

Best Practices for Storing the ArcGIS® Workflow Manager Workspace in an Enterprise Geodatabase for Oracle

An Esri® White Paper
October 2013



Copyright © 2013 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, 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 Workflow Manager Workspace in an Enterprise Geodatabase for Oracle

An Esri White Paper

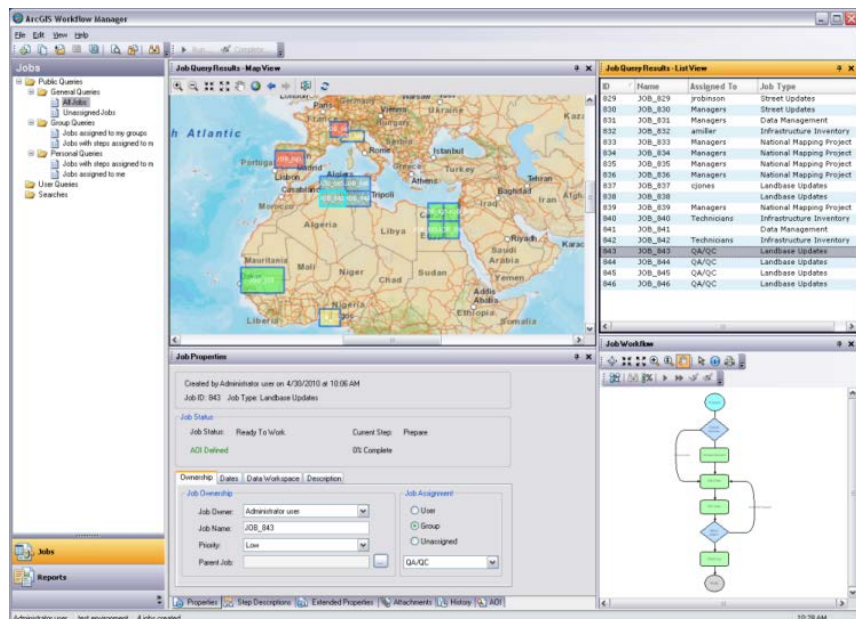
Contents	Page
Introduction.....	1
Postinstallation Utility	2
Preparing the Database	2
ArcSDE DBTUNE.....	2
Disk Configuration.....	3
Reducing Disk I/O Contention.....	3
Transparent Data Encryption	3
Using Data Compression	6
Compression and TDE.....	7
Step 1: Create Data Files.....	7
Step 2: Create Workflow Manager User.....	8
Step 3: Modify DBTUNE	8
Step 4: Configure Oracle Parameters.....	9
Step 5: Configure ArcSDE Parameters.....	10
Step 6: Create the Workspace	10

Contents	Page
Step 7: Verify Storage.....	15
Step 8: Grant Permissions and Roles	16
Step 9: Configure Log File Tables.....	16
Create Log File Tablespaces	17
Change DBTUNE Log File Parameters.....	17
Create Log File Tables.....	17
Step 10: Create Database Users	18
Editor User	18
Viewer User	18
Replication	18
Conclusion	19

Best Practices for Storing the ArcGIS Workflow Manager Workspace in an Enterprise Geodatabase for Oracle

Introduction

This white paper is intended to help database administrators establish the ArcGIS® Workflow Manager workspace in an enterprise geodatabase for Oracle®. The enterprise geodatabase uses ArcSDE® technology as the gateway between geographic information system (GIS) clients and Oracle. The Workflow Manager workspace uses ArcSDE, and this paper covers the best practices to create the ArcSDE instance. ArcGIS Workflow Manager is an enterprise workflow management application that provides an integration framework for ArcGIS multiuser geodatabase environments. It simplifies many aspects of job management and tracking and streamlines the workflow, resulting in significant time savings for any implementation. Workflow Manager provides tools for allocating resources and tracking the status and progress of jobs.



ArcGIS Workflow Manager Client Application Interface

A detailed history is automatically recorded for each job to give managers a complete report on how the job was completed. This information can be supplemented with comments and notes to provide even richer job documentation. Workflow Manager

handles complex geodatabase tasks behind the scenes by assisting the user in the creation and management of versions. Integration between Workflow Manager and ArcGIS geodatabase tools provides a way of tracking feature edits made through Workflow Manager jobs using the geodatabase archiving tools.

Postinstallation Utility

For the Workflow Manager applications to function, all the Workflow Manager system tables must exist in the database, and of those, the tables that host required configuration information must be populated with values that will drive the application behavior. The following section will detail how to create and start configuring the Workflow Manager system tables.

The Workflow Manager postinstallation utility is used to insert the Workflow Manager system tables into the workspace you've created for the Workflow Manager repository (also referred to as the Workflow Manager database).

This repository contains a set of geodatabase tables (Workflow Manager system tables) used to store the job and configuration information for your work management system and one feature class that is used to store the geometries for the areas of interest (AOIs) for your jobs.

The Workflow Manager system tables added by the postinstallation utility are geodatabase tables and are registered with your ArcSDE geodatabase. If at any time you are unsure of the state of the database, connect to the repository using ArcCatalog™. Because the Workflow Manager system tables are geodatabase tables, it is strongly recommended that you do not delete or modify these tables using your RDBMS tools.

Preparing the Database

Before you can install the Workflow Manager system tables into an ArcSDE instance, you must properly set up the instance to receive the repository; the steps below will show how to accomplish this task.

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 Workflow Manager system tables into 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

J10042

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 so more spindles can actively search for it. This can increase disk read time and decrease disk contention. However, involving 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/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 the autoextend by specifying a maximum growth size that leaves some hard disk space available. By putting 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

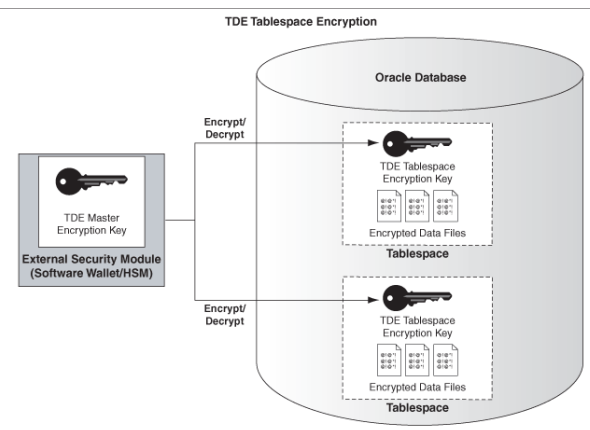
Transparent Data Encryption

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 TDE. 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 the data is transparently decrypted.
- Applications need not be modified to handle encrypted data. Data encryption and decryption is managed by the database.
- Key management operations are automated. The user or application does not need to manage encryption keys.



TDE Tablespace Encryption

J10042

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

```
-----*-----
--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\dba_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\dba_ora
      Password: xxxxxxxx

      Wallet Password: wmxadmin

      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 "wmxadmin";

--See the status of the wallet

select * from v$encryption_wallet;

--Make the wallet auto login

set ORACLE_SID=wxmdb

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

--Backup the wallet folder

cd C:\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 Oracle9i. 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 11g extends the compression technology to support these operations as well. Consequently, compression in Oracle Database 11g can be used for all kinds of workload—online transaction processing (OLTP) or data warehousing. It is important to mention that table compression enhancements introduced in Oracle database 11g 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 11g 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 above.

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.

J10042

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 11g: Transparent Data Encryption Frequently Asked Questions](#)

Step 1: Create Data Files

Create new tablespaces to store the Workflow Manager system tables.

```

TABLESPACE      ArcSDE_PARAMETER
WMX_BDATA       Business table
WMX_BINDEX      Business table index
WMX_FDATA       Feature table
WMX_FINDEX      Feature table index
WMX_SDATA       Spatial Index table
WMX_SINDEX      Spatial Index table index
WMX_ADATA       Adds table (versioned)
WMX_AINDEX      Adds table index
WMX_DDATA       Deletes table (versioned)
WMX_DINDEX      Deletes table index

ALTER SYSTEM SET WALLET OPEN IDENTIFIED BY "wmxadmin";

CREATE SMALLFILE TABLESPACE "WMX_BDATA"
DATAFILE D:\oracle\ORADATA\WMXDB\WMX_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 "WMX_BINDEX"
DATAFILE D:\oracle\ORADATA\WMXDB\WMX_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 "WMX_FDATA"
DATAFILE D:\oracle\ORADATA\WMXDB\WMX_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 "WMX_FINDEX"
DATAFILE D:\oracle\ORADATA\WMXDB\WMX_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 "WMX_SDATA"
DATAFILE D:\oracle\ORADATA\WMXDB\WMX_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 "WMX_SINDEX"
DATAFILE D:\oracle\ORADATA\WMXDB\WMX_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 "WMX_ADATA"
DATAFILE D:\oracle\ORADATA\WMXDB\WMX_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 "WMX_AINDEX"
DATAFILE D:\oracle\ORADATA\WMXDB\WMX_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 "WMX_DDATA"
DATAFILE D:\oracle\ORADATA\WMXDB\WMX_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 "WMX_DINDEX"
DATAFILE D:\oracle\ORADATA\WMXDB\WMX_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';
```

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\WMXDB\wmx_bdata01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\oracle\ORADATA\WMXDB\wmx_bindex01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\oracle\ORADATA\WMXDB\wmx_fdata01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\oracle\ORADATA\WMXDB\wmx_findex01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\oracle\ORADATA\WMXDB\wmx_sdata01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\oracle\ORADATA\WMXDB\wmx_sindex01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\oracle\ORADATA\WMXDB\wmx_adata01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\oracle\ORADATA\WMXDB\wmx_aindex01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\oracle\ORADATA\WMXDB\wmx_ddata01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\oracle\ORADATA\WMXDB\wmx_dindex01.dbf' RESIZE 400M;
```

Step 2: Create Workflow Manager User

Create a new database user to store the Workflow Manager system tables and grant the appropriate permissions.

```
CREATE USER WMX PROFILE DEFAULT IDENTIFIED BY wmx
DEFAULT TABLESPACE "WMX_BDATA"
TEMPORARY TABLESPACE "TEMP"
QUOTA UNLIMITED ON "WMX_BDATA"
QUOTA UNLIMITED ON "WMX_BINDEX"
QUOTA UNLIMITED ON "WMX_FDATA"
QUOTA UNLIMITED ON "WMX_FINDEX"
QUOTA UNLIMITED ON "WMX_SDATA"
QUOTA UNLIMITED ON "WMX_SINDEX"
QUOTA UNLIMITED ON "WMX_ADATA"
QUOTA UNLIMITED ON "WMX_AINDEX"
QUOTA UNLIMITED ON "WMX_DDATA"
QUOTA UNLIMITED ON "WMX_DINDEX"
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 WMX;
```

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:wmxdb
```

Copy *dbtune_exp.sde* to *dbtune_wmx.sde*.

J10042

- Create a new DBTUNE keyword following the steps below:
- Copy the ##DEFAULTS keyword and paste it at the end of the dbtune_wmx.sde file.
- Rename it ##WMX and change the tablespace name for the appropriate parameters.

```
dbtune_wmx.sde

##WMX
A_INDEX_RASTER "PCTFREE 0 INITRANS 4 TABLESPACE WMX_AINDEX NOLOGGING"
A_INDEX_ROWID "PCTFREE 0 INITRANS 4 TABLESPACE WMX_AINDEX STORAGE (INITIAL 409600) NOLOGGING"
A_INDEX_SHAPE "PCTFREE 0 INITRANS 4 TABLESPACE WMX_AINDEX STORAGE (INITIAL 409600) NOLOGGING"
A_INDEX_STATEID "PCTFREE 0 INITRANS 4 TABLESPACE WMX_AINDEX STORAGE (INITIAL 409600) NOLOGGING"
A_INDEX_USER "PCTFREE 0 INITRANS 4 TABLESPACE WMX_AINDEX STORAGE (INITIAL 409600) NOLOGGING"
A_INDEX_XML "PCTFREE 0 INITRANS 4 TABLESPACE WMX_AINDEX NOLOGGING"
A_STORAGE "PCTFREE 0 INITRANS 4 TABLESPACE WMX_ADATA STORAGE (INITIAL 409600)"
B_INDEX_RASTER "PCTFREE 0 INITRANS 4 TABLESPACE WMX_BINDEX NOLOGGING"
B_INDEX_ROWID "PCTFREE 0 INITRANS 4 TABLESPACE WMX_BINDEX STORAGE (INITIAL 409600) NOLOGGING"
B_INDEX_SHAPE "PCTFREE 0 INITRANS 4 TABLESPACE WMX_BINDEX NOLOGGING"
B_INDEX_TO_DATE "PCTFREE 0 INITRANS 4 TABLESPACE WMX_BINDEX NOLOGGING"
B_INDEX_USER "PCTFREE 0 INITRANS 4 TABLESPACE WMX_BINDEX STORAGE (INITIAL 409600) NOLOGGING"
B_INDEX_XML "PCTFREE 0 INITRANS 4 TABLESPACE WMX_BINDEX NOLOGGING"
B_STORAGE "PCTFREE 0 INITRANS 4 TABLESPACE WMX_BDATA STORAGE (INITIAL 409600)"
D_INDEX_DELETED_AT "PCTFREE 0 INITRANS 4 TABLESPACE WMX_DINDEX STORAGE (INITIAL 409600) NOLOGGING"
D_INDEX_STATE_ROWID "PCTFREE 0 INITRANS 4 TABLESPACE WMX_DINDEX STORAGE (INITIAL 409600) NOLOGGING"
D_STORAGE "PCTFREE 0 INITRANS 4 TABLESPACE WMX_DDATA STORAGE (INITIAL 409600)"
F_INDEX_AREA "PCTFREE 0 INITRANS 4 TABLESPACE WMX_FINDEX NOLOGGING"
F_INDEX_FID "PCTFREE 0 INITRANS 4 TABLESPACE WMX_FINDEX NOLOGGING"
F_INDEX_LEN "PCTFREE 0 INITRANS 4 TABLESPACE WMX_FINDEX NOLOGGING"
F_STORAGE "PCTFREE 0 INITRANS 4 TABLESPACE WMX_FDATA"
GEOMETRY_STORAGE "ST_GEOMETRY"
S_INDEX_ALL "PCTFREE 0 INITRANS 4 TABLESPACE WMX_SINDEX STORAGE (INITIAL 409600) NOLOGGING"
S_INDEX_SP_FID "PCTFREE 0 INITRANS 4 TABLESPACE WMX_SINDEX NOLOGGING"
S_STORAGE "PCTFREE 0 INITRANS 4 TABLESPACE WMX_SDATA STORAGE (INITIAL 409600)"
ST_GEOM_LOB_STORAGE "STORE AS (TABLESPACE WMX_FDATA ENABLE STORAGE IN ROW CHUNK 8K RETENTION CACHE
INDEX (TABLESPACE WMX_FINDEX) )"
END
```

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

- Import the modified *dbtune_wmx.sde* file.

```
sdedbtune -o import -f dbtune_wmx.sde -u sde -p sde
-i sde:oracle11g:wmxdb
```

Step 4: Configure Oracle Parameters

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

Oracle Parameters for Workflow Manager

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

Parameter Name	Value
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 Workflow Manager repository. 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 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=TRUE -i <service> -u sde -p <sde_password>
```

For more information, see the ArcSDE Administration Command Reference.

Step 6: Create the Workspace

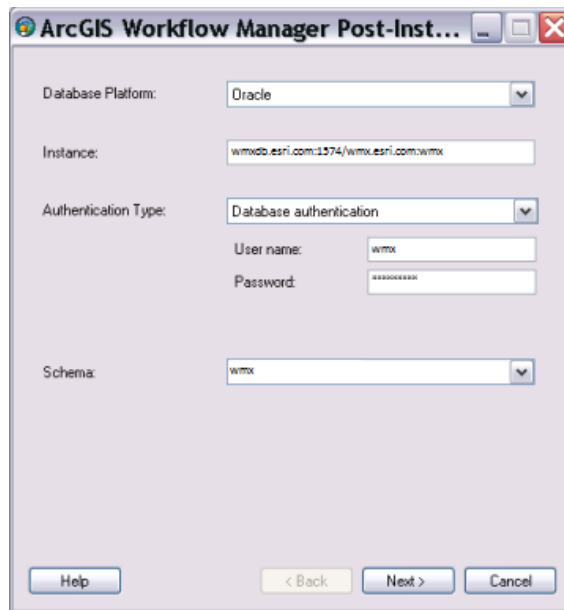
The postinstallation utility creates all the system tables required for you to use ArcGIS Workflow Manager.

- a. From the **Start** menu, navigate to **ArcGIS Workflow Manager** and select **Workflow Manager Post Installation**.

The **Workflow Manager Post-Installation** wizard appears.

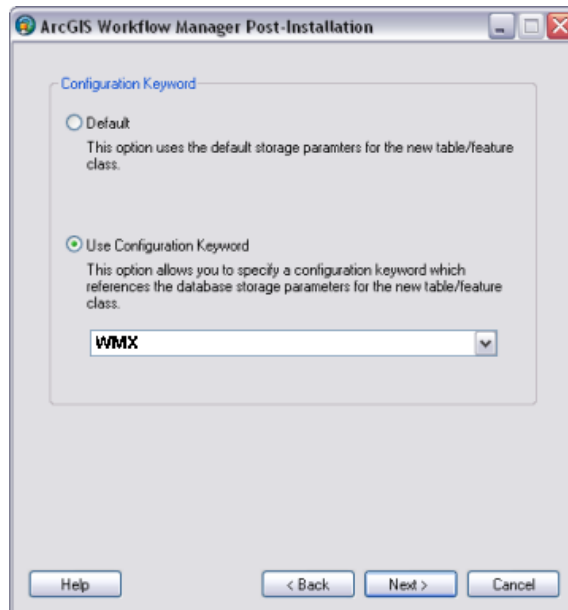
- b. Enter the connection information for the designated workspace, then test the connection information.

J10042



Enter connection information for the ArcSDE geodatabase that will be the container for the Workflow Manager repository.

- c. Click **Next**.
- d. Choose a configuration keyword for your Workflow Manager geodatabase; **use the WMX keyword created in step 3.**

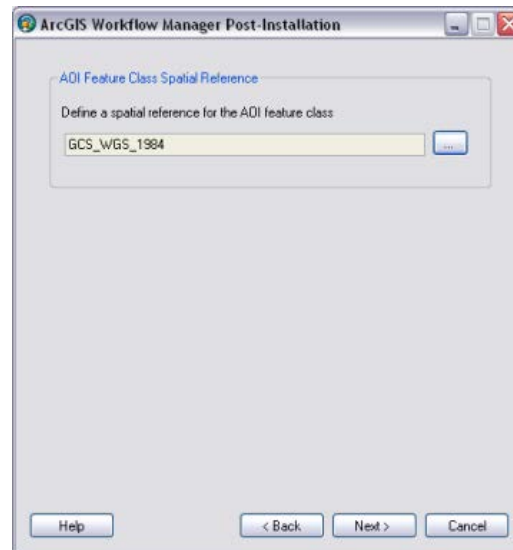


Choose the configuration keyword.

- e. Click **Next**.

- f. Define the spatial reference for the AOI feature class.

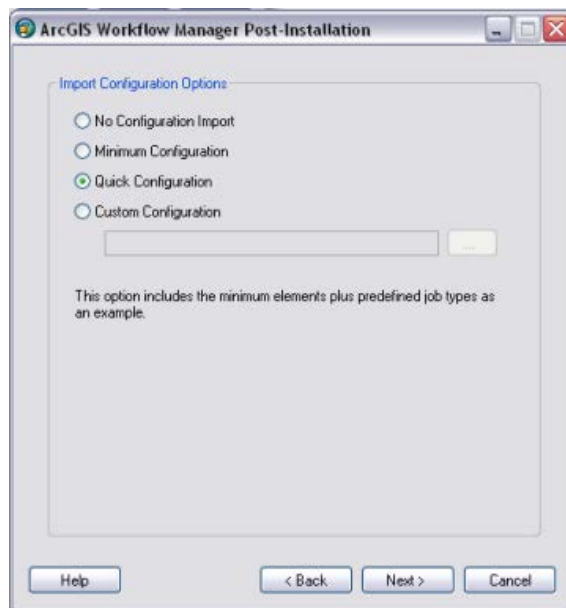
A separate, two-page wizard appears, allowing you to select the coordinate system and extents.



Define the spatial reference for the AOI feature class.

- g. Click **Next**.
- h. Choose your starting configuration import option for the Workflow Manager repository.
- **No configuration** will put the Workflow Manager system tables schema into your designated geodatabase and will not prepopulate any of the configuration elements.
 - **Minimum configuration** imports just the basic elements that the Workflow Manager system requires.
 - **Quick configuration** includes the minimum elements plus predefined job types as an example.
 - **Custom configuration** allows you to import a preexisting Workflow Manager configuration file.

J10042



Choose the starting configuration for your repository.

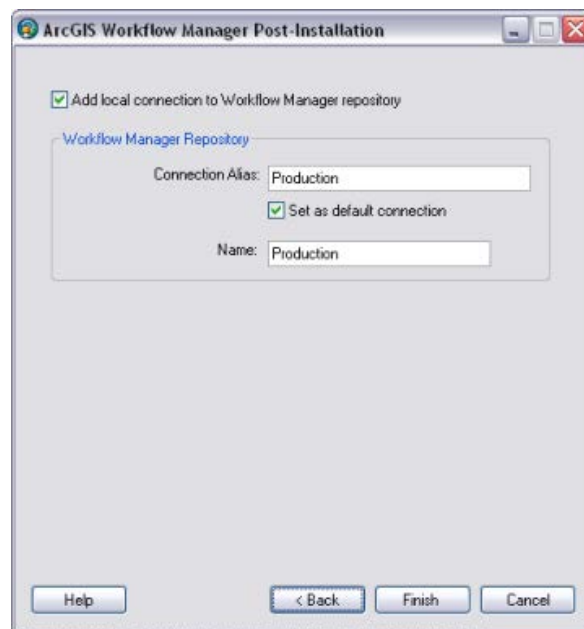
Note: If you plan to use all the functionality available in the Workflow Manager client application, it is recommended that you import at least the minimum configuration.

Note: Also, if you plan to import a Workflow Manager configuration exported from a 3.x database, choose the **No Configuration Import** option. Tools are provided in the Workflow Manager Configuration Manager to import the legacy configuration files.

- i. Click **Next**.
- j. Choose whether you would like to create a Workflow Manager database connection to this new repository when the postinstallation is complete.
 1. Specify a connection alias.
 2. Specify a repository name.

Note: This is used as a unique identifier when defining a cluster for Repository Replication.

3. Specify to set the database as default.



Decide if you want Workflow Manager to add a Workflow Manager database connection for you.

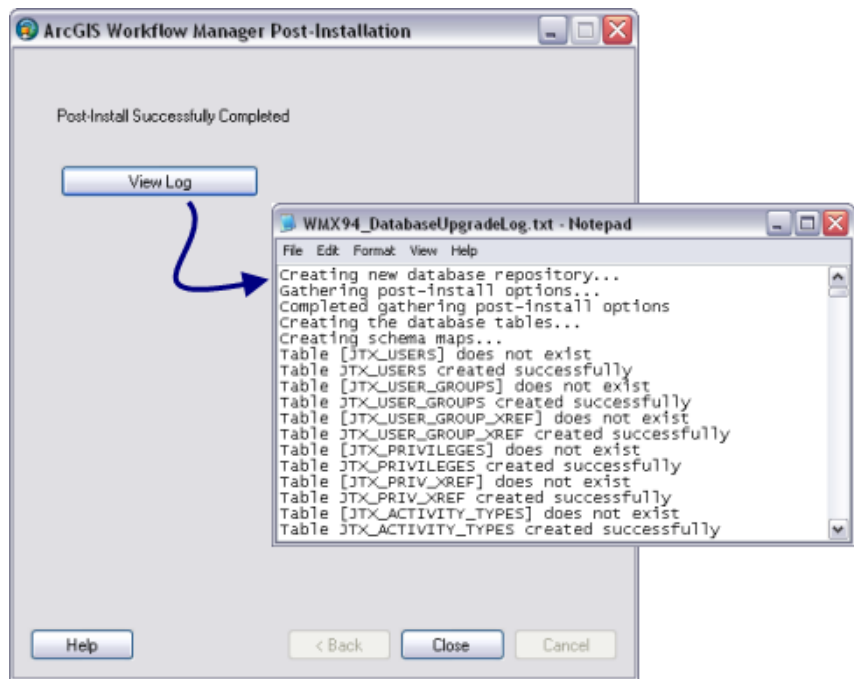
- k. Click **Finish**.

The system tables needed to use Workflow Manager are created in the database. Also, the current user running the postinstallation is automatically added to the list of users and has administrator access.

The final page in the wizard appears, indicating whether the postinstallation was successful.

Note: Administrator access means that the current user can connect to the selected database through the configuration manager to make edits to components of your system.

- l. Choose whether you would like to view the log.



The final page of the wizard indicates the success of the post installation and provides an option to see a log of the process.

m. Click **Close**.

Step 7: Verify Storage

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

```
sqlplus wmx/wmx@wmxddb

--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." Search for this document at <http://support.oracle.com>.

Step 8: Grant Permissions and Roles

Grant permissions to the Workflow Manager system tables through Oracle roles using the script below:

```
SET SERVEROUTPUT ON;

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

Then, simply grant the RLWMXEDITOR role to ArcSDE editor users and the RLWMXVIEWER role to ArcSDE viewer users.

Step 9: 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 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\WMXDB\SDE\sdelogfile01.dbf' SIZE 10M
AUTOEXTEND ON NEXT 1M MAXSIZE 100M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 512K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';

CREATE SMALLFILE TABLESPACE sdelogfileidx
DATAFILE 'D:\oracle\ORADATA\WMXDB\SDE\sdelogfileidx01.dbf' SIZE 10M
AUTOEXTEND ON NEXT 1M MAXSIZE 100M
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:wmxdb
```

- 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 import -f dbtune_logfile.sde -u sde -p sde
-i sde:oracle11g:wmxdb
```

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:
[resources.arcgis.com/en/help/main/10.2/index.html#
/What_are_ArcSDE_log_file_tables/002n00000013000000/](http://resources.arcgis.com/en/help/main/10.2/index.html#/What_are_ArcSDE_log_file_tables/002n00000013000000/)

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 10: Create Database Users

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

Editor User

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

Viewer User

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

Replication

For performance reasons, ArcGIS Workflow Manager repository replication is designed to allow organizations to maintain multiple Workflow Manager repositories locally in different geographic regions while the users at each location continue to work on the same set of jobs.

You will define a collection of Workflow Manager repositories that you want to have identical Workflow Manager contents. This collection of repositories is known as a repository cluster.

One Workflow Manager repository is designated as the parent repository and is tasked with coordinating the synchronization between each repository. After the parent repository has been specified, each child repository is added. The child repositories can be designated as either connected repositories, where the synchronization will happen automatically, or offline repositories, where the synchronization occurs by creating files that can be manually imported.

Learn more about replicating Workflow Manager repositories:

resources.arcgis.com/en/help/main/10.2/index.html#/Replicating_Workflow_Manager_Repositories/003800000077000000/

J10042

Conclusion

You can reduce disk contention and improve database I/O by storing the ArcGIS Workflow Manager 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/10.2/index.html#/Minimize_disk_I_O_contention_in_Oracle/002n00000025000000/

For more information on ArcGIS Workflow Manager, visit esri.com/workflowmanager or e-mail workflowmanager@esri.com.

Access blogs, forums, downloads, and more, from the ArcGIS Workflow Manager resource center at resources.arcgis.com/en/communities/workflow-manager/.

You can access other resources at ArcGIS 10.2 for Desktop Help (resources.arcgis.com/en/help/main/10.2/index.html#/Welcome_to_the_ArcGIS_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