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

Esri, 380 New York St., Redlands, CA 92373-8100 USA
TEL 909-793-2853 • FAX 909-793-5953 • E-MAIL info@esri.com • WEB esri.com

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

An Esri White Paper

Contents **Page**

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

Introduction

This white paper is intended for database administrators to help them establish the ArcGIS® Workflow Manager workspace in an enterprise geodatabase for SQL Server®. The enterprise geodatabase uses ArcSDE® technology as the gateway between geographic information system (GIS) clients and SQL Server. 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 of job actions is automatically recorded for each job to give managers a full 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 you in the creation and management of versions. An 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.

What Is the Postinstallation?

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, please 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.

DBTUNE

DBTUNE storage parameters let you control how ArcSDE technology creates objects within a SQL Server database. You can determine, for example, how to allocate space on a table or index, which FileGroup a table or index is created in, and other SQL Server-specific storage attributes. They also allow you to specify one of the available storage formats for the geometry of a spatial column.

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

If a large number of database connections are accessing the same files in the same location on disk, database performance will be slow 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 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 position FileGroup data files based on their usage pattern. For a multiversioned, highly active editing geodatabase, database files of the VERSIONS FileGroup may be separated and dispersed among available disks to avoid I/O contention.

Disk Configuration

Large production enterprise geodatabase systems should employ a hardware striping solution. Your best disk and data organization strategies involve spreading your data among multiple disks.

With data spread among 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: FileGroups and redundant array of independent disks (RAID). You can also combine the two by creating FileGroups 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.

The suggested SQL Server optimal configuration is as follows:

- DISK 0—SQL Server/Application software DISK 0
- DISK 1—master, model, msdb
- DISK 2—tempdb
- DISK 3—Log files
- DISK 4—Feature data tables
- DISK 5—Spatial index data tables
- DISK 6—Attribute data/Business tables
- DISK 7—SQL Server indexes

Reduce Disk I/O Contention

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

To configure data and log files for best performance, follow these best practices:

- To avoid disk contention, do not put data files on the same drive that contains the operating system files.

- Put transaction log files on a drive separate from the drive with data files. This gives you the best performance by reducing disk contention between data and transaction log files.

- Put the tempdb database on a separate drive if possible—preferably on a RAID 10 or RAID 5 system. With environments where there is intensive use of tempdb

databases, you can get better performance by putting tempdb on a separate drive, which lets SQL Server perform tempdb operations in parallel with database operations.

- The RAID configuration that is best for your database files depends on several factors, including performance and recoverability needs. RAID 10 is the recommended RAID system for transaction log, data, and index files. If you have budget restrictions, you can consider keeping transaction log files in a RAID 10 system and storing data and index files in a RAID 5 system.

  For more information about RAID, see RAID Levels and SQL Server at http://msdn.microsoft.com/en-us/ms190764.aspx and *Microsoft® Windows® 2000 Server Administrator's Companion* (Microsoft Press), chapter 7, "Planning Fault Tolerance and Avoidance," by Charlie Russel and Sharon Crawford, at http://technet.microsoft.com/en-us/library/bb742464.aspx.

- Use partitioning on large tables. Partitioning lets you split a table across multiple filegroups; by using partitions, you can place a subset of a table or index on a designated FileGroup. This capability lets you separate specific pieces of a table or index onto individual FileGroups and effectively manage file I/O for volatile tables. Partitions let you easily manage archival routines and data loading operations.

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

File Type	Database Activity	Move File to Disk With
Transaction log files	Frequent edits	Relatively low I/O
Transaction log files	Few or no edits	Moderate I/O
tempdb	Frequent edits	Low I/O, but separate from transaction log files
master, model, msdb	Few edits	Moderate I/O
Data	Frequent edits	Relatively low I/O

Creating the ArcSDE Instance

Step 1: Create Data Files

Create new FileGroups to store the Workflow Manager system tables.

```
FILEGROUP     SDE_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
```

J10043

```
USE MASTER
GO
ALTER DATABASE [WMX] ADD FILEGROUP [WMX_BDATA]
GO
ALTER DATABASE [WMX] ADD FILE(NAME = N'wmx_Bdata01', FILENAME =
N'C:\mssql\data\wmx\wmx_Bdata01.NDF' , SIZE = 1, MAXSIZE = 800, FILEGROWTH = 1) TO FILEGROUP
[WMX_BDATA]
GO
ALTER DATABASE [WMX] ADD FILEGROUP [WMX_BINDEX]
GO
ALTER DATABASE [WMX] ADD FILE(NAME = N'wmx_Bindex01', FILENAME =
N'C:\mssql\data\wmx\wmx_Bindex01.NDF' , SIZE = 1, MAXSIZE = 800, FILEGROWTH = 1) TO FILEGROUP
[WMX_BINDEX]
GO
ALTER DATABASE [WMX] ADD FILEGROUP [WMX_FDATA]
GO
ALTER DATABASE [WMX] ADD FILE(NAME = N'wmx_Fdata01', FILENAME =
N'C:\mssql\data\wmx\wmx_Fdata01.NDF' , SIZE = 1, MAXSIZE = 800, FILEGROWTH = 1) TO FILEGROUP
[WMX_FDATA]
GO
ALTER DATABASE [WMX] ADD FILEGROUP [WMX_FINDEX]
GO
ALTER DATABASE [WMX] ADD FILE(NAME = N'wmx_Findex01', FILENAME =
N'C:\mssql\data\wmx\wmx_Findex01.NDF' , SIZE = 1, MAXSIZE = 800, FILEGROWTH = 1) TO FILEGROUP
[WMX_FINDEX]
GO
ALTER DATABASE [WMX] ADD FILEGROUP [WMX_SDATA]
GO
ALTER DATABASE [WMX] ADD FILE(NAME = N'wmx_Sdata01', FILENAME =
N'C:\mssql\data\wmx\wmx_Sdata01.NDF' , SIZE = 1, MAXSIZE = 800, FILEGROWTH = 1) TO FILEGROUP
[WMX_SDATA]
GO
ALTER DATABASE [WMX] ADD FILEGROUP [WMX_SINDEX]
GO
ALTER DATABASE [WMX] ADD FILE(NAME = N'wmx_Sindex01', FILENAME =
N'C:\mssql\data\wmx\wmx_Sindex01.NDF' , SIZE = 1, MAXSIZE = 800, FILEGROWTH = 1) TO FILEGROUP
[WMX_SINDEX]
GO
ALTER DATABASE [WMX] ADD FILEGROUP [WMX_ADATA]
GO
ALTER DATABASE [WMX] ADD FILE(NAME = N'wmx__Adata01', FILENAME =
N'C:\mssql\data\wmx\wmx_Adata01.NDF' , SIZE = 1, MAXSIZE = 800, FILEGROWTH = 1) TO FILEGROUP
[WMX_ADATA]
GO
ALTER DATABASE [WMX] ADD FILEGROUP [WMX_AINDEX]
GO
ALTER DATABASE [WMX] ADD FILE(NAME = N'wmx__Aindex01', FILENAME =
N'C:\mssql\data\wmx\wmx_Aindex01.NDF' , SIZE = 1, MAXSIZE = 800, FILEGROWTH = 1) TO FILEGROUP
[WMX_AINDEX]
GO
ALTER DATABASE [WMX] ADD FILEGROUP [WMX_DDATA]
GO
ALTER DATABASE [WMX] ADD FILE(NAME = N'wmx__Ddata01', FILENAME =
N'C:\mssql\data\wmx\wmx_Ddata01.NDF' , SIZE = 1, MAXSIZE = 800, FILEGROWTH = 1) TO FILEGROUP
[WMX_DDATA]
GO
ALTER DATABASE [WMX] ADD FILEGROUP [WMX_DINDEX]
GO
ALTER DATABASE [WMX] ADD FILE(NAME = N'wmx__Dindex01', FILENAME =
N'C:\mssql\data\wmx\wmx_Dindex01.NDF' , SIZE = 1, MAXSIZE = 800, FILEGROWTH = 1) TO FILEGROUP
[WMX_DINDEX]
GO
```

Verify FileGroups and data files:

```
EXEC sp_helpdb wmx
GO
```

Step 2: Create a Workflow Manager User

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

Create user and schema:

```
USE WMX
GO
CREATE USER [wmx] FOR LOGIN [wmx]
GO
CREATE SCHEMA [wmx] AUTHORIZATION [wmx]
GO
ALTER USER [wmx] WITH DEFAULT_SCHEMA=[wmx]
GO
```

Grant privileges:

```
USE WMX
GO
EXEC sp_droprolemember 'gis_data_creator', 'wmx'
GO
EXEC sp_droprole 'gis_data_creator'
GO
CREATE ROLE gis_data_creator AUTHORIZATION dbo
GO
GRANT CREATE TABLE TO gis_data_creator
GO
GRANT CREATE PROCEDURE TO gis_data_creator
GO
GRANT CREATE VIEW TO gis_data_creator
GO
EXEC sp_addrolemember 'gis_data_creator', 'wmx'
GO
```

Verify role:

```
EXEC sp_helprolemember 'gis_data_creator'
GO
```

Verify role permissions:

```
select dp.NAME AS principal_name,
 dp.type_desc AS principal_type_desc,
 o.NAME AS object_name,
 p.permission_name,
 p.state_desc AS permission_state_desc
 from sys.database_permissions p
 left OUTER JOIN sys.all_objects o
 on p.major_id = o.OBJECT_ID
 inner JOIN sys.database_principals dp
 on p.grantee_principal_id = dp.principal_id
 where dp.NAME = 'gis_data_creator'
GO
```

Verify user permissions:

```
select USER_NAME(p.grantee_principal_id) AS principal_name,
 dp.type_desc AS principal_type_desc,
 p.class_desc,
 OBJECT_NAME(p.major_id) AS object_name,
 p.permission_name,
 p.state_desc AS permission_state_desc
from sys.database_permissions p
inner JOIN sys.database_principals dp
on p.grantee_principal_id = dp.principal_id
where USER_NAME(p.grantee_principal_id) = 'wmx'
```

Associate Login wmx with User wmx:

```
USE WMX
GO
EXEC sp_change_users_login 'update_one','wmx','wmx'
GO
EXEC sp_helpuser 'wmx'
GO
```

Step 3: Modify DBTUNE

Export the dbtune file before making any modification.

```
sdedbtune -o export -f dbtune_exp.sde -u sde -p sde -i 5151 –D wmx
```

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

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 FileGroup name for the appropriate parameters.

```
dbtune_wmx.sde

##WMX
A_INDEX_RASTER "WITH FILLFACTOR = 75 ON WMX_AINDEX"
A_INDEX_ROWID  "WITH FILLFACTOR = 75 ON WMX_AINDEX"
A_INDEX_SHAPE  "WITH FILLFACTOR = 75 ON WMX_AINDEX"
A_INDEX_STATEID    "WITH FILLFACTOR = 75 ON WMX_AINDEX"
A_INDEX_USER "WITH FILLFACTOR = 75 ON WMX_AINDEX"
A_INDEX_XML  "WITH FILLFACTOR = 75 ON WMX_AINDEX"
A_STORAGE    "ON WMX_ADATA"
AUX_INDEX_COMPOSITE    "WITH FILLFACTOR = 75 ON GIS_RASTERIDX"
AUX_STORAGE  "ON GIS_RASTER"
B_INDEX_RASTER     "WITH FILLFACTOR = 75 ON WMX_BINDEX"
B_INDEX_ROWID  "WITH FILLFACTOR = 75 ON WMX_BINDEX"
B_INDEX_SHAPE  "WITH FILLFACTOR = 75 ON WMX_BINDEX"
B_INDEX_TO_DATE    "WITH FILLFACTOR = 75 ON WMX_BINDEX"
B_INDEX_USER "WITH FILLFACTOR = 75 ON WMX_BINDEX"
B_INDEX_XML  "WITH FILLFACTOR = 75 ON WMX_BINDEX"
B_STORAGE    "ON WMX_BDATA"
BLK_INDEX_COMPOSITE    "WITH FILLFACTOR = 75 ON GIS_RASTERIDX"
BLK_STORAGE  "ON GIS_RASTER"
BND_INDEX_COMPOSITE    "WITH FILLFACTOR = 75 ON GIS_RASTERIDX"
BND_INDEX_ID "WITH FILLFACTOR = 75 ON GIS_RASTERIDX"
BND_STORAGE  "ON GIS_RASTER"
D_INDEX_ALL    "WITH FILLFACTOR = 75 ON WMX_DINDEX"
D_INDEX_DELETED_AT     "WITH FILLFACTOR = 75 ON WMX_DINDEX"
D_STORAGE    "ON WMX_DDATA"
F_INDEX_AREA "WITH FILLFACTOR = 75 ON WMX_FINDEX"
F_INDEX_FID  "WITH FILLFACTOR = 75 ON WMX_FINDEX"
F_INDEX_LEN  "WITH FILLFACTOR = 75 ON WMX_FINDEX"
F_STORAGE    "ON WMX_FDATA"
GEOMETRY_STORAGE   "SDEBINARY"
GEOMTAB_PK   "WITH FILLFACTOR = 75 ON WMX_FINDEX"
RAS_INDEX_ID "WITH FILLFACTOR = 75 ON GIS_RASTERIDX"
RAS_STORAGE  "ON GIS_RASTER"
S_INDEX_ALL  "WITH FILLFACTOR = 75 ON WMX_SINDEX"
S_INDEX_SP_FID     "WITH FILLFACTOR = 75 ON WMX_SINDEX"
S_STORAGE    "ON WMX_SDATA"
END
```

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

```
sdedbtune -o import -f dbtune_wmx.sde -u sde -p sde -i 5151 –D wmx
```

Step 4: Create the Workspace

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

Steps

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

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

b. Enter the connection information for the designated workspace and test the connection information.

J10043

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 Area of Interest feature class.

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

Define the spatial reference for the Area of Interest feature class.

g. Click **Next**.

- Choose your starting configuration option for the Workflow Manager repository.
  - **No Configuration Import** puts the Workflow Manager system tables schema into your designated geodatabase but will not prepopulate any of the configuration elements.
  - **Minimum Configuration** imports just the basic elements that the Workflow Manager system requires.

    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.
  - **Quick Configuration** includes the minimum elements and predefined job types as an example.
  - **Custom Configuration** allows you to import a preexisting Workflow Manager configuration file.

J10043

Choose the starting configuration for your repository.

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

h. Click **Next**.

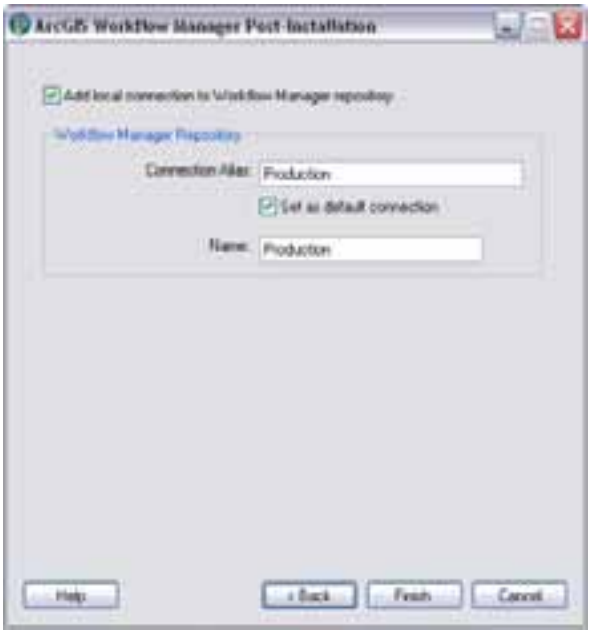
i. Choose whether you would like to create a Workflow Manager database connection in this new repository when the postinstallation is complete.

- Specify a connection alias.
- Specify a repository name.

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

- Specify to set the database as default.

J10043



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

j. Click **Finish**.

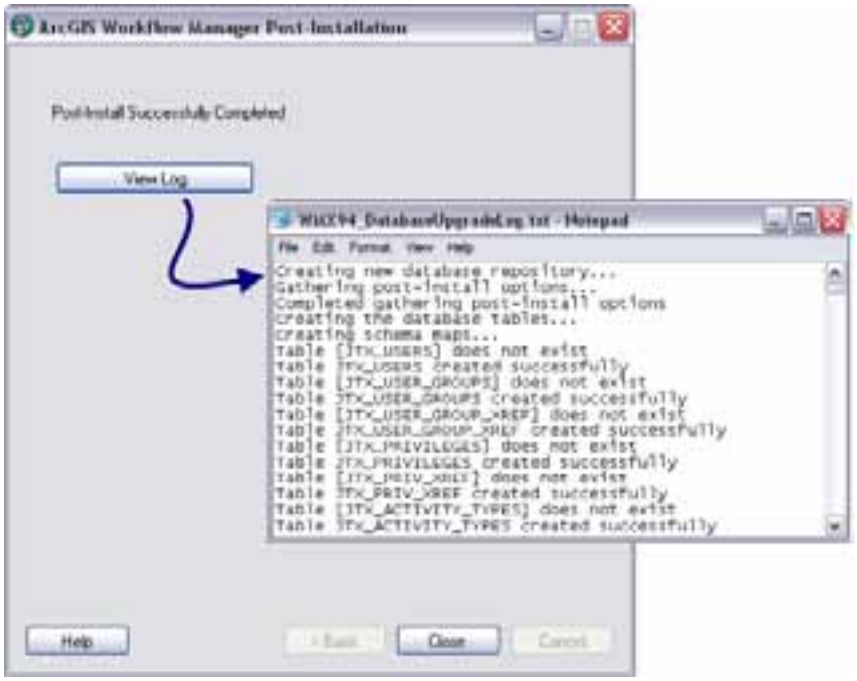
The system tables needed to use Workflow Manager are created in the database. Also, the current user running the postinstallation is automatically added to the list of users and has administrator access.

The final page in the wizard appears, indicating whether the postinstallation was successful or not.

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.

k. Choose whether you would like to view the log.

J10043



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

l. Click **Close**.

Step 5: Verify Storage

Run the SQL queries below to verify that the Workflow Manager workspace was created under the correct FileGroups.

```
USE WMX
GO
```

List FileGroups and data files:

```
EXEC sp_helpdb wmx
GO
```

List FileGroup data files:

```
EXEC sp_helpfilegroup 'PRIMARY'
GO
```

List tables by FileGroup:

```
SELECT USER_NAME(o.uid) [Owner],
 OBJECT_NAME(i.id) [Table Name],
 FILEGROUP_NAME(groupid) AS [Filegroup Name]
 FROM sysindexes i inner join sysobjects o
 ON i.id = o.id
 WHERE i.indid IN (0, 1) AND OBJECTPROPERTY(i.id, 'IsMSShipped') = 0 AND
 USER_NAME(o.uid) = 'wmx'
 ORDER BY 1,3,2
GO
```

List indexes by table and FileGroup:

```
select 'owner'=user_name(o.uid)
,'table_name'=object_name(i.id),i.indid
,'index_name'=i.name ,i.groupid
,'filegroup'=f.name ,'file_name'=d.physical_name
,'dataspace'=s.name from sys.sysindexes i
,sys.sysobjects o,sys.filegroups f
,sys.database_files d, sys.data_spaces s
where objectproperty(i.id,'IsUserTable') = 1
and i.id = o.id
and f.data_space_id = i.groupid
and f.data_space_id = d.data_space_id
and f.data_space_id = s.data_space_id
and user_name(o.uid) = 'wmx'
order by object_name(i.id),i.name,f.name
go
```

If any tables or indexes are stored in the wrong FileGroup, ALTER TABLE and ALTER INDEX can be used to change the FileGroup (see SQL Server Books Online at http://msdn.microsoft.com/en-us/library/ms130214.aspx).

Also, in Management Studio, you can re-create the DDL script of tables and indexes. Then, within *create script*, you can modify the FileGroup parameter and re-create the tables and indexes in the correct FileGroups. This is particularly useful when tables are empty and you are allowed to re-create database objects.

Step 6: Grant Permissions and Roles

Grant permissions to the Workflow Manager system tables through the schema.

```
USE [wmx]
GO
GRANT DELETE ON SCHEMA::[wmx] TO [wmxeditor]
GO
GRANT EXECUTE ON SCHEMA::[wmx] TO [wmxeditor]
GO
GRANT INSERT ON SCHEMA::[wmx] TO [wmxeditor]
GO
GRANT SELECT ON SCHEMA::[wmx] TO [wmxeditor]
GO
GRANT UPDATE ON SCHEMA::[wmx] TO [wmxeditor]
GO
```

Verify user permissions:

```
select USER_NAME(p.grantee_principal_id) AS principal_name,
 dp.type_desc AS principal_type_desc,
 p.class_desc,
 OBJECT_NAME(p.major_id) AS object_name,
 p.permission_name,
 p.state_desc AS permission_state_desc
from sys.database_permissions p
inner JOIN sys.database_principals dp
on p.grantee_principal_id = dp.principal_id
where USER_NAME(p.grantee_principal_id) = 'wmxeditor'
```

Step 7: Use Log File Tables

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

In ArcGIS, by default, log file tables are used if the selection set contains 100 or more records. This selection threshold of 100 features is set in the registry. It can be changed; however, Esri does not recommend doing so. There is no proven performance reason for changing it, and doing so could even cause performance problems. Thus, log file tables store feature selections in ArcMap™ that are greater than 100 for each connected SDE editor/viewer user.

Log file options are set using specific parameters in the SERVER_CONFIG and DBTUNE tables (sde_server_config and sde_dbtune in a SQL Server database). Parameters in these tables are altered using the sdeconfig and sdedbtune commands, respectively.

In SQL Server, one table is created in tempdb in the format ##SDE_session<sde_id>. This table is truncated when the connecting application deletes its log files, and the table is dropped when the session disconnects. When using the default setting, users do not require CREATE TABLE permission in the database for the session table to be created in tempdb.

The DBTUNE SESSION_TEMP_TABLE parameter must be set to 1 (true) to allow the session-based log file table to be created in tempdb. If you change the SESSION_TEMP_TABLE parameter to 0 (false), the SDE_LOGFILES, SDE_LOGFILE_DATA, and SDE_SESSION<SDE_ID> tables will be created in the connecting user's schema; hence, the user requires CREATE TABLE permission.

Learn more about ArcSDE log file tables at http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/Log_file_table_configuration_options_for_geodatabases_in_SQL_Server/002q00000012000000/.

Step 8: Create Database Users

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

Editor User

```
USE master
GO
EXEC sp_addlogin N'wmxeditor', 'wmx$editor', @logindb, @loginlang
GO
```

Create user:

```
USE [wmx]
GO
CREATE USER [wmxeditor] FOR LOGIN [wmxeditor]
GO
```

Grant privileges:

```
USE [wmx]
GO
GRANT DELETE ON SCHEMA::[wmx] TO [wmxeditor]
GO
GRANT EXECUTE ON SCHEMA::[wmx] TO [wmxeditor]
GO
GRANT INSERT ON SCHEMA::[wmx] TO [wmxeditor]
GO
GRANT SELECT ON SCHEMA::[wmx] TO [wmxeditor]
GO
GRANT UPDATE ON SCHEMA::[wmx] TO [wmxeditor]
GO
```

Viewer User

```
USE master
GO
EXEC sp_addlogin N'wmxviewer', 'wmx$viewer', @logindb, @loginlang
GO
```

Create user:

```
USE [wmx]
GO
CREATE USER [wmxviewer] FOR LOGIN [wmxviewer]
GO
```

Grant privileges:

```
USE [wmx]
GO
GRANT SELECT ON SCHEMA::[wmx] TO [wmxviewer]
GO
```

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 at help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/Minimize_disk_I_O_contention_in_SQL_Server/002q00000021000000/.

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/content/workflow-manager/10.0/about.

About Esri

Since 1969, Esri has been helping organizations map and model our world. Esri's GIS software tools and methodologies enable these organizations to effectively analyze and manage their geographic information and make better decisions. They are supported by our experienced and knowledgeable staff and extensive network of business partners and international distributors.

A full-service GIS company, Esri supports the implementation of GIS technology on desktops, servers, online services, and mobile devices. These GIS solutions are flexible, customizable, and easy to use.

Our Focus

Esri software is used by hundreds of thousands of organizations that apply GIS to solve problems and make our world a better place to live. We pay close attention to our users to ensure they have the best tools possible to accomplish their missions. A comprehensive suite of training options offered worldwide helps our users fully leverage their GIS applications.

Esri is a socially conscious business, actively supporting organizations involved in education, conservation, sustainable development, and humanitarian affairs.

Contact Esri

1 800 GIS XPRT (1 800 447 9778)
T 909 793 2853
F 909 793 5953
info@esri.com
esri.com

Offices worldwide
esri.com/locations

esri®
380 New York Street
Redlands, California 92373-8100 USA

Printed in USA