradata\MCS\PM\PM_NET_SINDEX01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';

CREATE SMALLFILE TABLESPACE "PM_RASTER"
DATAFILE 'D:\oradata\MCS\PM\PM_RASTER01.dbf' SIZE 10M AUTOEXTEND ON NEXT 4M MAXSIZE 800M
BLOCKSIZE 8k
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';

CREATE SMALLFILE TABLESPACE "PM_RINDEX"
DATAFILE 'D:\oradata\MCS\PM\PM_RINDEX01.dbf' SIZE 10M AUTOEXTEND ON NEXT 4M MAXSIZE 800M
BLOCKSIZE 8K
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';

CREATE SMALLFILE TABLESPACE "PM_RBLK"
DATAFILE 'D:\oradata\MCS\PM\PM_RBLK01.dbf' SIZE 10M AUTOEXTEND ON NEXT 4M MAXSIZE 8G
BLOCKSIZE 8k
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';

CREATE SMALLFILE TABLESPACE "PM_RBLKIDX"
DATAFILE 'D:\oradata\MCS\PM\PM_RBLKIDX01.dbf' SIZE 10M AUTOEXTEND ON NEXT 4M MAXSIZE 800M
BLOCKSIZE 8K
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';
```

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:\ORADATA\MCS\pm\pm_Bdata01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\ORADATA\MCS\pm\pm_Bindex01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\ORADATA\MCS\pm\pm_Fdata01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\ORADATA\MCS\pm\pm_Findex01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\ORADATA\MCS\pm\pm_Sdata01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\ORADATA\MCS\pm\pm_Sindex01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\ORADATA\MCS\pm\pm_Adata01.dbf' RESIZE 400M;
```

```
ALTER DATABASE DATAFILE 'D:\ORADATA\MCS\pm\pm_Aindex01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\ORADATA\MCS\pm\pm_Ddata01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\ORADATA\MCS\pm\pm_Dindex01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\ORADATA\MCS\pm\pm_Xmldoc01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\ORADATA\MCS\pm\pm_Xmlidx01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\ORADATA\MCS\pm\pm_Raster01.dbf' RESIZE 800M;
ALTER DATABASE DATAFILE 'D:\ORADATA\MCS\pm\pm_Rindex01.dbf' RESIZE 800M;
ALTER DATABASE DATAFILE 'D:\ORADATA\MCS\pm\pm_Rblk01.dbf' RESIZE 2G;
ALTER DATABASE DATAFILE 'D:\ORADATA\MCS\pm\pm_Rblkidx01.dbf' RESIZE 800M;
ALTER DATABASE DATAFILE 'D:\ORADATA\MCS\pm\pm_Topo_Bdata01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\ORADATA\MCS\pm\pm_Topo_Bindex01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\ORADATA\MCS\pm\pm_Topo_Fdata01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\ORADATA\MCS\pm\pm_Topo_Findex01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\ORADATA\MCS\pm\pm_Topo_Sdata01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\ORADATA\MCS\pm\pm_Topo_Sindex01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\ORADATA\MCS\pm\pm_Terrain_Bdata01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\ORADATA\MCS\pm\pm_Terrain_Bindex01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\ORADATA\MCS\pm\pm_Terrain_Fdata01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\ORADATA\MCS\pm\pm_Terrain_Findex01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\ORADATA\MCS\pm\pm_Terrain_Sdata01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\ORADATA\MCS\pm\pm_Terrain_Sindex01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\ORADATA\MCS\pm\pm_Net_Bdata01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\ORADATA\MCS\pm\pm_Net_Bindex01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\ORADATA\MCS\pm\pm_Net_Fdata01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\ORADATA\MCS\pm\pm_Net_Findex01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\ORADATA\MCS\pm\pm_Net_Sdata01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\ORADATA\MCS\pm\pm_Net_Sindex01.dbf' RESIZE 400M;
```

Step 2: Create PM User

Create a new database user to store the production data and grant the appropriate permissions.

```
CREATE USER PM PROFILE DEFAULT IDENTIFIED BY pmadmin
     DEFAULT TABLESPACE "PM_BDATA"
     TEMPORARY TABLESPACE "TEMP"
  QUOTA UNLIMITED ON "PM_BDATA"
  QUOTA UNLIMITED ON "PM_BINDEX"
  QUOTA UNLIMITED ON "PM_FDATA"
  QUOTA UNLIMITED ON "PM_FINDEX"
  QUOTA UNLIMITED ON "PM_SDATA"
  QUOTA UNLIMITED ON "PM_SINDEX"
  QUOTA UNLIMITED ON "PM_ADATA"
  QUOTA UNLIMITED ON "PM_AINDEX"
  QUOTA UNLIMITED ON "PM_DDATA"
  QUOTA UNLIMITED ON "PM_DINDEX"
  QUOTA UNLIMITED ON "PM_XMLDOC"
  QUOTA UNLIMITED ON "PM_XMLIDX"
  QUOTA UNLIMITED ON "PM_RASTER"
  QUOTA UNLIMITED ON "PM_RINDEX"
  QUOTA UNLIMITED ON "PM_RBLK"
  QUOTA UNLIMITED ON "PM_RBLKIDX"
  QUOTA UNLIMITED ON "PM_TOPO_BDATA"
  QUOTA UNLIMITED ON "PM_TOPO_BINDEX"
  QUOTA UNLIMITED ON "PM_TOPO_FDATA"
  QUOTA UNLIMITED ON "PM_TOPO_FINDEX"
  QUOTA UNLIMITED ON "PM_TOPO_SDATA"
  QUOTA UNLIMITED ON "PM_TOPO_SINDEX"
  QUOTA UNLIMITED ON "PM_TERRAIN_BDATA"
  QUOTA UNLIMITED ON "PM_TERRAIN_BINDEX"
  QUOTA UNLIMITED ON "PM_TERRAIN_FDATA"
  QUOTA UNLIMITED ON "PM_TERRAIN_FINDEX"
  QUOTA UNLIMITED ON "PM_TERRAIN_SDATA"
  QUOTA UNLIMITED ON "PM_TERRAIN_SINDEX"
  QUOTA UNLIMITED ON "PM_NET_BDATA"
  QUOTA UNLIMITED ON "PM_NET_BINDEX"
  QUOTA UNLIMITED ON "PM_NET_FDATA"
  QUOTA UNLIMITED ON "PM_NET_FINDEX"
  QUOTA UNLIMITED ON "PM_NET_SDATA"
  QUOTA UNLIMITED ON "PM_NET_SINDEX"
  ACCOUNT UNLOCK;

CREATE ROLE "GIS_DATA_OWNER"  NOT IDENTIFIED;
GRANT CREATE SESSION TO "GIS_DATA_OWNER";
GRANT CREATE SEQUENCE TO "GIS_DATA_OWNER";
GRANT CREATE TRIGGER TO "GIS_DATA_OWNER";
GRANT CREATE VIEW TO "GIS_DATA_OWNER";
GRANT CREATE TABLE TO "GIS_DATA_OWNER";
GRANT GIS_DATA_OWNER TO PM;
```

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:mcs
```

- Copy ***dbtune_exp.sde*** to ***dbtune_pm.sde***.

- Fill the new dbtune file with the proper tablespaces:

```
dbtune_pm.sde

##DATA_DICTIONARY
ATTRIBUTE_BINARY        "BLOB"
B_STORAGE               "PCTFREE 0 INITRANS 4 TABLESPACE PM_SDEDICT STORAGE (INITIAL 40K)"
B_INDEX_ROWID           "PCTFREE 0 INITRANS 4 TABLESPACE PM_SDEDICTINDEX STORAGE (INITIAL 40K)
NOLOGGING"
B_INDEX_USER            "PCTFREE 0 INITRANS 4 TABLESPACE PM_SDEDICTINDEX STORAGE (INITIAL 40K)
NOLOGGING"
STATES_TABLE            "INITRANS 4 TABLESPACE PM_SDEDICT STORAGE (INITIAL 1M)"
STATES_INDEX            "INITRANS 5 TABLESPACE PM_SDEDICTINDEX STORAGE (INITIAL 128K)
NOLOGGING"
STATE_LINEAGES_TABLE    "PCTFREE 0 INITRANS 4 TABLESPACE PM_SDEDICT STORAGE (INITIAL 7M)"
STATE_LINEAGES_INDEX    "PCTFREE 0 INITRANS 4 TABLESPACE PM_SDEDICTINDEX STORAGE (INITIAL 5M)
NOLOGGING"
VERSIONS_TABLE          "INITRANS 4 TABLESPACE PM_SDEDICT STORAGE (INITIAL 256K)"
VERSIONS_INDEX          "INITRANS 4 TABLESPACE PM_SDEDICTINDEX STORAGE (INITIAL 128K)
NOLOGGING"
MVTABLES_MODIFIED_TABLE "INITRANS 4 TABLESPACE PM_SDEDICT STORAGE (INITIAL 2M)"
MVTABLES_MODIFIED_INDEX "INITRANS 4 TABLESPACE PM_SDEDICTINDEX STORAGE (INITIAL  2M)
NOLOGGING"
XML_INDEX_TAGS_TABLE    "INITRANS 4 TABLESPACE PM_SDEDICT STORAGE (INITIAL 1M)"
XML_INDEX_TAGS_INDEX    "INITRANS 5 TABLESPACE PM_SDEDICTINDEX STORAGE (INITIAL 1M)"
END

##DEFAULTS
GEOMETRY_STORAGE    "ST_GEOMETRY"
ST_GEOM_LOB_STORAGE  " STORE AS (TABLESPACE PM_FDATA ENABLE STORAGE IN ROW CHUNK 8K RETENTION
CACHE INDEX (TABLESPACE PM_FINDEX)) "
ATTRIBUTE_BINARY    "BLOB"
RASTER_STORAGE      "BLOB"
B_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE PM_BDATA"
B_INDEX_ROWID       "PCTFREE 0 INITRANS 4 TABLESPACE PM_BINDEX NOLOGGING"
B_INDEX_USER        "PCTFREE 0 INITRANS 4 TABLESPACE PM_BINDEX NOLOGGING"
B_INDEX_XML         "PCTFREE 0 INITRANS 4 TABLESPACE PM_BINDEX NOLOGGING"
B_INDEX_RASTER      "PCTFREE 0 INITRANS 4 TABLESPACE PM_BINDEX NOLOGGING"
B_INDEX_TO_DATE     "PCTFREE 0 INITRANS 4 TABLESPACE PM_BINDEX NOLOGGING"
S_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE PM_SDATA"
S_INDEX_ALL         "PCTFREE 0 INITRANS 4 TABLESPACE PM_SINDEX NOLOGGING"
A_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE PM_ADATA"
A_INDEX_ROWID       "PCTFREE 0 INITRANS 4 TABLESPACE PM_AINDEX NOLOGGING"
A_INDEX_SHAPE       "PCTFREE 0 INITRANS 4 TABLESPACE PM_AINDEX NOLOGGING"
A_INDEX_STATEID     "PCTFREE 0 INITRANS 4 TABLESPACE PM_AINDEX NOLOGGING"
A_INDEX_USER        "PCTFREE 0 INITRANS 4 TABLESPACE PM_AINDEX NOLOGGING"
A_INDEX_XML         "PCTFREE 0 INITRANS 4 TABLESPACE PM_AINDEX NOLOGGING"
A_INDEX_RASTER      "PCTFREE 0 INITRANS 4 TABLESPACE PM_AINDEX NOLOGGING"
D_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE PM_DDATA"
D_INDEX_STATE_ROWID "PCTFREE 0 INITRANS 4 TABLESPACE PM_DINDEX NOLOGGING"
D_INDEX_DELETED_AT  "PCTFREE 0 INITRANS 4 TABLESPACE PM_DINDEX NOLOGGING"
RAS_STORAGE         "PCTFREE 0 INITRANS 4 TABLESPACE PM_RASTER"
RAS_INDEX_ID        "PCTFREE 0 INITRANS 4 TABLESPACE PM_RINDEX NOLOGGING"
BND_STORAGE         "PCTFREE 0 INITRANS 4 TABLESPACE PM_RASTER"
BND_INDEX_COMPOSITE "PCTFREE 0 INITRANS 4 TABLESPACE PM_RINDEX NOLOGGING"
BND_INDEX_ID        "PCTFREE 0 INITRANS 4 TABLESPACE PM_RINDEX NOLOGGING"
AUX_STORAGE         "PCTFREE 0 INITRANS 4 TABLESPACE PM_RASTER"
AUX_INDEX_COMPOSITE "PCTFREE 0 INITRANS 4 TABLESPACE PM_RINDEX NOLOGGING"
BLK_STORAGE         "PCTFREE 0 INITRANS 4 TABLESPACE PM_RBLK"
BLK_INDEX_COMPOSITE "PCTFREE 0 INITRANS 4 TABLESPACE PM_RBLKIDX NOLOGGING"
UI_TEXT             "User Interface text for DEFAULTS"
XML_DOC_UNCOMPRESSED_TYPE   "CLOB"
XML_IDX_TEXT_UPDATE_METHOD "NONE"
XML_DOC_MODE                "COMPRESSED"
XML_DOC_LOB_STORAGE         "NOCACHE NOLOGGING CHUNK 4K PCTVERSION 5 DISABLE STORAGE IN ROW
TABLESPACE PM_XMLDOC"
XML_DOC_VAL_LOB_STORAGE     "NOCACHE NOLOGGING CHUNK 4K PCTVERSION 5 DISABLE STORAGE IN ROW
TABLESPACE PM_XMLDOC"
XML_DOC_STORAGE     "PCTFREE 0 INITRANS 4 TABLESPACE PM_XMLDOC"
XML_DOC_INDEX       "PCTFREE 0 INITRANS 4 TABLESPACE PM_XMLIDX NOLOGGING"
XML_IDX_STORAGE     "PCTFREE 0 INITRANS 4 TABLESPACE PM_XMLIDX"
XML_IDX_TEXT_TAG_STORAGE   ""
XML_IDX_INDEX_PK    "PCTFREE 0 INITRANS 4 TABLESPACE PM_XMLIDX NOLOGGING"
XML_IDX_INDEX_ID    "PCTFREE 0 INITRANS 4 TABLESPACE PM_XMLIDX NOLOGGING"
XML_IDX_INDEX_TEXT  ""
```

```
XML_IDX_TEXT_UPDATE_MEMORY ""
XML_IDX_INDEX_TAG    "PCTFREE 0 INITRANS 4 TABLESPACE PM_XMLIDX NOLOGGING"
XML_IDX_INDEX_DOUBLE "PCTFREE 0 INITRANS 4 TABLESPACE PM_XMLIDX NOLOGGING"
XML_IDX_INDEX_STRING "PCTFREE 0 INITRANS 4 TABLESPACE PM_XMLIDX NOLOGGING"
END

##SDO_GEOMETRY
GEOMETRY_STORAGE    "SDO_GEOMETRY"
ATTRIBUTE_BINARY    "BLOB"
RASTER_STORAGE      "SDO_GEORASTER"
SDO_COMMIT_INTERVAL  1000
UI_TEXT             "User Interface text description for SDO_GEOMETRY"
COMMENT             "Any general comment for SDO_GEOMETRY keyword"
END

##SDO_GEORASTER
GEOMETRY_STORAGE    "SDO_GEOMETRY"
RASTER_STORAGE      "SDO_GEORASTER"
ATTRIBUTE_BINARY    "BLOB"
SDO_COMMIT_INTERVAL  1000
RDT_STORAGE         "PCTFREE 0 INITRANS 4 TABLESPACE PM_RASTER"
RDT_INDEX_COMPOSITE "PCTFREE 0 INITRANS 4 TABLESPACE PM_RINDEX STORAGE ( INITIAL 409600)
NOLOGGING"
UI_TEXT             "User Interface text description for SDO_GEORASTER"
COMMENT             "Any general comment for SDO_GEORASTER keyword"
END

##SDELOB
GEOMETRY_STORAGE    "SDELOB"
ATTRIBUTE_BINARY    "BLOB"
RASTER_STORAGE   "BLOB"
B_INDEX_SHAPE       "PCTFREE 0 INITRANS 4 TABLESPACE PM_BINDEX NOLOGGING"
F_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE PM_FDATA"
F_INDEX_FID         "PCTFREE 0 INITRANS 4 TABLESPACE PM_FINDEX NOLOGGING"
F_INDEX_AREA        "PCTFREE 0 INITRANS 4 TABLESPACE PM_FINDEX NOLOGGING"
F_INDEX_LEN         "PCTFREE 0 INITRANS 4 TABLESPACE PM_FINDEX NOLOGGING"
S_INDEX_SP_FID      "PCTFREE 0 INITRANS 4 TABLESPACE PM_SINDEX NOLOGGING"
UI_TEXT             "User Interface text description for SDELOB"
COMMENT             "Any general comment for SDELOB keyword"
END

##WKB_GEOMETRY
GEOMETRY_STORAGE    "OGCWKB"
ATTRIBUTE_BINARY    "BLOB"
RASTER_STORAGE      "BLOB"
UI_TEXT             "User Interface text description for OGC WKB"
END

##LOGFILE_DEFAULTS
LF_STORAGE          "PCTFREE 0 INITRANS 4 TABLESPACE PM_SDELOGFILE"
LF_INDEXES          "PCTFREE 0 INITRANS 4 TABLESPACE PM_SDELOGFILEIDX NOLOGGING"
LD_STORAGE          "PCTFREE 0 INITRANS 4 TABLESPACE PM_SDELOGFILE"
LD_INDEX_ROWID      "PCTFREE 0 INITRANS 4 TABLESPACE PM_SDELOGFILEIDX NOLOGGING"
LD_INDEX_DATA_ID    "PCTFREE 0 INITRANS 4 TABLESPACE PM_SDELOGFILEIDX NOLOGGING"
SESSION_STORAGE     "PCTFREE 0 INITRANS 4 TABLESPACE PM_SDELOGFILE"
SESSION_INDEX       "PCTFREE 0 INITRANS 4 TABLESPACE PM_SDELOGFILEIDX NOLOGGING"
SESSION_TEMP_TABLE 0
END

##NETWORK_DEFAULTS
ATTRIBUTE_BINARY    "BLOB"
B_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE PM_NET_BDATA"
B_INDEX_ROWID       "PCTFREE 0 INITRANS 4 TABLESPACE PM_NET_BINDEX NOLOGGING"
B_INDEX_SHAPE       "PCTFREE 0 INITRANS 4 TABLESPACE PM_NET_BINDEX NOLOGGING"
B_INDEX_USER        "PCTFREE 0 INITRANS 4 TABLESPACE PM_NET_BINDEX NOLOGGING"
F_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE PM_NET_FDATA"
F_INDEX_FID         "PCTFREE 0 INITRANS 4 TABLESPACE PM_NET_FINDEX NOLOGGING"
F_INDEX_AREA        "PCTFREE 0 INITRANS 4 TABLESPACE PM_NET_FINDEX NOLOGGING"
F_INDEX_LEN         "PCTFREE 0 INITRANS 4 TABLESPACE PM_NET_FINDEX NOLOGGING"
S_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE PM_NET_SDATA"
S_INDEX_ALL         "PCTFREE 0 INITRANS 4 TABLESPACE PM_NET_SINDEX NOLOGGING"
S_INDEX_SP_FID      "PCTFREE 0 INITRANS 4 TABLESPACE PM_NET_SINDEX NOLOGGING"
A_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE PM_ADATA"
A_INDEX_ROWID       "PCTFREE 0 INITRANS 4 TABLESPACE PM_AINDEX NOLOGGING"
A_INDEX_SHAPE       "PCTFREE 0 INITRANS 4 TABLESPACE PM_AINDEX NOLOGGING"
A_INDEX_STATEID     "PCTFREE 0 INITRANS 4 TABLESPACE PM_AINDEX NOLOGGING"
A_INDEX_USER        "PCTFREE 0 INITRANS 4 TABLESPACE PM_AINDEX NOLOGGING"
D_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE PM_DDATA"
D_INDEX_STATE_ROWID "PCTFREE 0 INITRANS 4 TABLESPACE PM_DINDEX NOLOGGING"
D_INDEX_DELETED_AT  "PCTFREE 0 INITRANS 4 TABLESPACE PM_DINDEX NOLOGGING"
COMMENT             "The base system initialization parameters for NETWORK_DEFAULTS"
UI_NETWORK_TEXT     "The network default configuration"
END
```

J10066

```
##NETWORK_DEFAULTS::DESC
B_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE PM_NET_BDATA"
B_INDEX_ROWID       "PCTFREE 0 INITRANS 4 TABLESPACE PM_NET_BINDEX NOLOGGING"
B_INDEX_USER        "PCTFREE 0 INITRANS 4 TABLESPACE PM_NET_BINDEX NOLOGGING"
A_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE PM_ADATA"
A_INDEX_ROWID       "PCTFREE 0 INITRANS 4 TABLESPACE PM_AINDEX NOLOGGING"
A_INDEX_STATEID     "PCTFREE 0 INITRANS 4 TABLESPACE PM_AINDEX NOLOGGING"
A_INDEX_USER        "PCTFREE 0 INITRANS 4 TABLESPACE PM_AINDEX NOLOGGING"
D_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE PM_DDATA"
D_INDEX_STATE_ROWID "PCTFREE 0 INITRANS 4 TABLESPACE PM_DINDEX NOLOGGING"
D_INDEX_DELETED_AT  "PCTFREE 0 INITRANS 4 TABLESPACE PM_DINDEX NOLOGGING"
END

##NETWORK_DEFAULTS::NETWORK
B_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE PM_NET_BDATA"
B_INDEX_ROWID       "PCTFREE 0 INITRANS 4 TABLESPACE PM_NET_BINDEX NOLOGGING"
B_INDEX_USER        "PCTFREE 0 INITRANS 4 TABLESPACE PM_NET_BINDEX NOLOGGING"
A_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE PM_ADATA"
A_INDEX_ROWID       "PCTFREE 0 INITRANS 4 TABLESPACE PM_AINDEX NOLOGGING"
A_INDEX_STATEID     "PCTFREE 0 INITRANS 4 TABLESPACE PM_AINDEX NOLOGGING"
A_INDEX_USER        "PCTFREE 0 INITRANS 4 TABLESPACE PM_AINDEX NOLOGGING"
D_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE PM_DDATA"
D_INDEX_STATE_ROWID "PCTFREE 0 INITRANS 4 TABLESPACE PM_DINDEX NOLOGGING"
D_INDEX_DELETED_AT  "PCTFREE 0 INITRANS 4 TABLESPACE PM_DINDEX NOLOGGING"
END

##IMS_GAZETTEER
XML_DOC_LOB_STORAGE        "NOCACHE NOLOGGING CHUNK 4K PCTVERSION 5"
XML_DOC_VAL_LOB_STORAGE    "NOCACHE NOLOGGING CHUNK 4K PCTVERSION 5"
END

##TOPOLOGY_DEFAULTS
UI_TOPOLOGY_TEXT    "The topology default configuration"
B_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE PM_TOPO_BDATA"
B_INDEX_ROWID       "PCTFREE 0 INITRANS 4 TABLESPACE PM_TOPO_BINDEX NOLOGGING"
B_INDEX_SHAPE       "PCTFREE 0 INITRANS 4 TABLESPACE PM_TOPO_BINDEX NOLOGGING"
B_INDEX_USER        "PCTFREE 0 INITRANS 4 TABLESPACE PM_TOPO_BINDEX NOLOGGING"
F_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE PM_TOPO_FDATA"
F_INDEX_AREA        "PCTFREE 0 INITRANS 4 TABLESPACE PM_TOPO_FINDEX NOLOGGING"
F_INDEX_FID         "PCTFREE 0 INITRANS 4 TABLESPACE PM_TOPO_FINDEX NOLOGGING"
F_INDEX_LEN         "PCTFREE 0 INITRANS 4 TABLESPACE PM_TOPO_FINDEX NOLOGGING"
S_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE PM_TOPO_SDATA"
S_INDEX_ALL         "PCTFREE 0 INITRANS 4 TABLESPACE PM_TOPO_SINDEX NOLOGGING"
S_INDEX_SP_FID      "PCTFREE 0 INITRANS 4 TABLESPACE PM_TOPO_SINDEX NOLOGGING"
A_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE PM_ADATA"
A_INDEX_ROWID       "PCTFREE 0 INITRANS 4 TABLESPACE PM_AINDEX NOLOGGING"
A_INDEX_STATEID     "PCTFREE 0 INITRANS 4 TABLESPACE PM_AINDEX NOLOGGING"
A_INDEX_SHAPE       "PCTFREE 0 INITRANS 4 TABLESPACE PM_AINDEX NOLOGGING"
A_INDEX_USER        "PCTFREE 0 INITRANS 4 TABLESPACE PM_AINDEX NOLOGGING"
D_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE PM_DDATA"
D_INDEX_DELETED_AT  "PCTFREE 0 INITRANS 4 TABLESPACE PM_DINDEX NOLOGGING"
D_INDEX_STATE_ROWID "PCTFREE 0 INITRANS 4 TABLESPACE PM_DINDEX NOLOGGING"
END

##TOPOLOGY_DEFAULTS::DIRTYAREAS
B_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE PM_TOPO_BDATA"
B_INDEX_ROWID       "PCTFREE 0 INITRANS 4 TABLESPACE PM_TOPO_BINDEX NOLOGGING"
B_INDEX_SHAPE       "PCTFREE 0 INITRANS 4 TABLESPACE PM_TOPO_BINDEX NOLOGGING"
B_INDEX_USER        "PCTFREE 0 INITRANS 4 TABLESPACE PM_TOPO_BINDEX NOLOGGING"
F_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE PM_TOPO_FDATA"
F_INDEX_AREA        "PCTFREE 0 INITRANS 4 TABLESPACE PM_TOPO_FINDEX NOLOGGING"
F_INDEX_FID         "PCTFREE 0 INITRANS 4 TABLESPACE PM_TOPO_FINDEX NOLOGGING"
F_INDEX_LEN         "PCTFREE 0 INITRANS 4 TABLESPACE PM_TOPO_FINDEX NOLOGGING"
S_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE PM_TOPO_SDATA"
S_INDEX_ALL         "PCTFREE 0 INITRANS 4 TABLESPACE PM_TOPO_SINDEX NOLOGGING"
S_INDEX_SP_FID      "PCTFREE 0 INITRANS 4 TABLESPACE PM_TOPO_SINDEX NOLOGGING"
A_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE PM_ADATA"
A_INDEX_ROWID       "PCTFREE 0 INITRANS 4 TABLESPACE PM_AINDEX NOLOGGING"
A_INDEX_STATEID     "PCTFREE 0 INITRANS 4 TABLESPACE PM_AINDEX NOLOGGING"
A_INDEX_SHAPE       "PCTFREE 0 INITRANS 4 TABLESPACE PM_AINDEX NOLOGGING"
A_INDEX_USER        "PCTFREE 0 INITRANS 4 TABLESPACE PM_AINDEX NOLOGGING"
D_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE PM_DDATA"
D_INDEX_DELETED_AT  "PCTFREE 0 INITRANS 4 TABLESPACE PM_DINDEX NOLOGGING"
D_INDEX_STATE_ROWID "PCTFREE 0 INITRANS 4 TABLESPACE PM_DINDEX NOLOGGING"
END

##TERRAIN_DEFAULTS
UI_TERRAIN_TEXT    "The terrain default configuration"
B_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE PM_TERRAIN_BDATA"
B_INDEX_ROWID       "PCTFREE 0 INITRANS 4 TABLESPACE PM_TERRAIN_BINDEX NOLOGGING"
B_INDEX_SHAPE       "PCTFREE 0 INITRANS 4 TABLESPACE PM_TERRAIN_BINDEX NOLOGGING"
B_INDEX_USER        "PCTFREE 0 INITRANS 4 TABLESPACE PM_TERRAIN_BINDEX NOLOGGING"
F_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE PM_TERRAIN_FDATA"
F_INDEX_AREA        "PCTFREE 0 INITRANS 4 TABLESPACE PM_TERRAIN_FINDEX NOLOGGING"
F_INDEX_FID         "PCTFREE 0 INITRANS 4 TABLESPACE PM_TERRAIN_FINDEX NOLOGGING"
F_INDEX_LEN         "PCTFREE 0 INITRANS 4 TABLESPACE PM_TERRAIN_FINDEX NOLOGGING"
S_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE PM_TERRAIN_SDATA"
S_INDEX_ALL         "PCTFREE 0 INITRANS 4 TABLESPACE PM_TERRAIN_SINDEX NOLOGGING"
S_INDEX_SP_FID      "PCTFREE 0 INITRANS 4 TABLESPACE PM_TERRAIN_SINDEX NOLOGGING"
```

```
A_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE PM_ADATA"
A_INDEX_ROWID       "PCTFREE 0 INITRANS 4 TABLESPACE PM_AINDEX NOLOGGING"
A_INDEX_STATEID     "PCTFREE 0 INITRANS 4 TABLESPACE PM_AINDEX NOLOGGING"
A_INDEX_SHAPE       "PCTFREE 0 INITRANS 4 TABLESPACE PM_AINDEX NOLOGGING"
A_INDEX_USER        "PCTFREE 0 INITRANS 4 TABLESPACE PM_AINDEX NOLOGGING"
D_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE PM_DDATA"
D_INDEX_DELETED_AT  "PCTFREE 0 INITRANS 4 TABLESPACE PM_DINDEX NOLOGGING"
D_INDEX_STATE_ROWID "PCTFREE 0 INITRANS 4 TABLESPACE PM_DINDEX NOLOGGING"
END

##TERRAIN_DEFAULTS::EMBEDDED
B_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE PM_TERRAIN_BDATA"
B_INDEX_ROWID       "PCTFREE 0 INITRANS 4 TABLESPACE PM_TERRAIN_BINDEX NOLOGGING"
B_INDEX_SHAPE       "PCTFREE 0 INITRANS 4 TABLESPACE PM_TERRAIN_BINDEX NOLOGGING"
B_INDEX_USER        "PCTFREE 0 INITRANS 4 TABLESPACE PM_TERRAIN_BINDEX NOLOGGING"
F_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE PM_TERRAIN_FDATA"
F_INDEX_AREA        "PCTFREE 0 INITRANS 4 TABLESPACE PM_TERRAIN_FINDEX NOLOGGING"
F_INDEX_FID         "PCTFREE 0 INITRANS 4 TABLESPACE PM_TERRAIN_FINDEX NOLOGGING"
F_INDEX_LEN         "PCTFREE 0 INITRANS 4 TABLESPACE PM_TERRAIN_FINDEX NOLOGGING"
S_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE PM_TERRAIN_SDATA"
S_INDEX_ALL         "PCTFREE 0 INITRANS 4 TABLESPACE PM_TERRAIN_SINDEX NOLOGGING"
S_INDEX_SP_FID      "PCTFREE 0 INITRANS 4 TABLESPACE PM_TERRAIN_SINDEX NOLOGGING"
A_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE PM_ADATA"
A_INDEX_ROWID       "PCTFREE 0 INITRANS 4 TABLESPACE PM_AINDEX NOLOGGING"
A_INDEX_STATEID     "PCTFREE 0 INITRANS 4 TABLESPACE PM_AINDEX NOLOGGING"
A_INDEX_SHAPE       "PCTFREE 0 INITRANS 4 TABLESPACE PM_AINDEX NOLOGGING"
A_INDEX_USER        "PCTFREE 0 INITRANS 4 TABLESPACE PM_AINDEX NOLOGGING"
D_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE PM_DDATA"
D_INDEX_DELETED_AT  "PCTFREE 0 INITRANS 4 TABLESPACE PM_DINDEX NOLOGGING"
D_INDEX_STATE_ROWID "PCTFREE 0 INITRANS 4 TABLESPACE PM_DINDEX NOLOGGING"
END

##ST_RASTER
RASTER_STORAGE            "ST_RASTER"
UI_TEXT                   "UI_TEXT"
END
```

- Import the modified ***dbtune_pm.sde*** file.

```
sdedbtune -o import -f dbtune_pm.sde -u sde -p sde -i sde:oracle11g:mcs
```

Step 4: Configure Oracle Parameters

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

Oracle Parameters

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 need to configure the MAXBLOBSIZE and TCPKEEPALIVE parameters for the ArcSDE geodatabase used as the product library. 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 TRUE. This command should be used at 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=TRUE -i <service> -u sde -p <sde_password>
```

For more information, see the ArcSDE Administration Command Reference.

Step 6: Load the Data

Prepare your geodatabase for loading data.

- Back up your database.
- Change the ArcSDE buffer size:

```
sdeconfig -o alter -i sde:oracle11g:mcs -u sde -p sdeadmin  -N -q -v MINBUFSIZE=409600
sdeconfig -o alter -i sde:oracle11g:mcs -u sde -p sdeadmin -N -q -v MAXBUFSIZE=819200
```

- Set the ArcSDE temp folder:

```
sdeconfig -o alter -i sde:oracle11g:mcs -u sde -p sdeadmin -N -q -v TEMP=D:\TEMP
```

- List ArcSDE parameters:

```
sdeconfig -o list -i sde:oracle11g:mcs -u sde -p sdeadmin
```

- Set the database to no archive log mode:

```
sqlplus / as sysdba
shutdown immediate;
startup mount;
alter database flashback off;
alter database noarchivelog;
alter database open;
select flashback_on from v$database;
archive log list;
```

- Disable tablespace logging:
  - Logging—Generate redo logs for creation of tables, indexes, and partitions and for subsequent inserts. This data is recoverable.
  - No Logging—Redo log entries are smaller, so the above operations are not logged and not recoverable. Data loading of large feature classes and tables is faster.

```
sqlplus /nolog
connect / as sysdba;

select 'ALTER TABLESPACE ' || CHR(34) || NAME || CHR(34) || ' NOLOGGING;'
  FROM V$TABLESPACE
  WHERE NAME NOT IN ('SYSTEM','SYSAUX','USERS','TEMP','UNDOTBS1') ORDER BY NAME;

ALTER TABLESPACE "PM_BDATA" NOLOGGING;
ALTER TABLESPACE "PM_BINDEX" NOLOGGING;
ALTER TABLESPACE "PM_FDATA" NOLOGGING;
ALTER TABLESPACE "PM_FINDEX" NOLOGGING;
ALTER TABLESPACE "PM_SDATA" NOLOGGING;
ALTER TABLESPACE "PM_SINDEX" NOLOGGING;
ALTER TABLESPACE "PM_ADATA" NOLOGGING;
```

```
ALTER TABLESPACE "PM_AINDEX" NOLOGGING;
ALTER TABLESPACE "PM_DDATA" NOLOGGING;
ALTER TABLESPACE "PM_DINDEX" NOLOGGING;
ALTER TABLESPACE "PM_RASTER" NOLOGGING;
ALTER TABLESPACE "PM_RINDEX" NOLOGGING;
ALTER TABLESPACE "PM_RBLK" NOLOGGING;
ALTER TABLESPACE "PM_RBLKIDX" NOLOGGING;
ALTER TABLESPACE "PM_XMLDOC" NOLOGGING;
ALTER TABLESPACE "PM_XMLIDX" NOLOGGING;
ALTER TABLESPACE "PM_TOPO_BDATA " NOLOGGING;
ALTER TABLESPACE "PM_TOPO_BINDEX " NOLOGGING;
ALTER TABLESPACE "PM_TOPO_FDATA " NOLOGGING;
ALTER TABLESPACE "PM_TOPO_FINDEX " NOLOGGING;
ALTER TABLESPACE "PM_TOPO_SDATA " NOLOGGING;
ALTER TABLESPACE "PM_TOPO_SINDEX " NOLOGGING;
ALTER TABLESPACE "PM_TERRAIN_BDATA " NOLOGGING;
ALTER TABLESPACE "PM_TERRAIN_BINDEX " NOLOGGING;
ALTER TABLESPACE "PM_TERRAIN_FDATA " NOLOGGING;
ALTER TABLESPACE "PM_TERRAIN_FINDEX " NOLOGGING;
ALTER TABLESPACE "PM_TERRAIN_SDATA " NOLOGGING;
ALTER TABLESPACE "PM_TERRAIN_SINDEX " NOLOGGING;
ALTER TABLESPACE "PM_NET_BDATA " NOLOGGING;
ALTER TABLESPACE "PM_NET_BINDEX " NOLOGGING;
ALTER TABLESPACE "PM_NET_FDATA " NOLOGGING;
ALTER TABLESPACE "PM_NET_FINDEX " NOLOGGING;
ALTER TABLESPACE "PM_NET_SDATA " NOLOGGING;
ALTER TABLESPACE "PM_NET_SINDEX " NOLOGGING;
```

- Load 10 percent of the data, estimate the total size of each data file, and then resize the data files accordingly.

  If loading or appending data to an existing feature class, even if the feature class is empty but you have to load a large amount of data, change the layer I/O mode to load_only_io. The loading will be faster because indexes are disabled.

```
sdelayer –o  load_only_io -l contour_l,shape -i sde:oracle11g:mcs -s mysrv -u sde -p sde
```

- Load your data model with the PM user and then load the production mapping data.
- Back up your database.

Step 7: Register as Versioned

In ArcCatalog™, register the PM schema as versioned.

Step 8: Verify Storage

Run the SQL queries below to verify that the PM tables and indexes were created under the correct tablespaces:

```
sqlplus pm/pm@mcs

--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), read Oracle MetaLink Doc ID 130814.1, "How to Move LOB Data to Another Tablespace."

Step 9: Prepare Geodatabase for Editing

Prepare the geodatabase for normal online transaction processing (OLTP) editing.

- After loading the data, change the layer to normal I/O:

```
sdelayer –o  normal_io -l contour_l,shape -i sde:oracle11g:mcs -s mysrv -u sde -p sde
```

- Change the ArcSDE buffer size:

```
sdeconfig -o alter -i sde:oracle11g:mcs -u sde -p sdeadmin  -N -q -v MINBUFSIZE=16384
sdeconfig -o alter -i sde:oracle11g:mcs -u sde -p sdeadmin -N -q -v MAXBUFSIZE=65536
```

- Enable tablespace logging:

```
sqlplus /nolog
connect / as sysdba;

select 'ALTER TABLESPACE ' || CHR(34) || NAME || CHR(34) || ' LOGGING;'
  FROM V$TABLESPACE
  WHERE NAME NOT IN ('SYSTEM','SYSAUX','USERS','TEMP','UNDOTBS1') ORDER BY NAME;

ALTER TABLESPACE "PM_BDATA" LOGGING;
ALTER TABLESPACE "PM_BINDEX" LOGGING;
ALTER TABLESPACE "PM_FDATA" LOGGING;
ALTER TABLESPACE "PM_FINDEX" LOGGING;
ALTER TABLESPACE "PM_SDATA" LOGGING;
ALTER TABLESPACE "PM_SINDEX" LOGGING;
ALTER TABLESPACE "PM_ADATA" LOGGING;
ALTER TABLESPACE "PM_AINDEX" LOGGING;
ALTER TABLESPACE "PM_DDATA" LOGGING;
ALTER TABLESPACE "PM_DINDEX" LOGGING;
ALTER TABLESPACE "PM_RASTER" LOGGING;
ALTER TABLESPACE "PM_RINDEX" LOGGING;
ALTER TABLESPACE "PM_RBLK" LOGGING;
ALTER TABLESPACE "PM_RBLKIDX" LOGGING;
ALTER TABLESPACE "PM_XMLDOC" LOGGING;
ALTER TABLESPACE "PM_XMLIDX" LOGGING;
ALTER TABLESPACE "PM_TOPO_BDATA " LOGGING;
ALTER TABLESPACE "PM_TOPO_BINDEX " LOGGING;
ALTER TABLESPACE "PM_TOPO_FDATA " LOGGING;
ALTER TABLESPACE "PM_TOPO_FINDEX " LOGGING;
ALTER TABLESPACE "PM_TOPO_SDATA " LOGGING;
ALTER TABLESPACE "PM_TOPO_SINDEX " LOGGING;
ALTER TABLESPACE "PM_TERRAIN_BDATA " LOGGING;
ALTER TABLESPACE "PM_TERRAIN_BINDEX " LOGGING;
ALTER TABLESPACE "PM_TERRAIN_FDATA " LOGGING;
ALTER TABLESPACE "PM_TERRAIN_FINDEX " LOGGING;
ALTER TABLESPACE "PM_TERRAIN_SDATA " LOGGING;
ALTER TABLESPACE "PM_TERRAIN_SINDEX " LOGGING;
ALTER TABLESPACE "PM_NET_BDATA " LOGGING;
ALTER TABLESPACE "PM_NET_BINDEX " LOGGING;
ALTER TABLESPACE "PM_NET_FDATA " LOGGING;
ALTER TABLESPACE "PM_NET_FINDEX " LOGGING;
ALTER TABLESPACE "PM_NET_SDATA " LOGGING;
ALTER TABLESPACE "PM_NET_SINDEX " LOGGING;
```

- Set the database to archive log mode:

```
sqlplus / as sysdba
alter system set db_recovery_file_dest_size=10G scope=spfile;
alter system set db_recovery_file_dest='D:\oradata\flash_recovery_area' scope=spfile;
alter system set log_archive_dest_1='LOCATION=USE_DB_RECOVERY_FILE_DEST' scope=spfile;
shutdown immediate;
startup mount;
alter database archivelog;
alter database flashback on;
alter database open;
select flashback_on from v$database;
archive log list;
```

- Back up your database.

Step 10: Grant Permissions and Roles

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

```
SET SERVEROUTPUT ON;

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

- Grant the PMEDITOR role to ArcSDE editor users and the PMVIEWER role to ArcSDE viewer users.

Step 11: Configure 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 have more than 100 records for each connected ArcSDE editor/viewer user. Hence, it is recommended that you 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 PM_SDELOGFILE
DATAFILE ' D:\oradata\MCS\PM\pm_sdelogfile01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 800M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 512K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';

CREATE SMALLFILE TABLESPACE PM_SDELOGFILEIDX
DATAFILE D:\oradata\MCS\PM\pm_sdelogfileidx01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 512K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';
```

Change DBTUNE Log File Parameters

- Export the DBTUNE table.

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

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

```
##LOGFILE_DEFAULTS
LD_INDEX_DATA_ID         "PCTFREE 0 INITRANS 4 TABLESPACE PM_SDELOGFILEIDX NOLOGGING "
LF_INDEXES               "PCTFREE 0 INITRANS 4 TABLESPACE PM_SDELOGFILEIDX NOLOGGING "
LF_STORAGE               "PCTFREE 0 INITRANS 4 TABLESPACE PM_SDELOGFILE"
SESSION_INDEX            "PCTFREE 0 INITRANS 4 TABLESPACE PM_SDELOGFILEIDX NOLOGGING "
```

```
SESSION_TEMP_TABLE       0
SESSION_STORAGE          "PCTFREE 0 INITRANS 4 TABLESPACE PM_SDELOGFILE"
LD_STORAGE               "PCTFREE 0 INITRANS 4 TABLESPACE PM_SDELOGFILE "
LD_INDEX_ROWID           "PCTFREE 0 INITRANS 4 TABLESPACE PM_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 "PM_SDELOGFILE";
ALTER USER <user_name> QUOTA UNLIMITED ON "PM_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 Log file table configuration options for geodatabases in Oracle.

Esri Knowledge Base—Technical Articles

Article ID 32005—How to Utilize Oracle's optimizer_dynamic_sampling

Article ID 32164—How to Lock SDE_LOGFILE_DATA table statistics with Oracle

Article ID 37841—Problem Slow performance making large selections from SDO_Geometry feature classes in ArcMap

Step 12: Create ArcSDE Users

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";
/*-- PMEDITOR role has SELECT, INSERT, UPDATE and DELETE permission on the PM data (featureclasses,
tables, etc.)*/
GRANT "PMEDITOR" TO "GIS_EDITOR";
ALTER USER GIS_EDITOR QUOTA UNLIMITED ON "PM_SDELOGFILE";
ALTER USER GIS_EDITOR QUOTA UNLIMITED ON "PM_SDELOGFILEIDX";
CREATE USER GIS_VIEWER PROFILE "DEFAULT"
IDENTIFIED BY viewer
```

Viewer User

```
DEFAULT TABLESPACE "USERS"
TEMPORARY TABLESPACE "TEMP" ACCOUNT UNLOCK;
GRANT "CONNECT" TO "GIS_VIEWER";
GRANT CREATE TABLE TO "GIS_VIEWER";
/*-- PMVIEWER role has SELECT  permission on the PM data (featureclasses, tables, etc.)*/
GRANT "PMVIEWER" TO "GIS_VIEWER";
ALTER USER GIS_VIEWER QUOTA UNLIMITED ON "PM_SDELOGFILE";
ALTER USER GIS_VIEWER QUOTA UNLIMITED ON "PM_SDELOGFILEIDX";
```

Conclusion

You can reduce disk contention and improve database I/O by storing the production mapping data 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 at the following:

- Minimize disk I/O contention in Oracle
- What type of maintenance is needed for a geodatabase?

For more information on Esri Production Mapping, visit esri.com/productionmapping or e-mail productionmapping@esri.com.

Access blogs, forums, downloads, and more, from the Esri Production Mapping resource center.

You can access other resources at ArcGIS 10.2 for Desktop Help and Esri Support.

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

Printed in USA