TRAINING GUIDE

ArcGIS Server and Geodatabase Administration for 10.2

Part 2

ArcGIS for Server v10.2 and Geodatabase Administration - Part 2

This session touches on key elements of maintaining enterprise geodatabases that help drive ArcGIS for Server services as well as backup techniques.

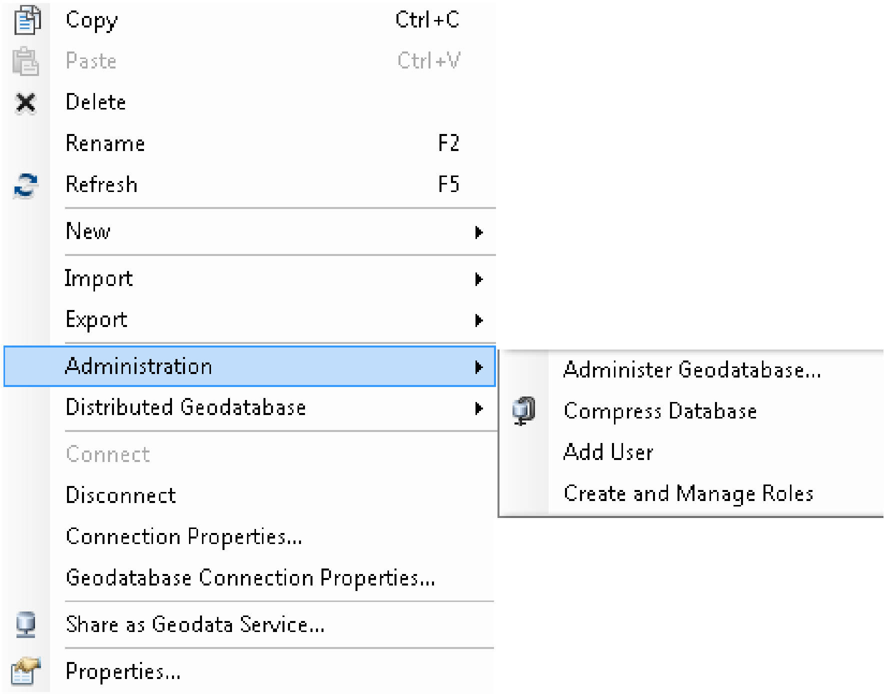
Table of Contents

Maintenance

Geodatabase Administration Tools

Starting in ArcGIS v10.1 and higher, the interface has changed so an admin user can administrate the Enterprise Geodatabase.

1. Go to ArcCatalog and connect to a Geodatabase as an administrator with Dbo privileges. Once connected, right click on the connection and choose administration.



2. In the administration menu, you will see four choices. Compress is now integrated within the administration menu. Also, you can add users and create/manage roles in your enterprise Geodatabase without touching SQL Server. Roles will allow you to give permissions to the role without having to give permissions to each and every user. Think of a role as a group.

Notes:____

____

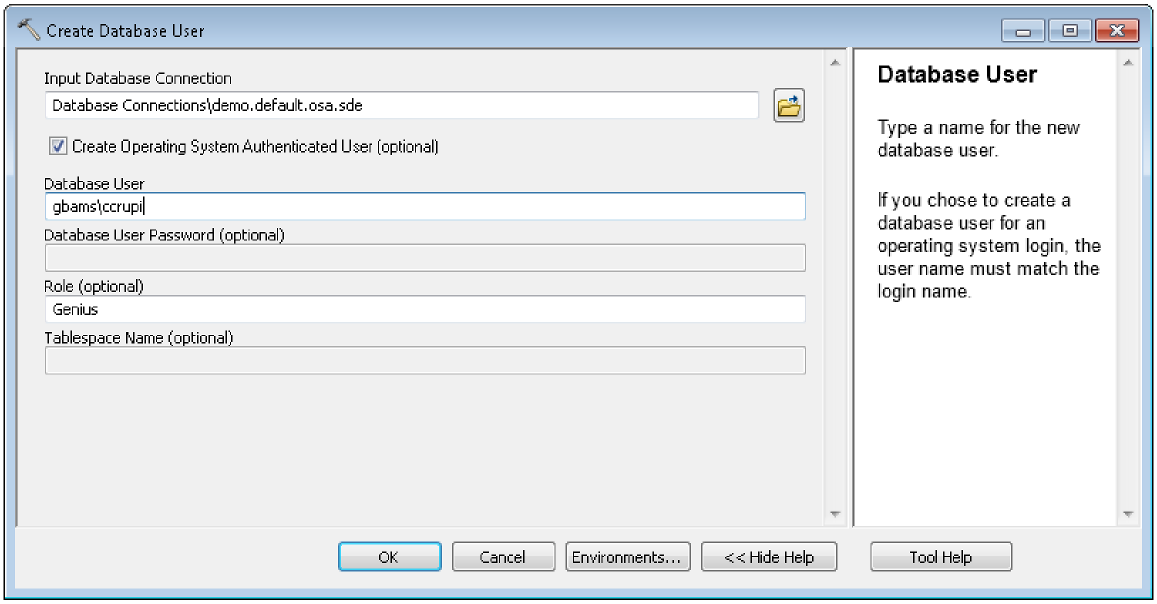
____

____

____

____

3. Click on Add User. You have two choices, operating system authentication or sql server authentication which gives an arbitrary login and password not related to windows but related to sql server only. You have the choice of adding the user to a role.



Input Database Connection: Database Connections\demo.default.osa.sde

☑ Create Operating System Authenticated User (optional)

Database User: gbams\ccrupi

Database User Password (optional):

Role (optional): Genius

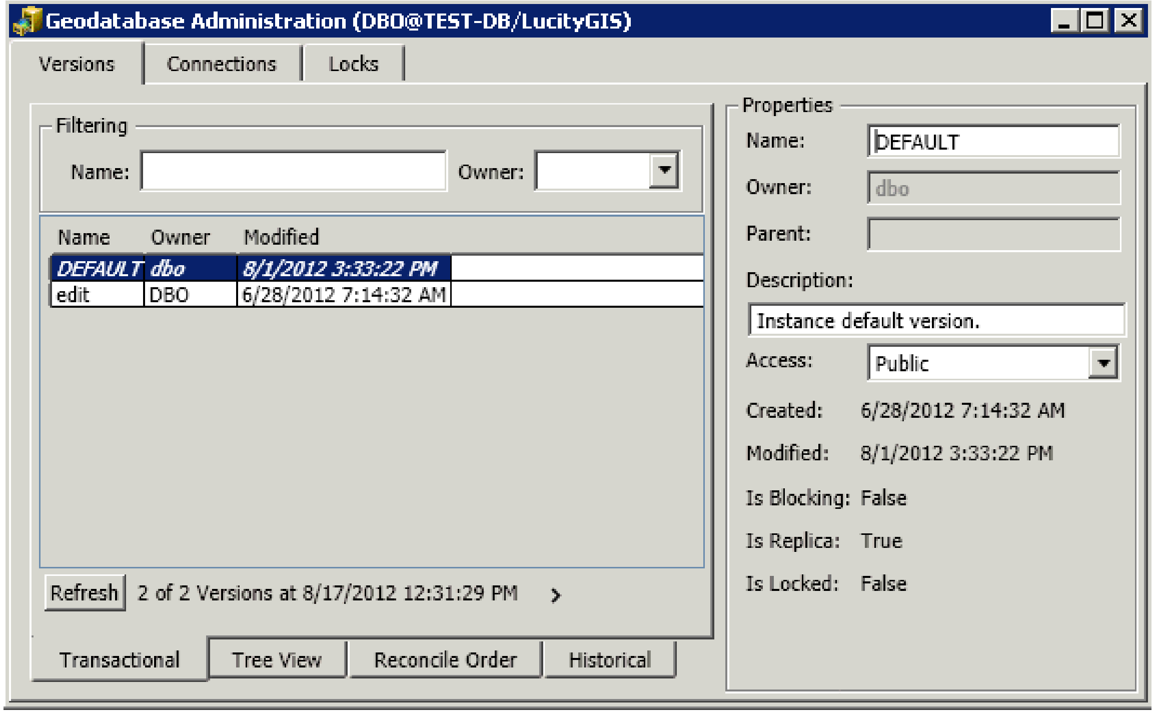
Tablespace Name (optional):

Database User

Type a name for the new database user.

If you chose to create a database user for an operating system login, the user name must match the login name.

4. Go to Administer Geodatabase on the administration menu, the Geodatabase Administration dialog will appear. You can create versions, check connections and locks.



Name:
Owner:

Name	Owner	Modified
DEFAULT	*dbo*	*8/1/2012 3:33:22 PM*
edit	DBO	6/28/2012 7:14:32 AM

2 of 2 Versions at 8/17/2012 12:31:29 PM

Properties

Name: DEFAULT

Owner: dbo

Parent:

Description: Instance default version.

Access: Public

Created: 6/28/2012 7:14:32 AM

Modified: 8/1/2012 3:33:22 PM

Is Blocking: False

Is Replica: True

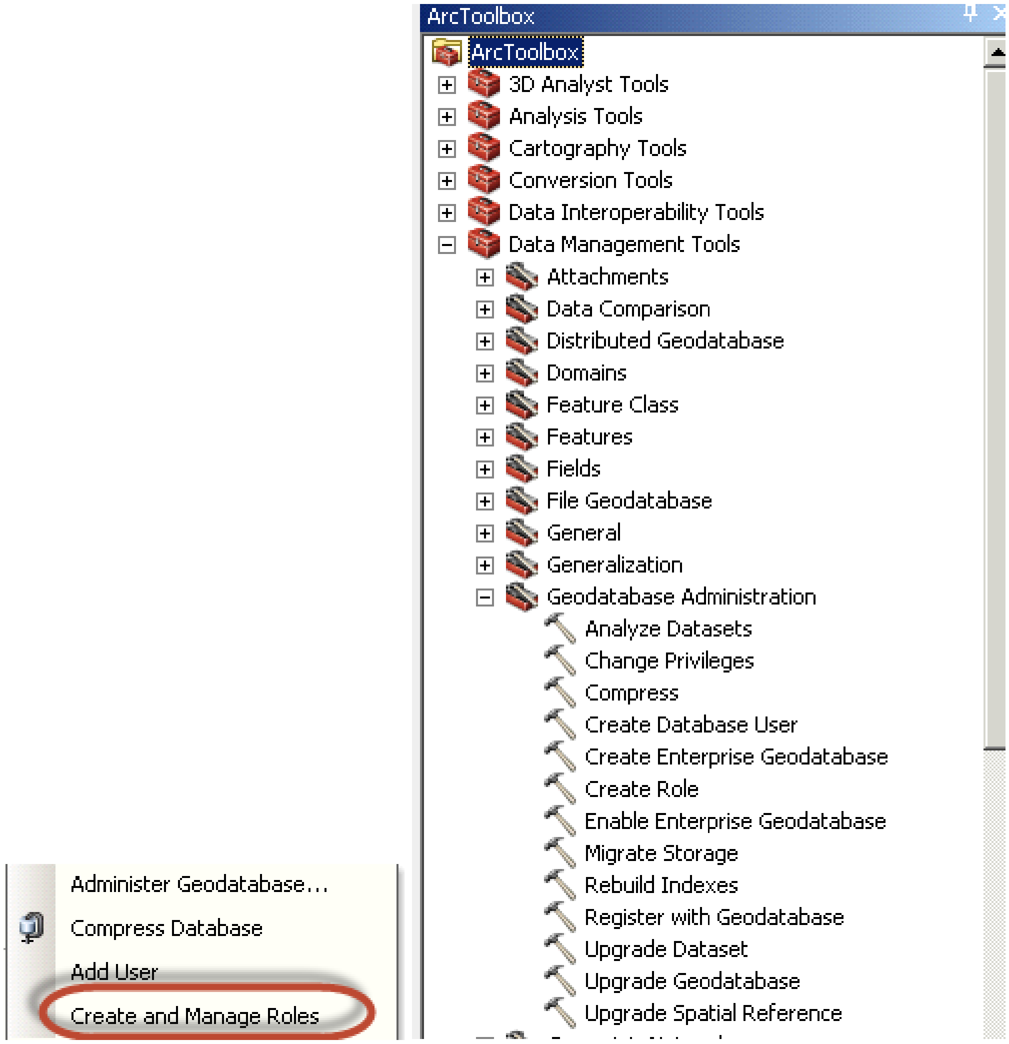
Is Locked: False

Create Role

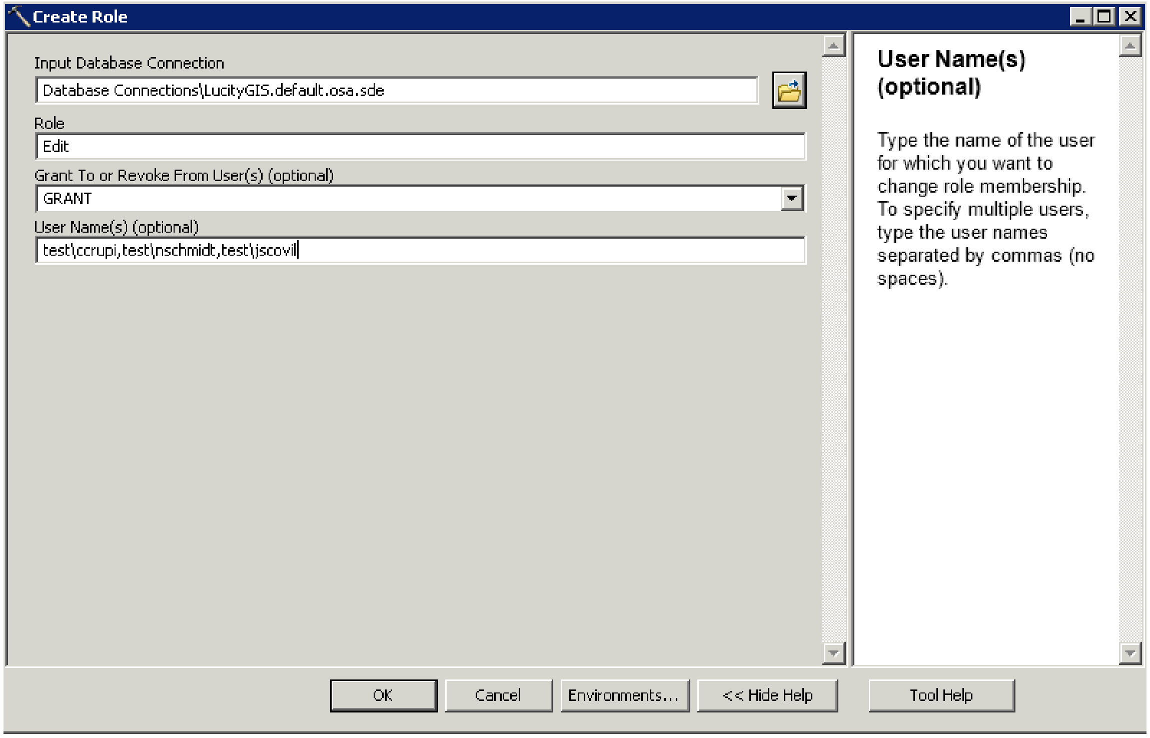
There are two different ways you can create roles. One way is to create an SDE role and assign users to the roles. The other way is to create Active Directory groups, add active directory groups to SQL Server allowing for Active Directory groups to act like roles. The great thing is that Active Directory is then managed by your IT user administration system without needing to manage all the users through SQL Server. We will show both ways below.

Create Role within ArcCatalog

1. Right click on the database connection and go to Create and Manage Roles. You can access this from ArcToolbox Data Management Tools as well.



2. The create role dialog will appear. You can grant or revoke users from roles from this interface.



Input Database Connection: Database Connections\LucityGIS.default.osa.sde

Role: Edit

Grant To or Revoke From User(s) (optional): GRANT

User Name(s) (optional): test\ccrupi,test\nschmidt,test\jscovil

User Name(s) (optional)

Type the name of the user for which you want to change role membership. To specify multiple users, type the user names separated by commas (no spaces).

3. Create Active Directory Groups for the Enterprise Geodatabase

Notes:____

____

____

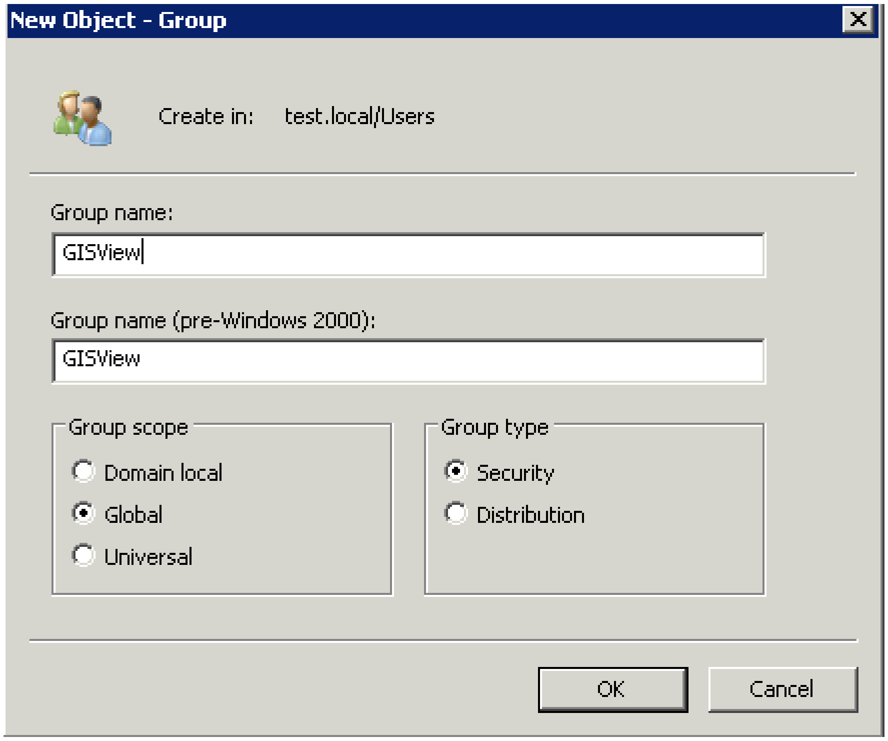
____

____

____

____

4. Open the Active Directory Users and Computers dialog from your domain controller and add a group by right clicking on users and select new group. Add a name such as GISView. The default settings are fine. Click ok.


New Object - Group

Create in: test.local/Users

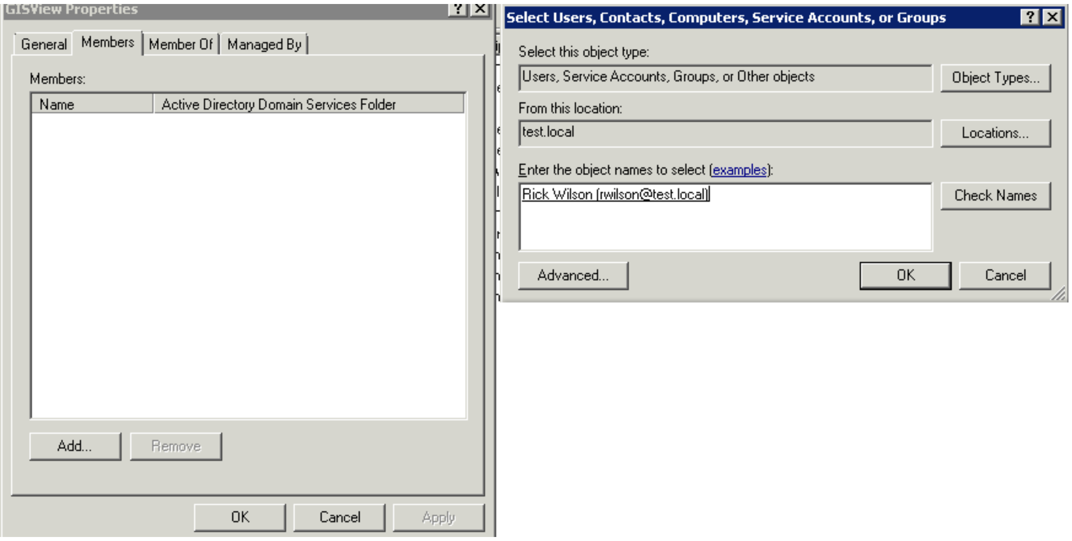
Group name: GISView

Group name (pre-Windows 2000): GISView

Group scope
○ Domain local
● Global
○ Universal

Group type
● Security
○ Distribution

5. Double click on the new group and add members within the members tab.


GISView Properties

Members:

Name	Active Directory Domain Services Folder

Select Users, Contacts, Computers, Service Accounts, or Groups

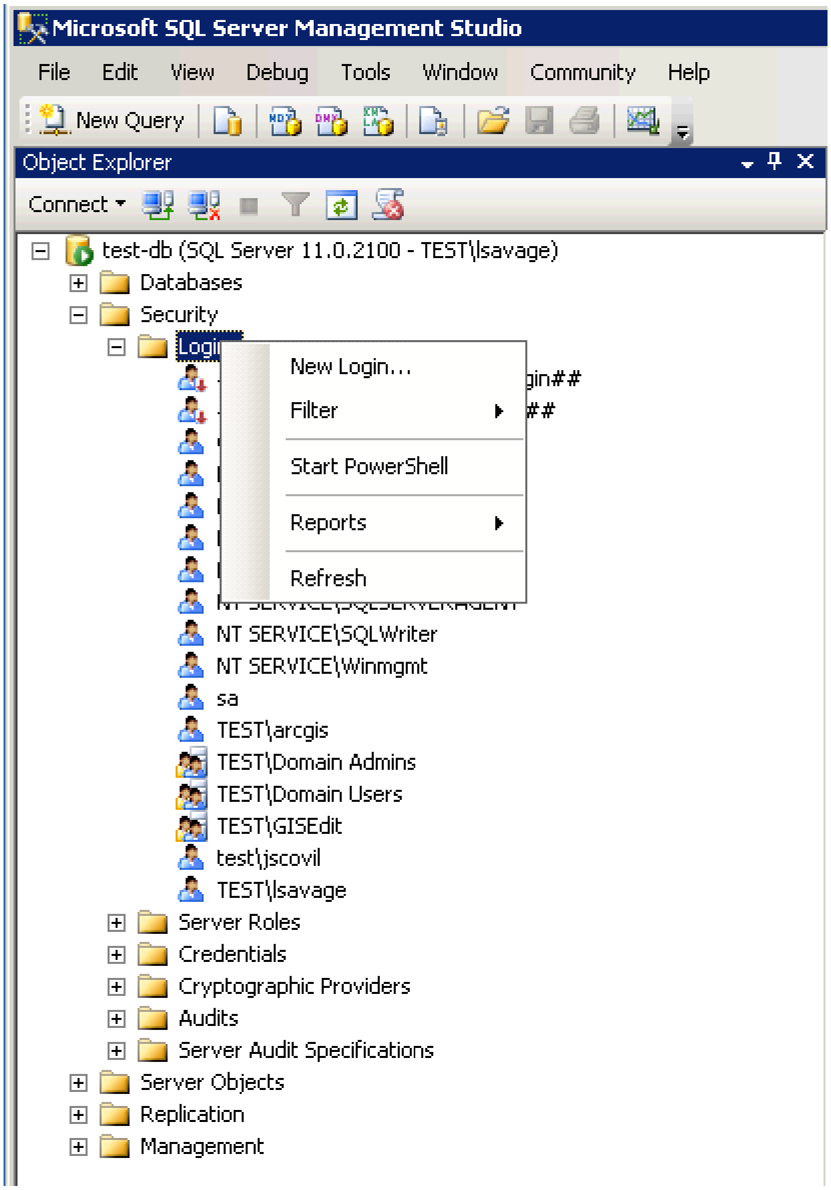
Select this object type: Users, Service Accounts, Groups, or Other objects

From this location: test.local

Enter the object names to select (examples): Rick Wilson (rwilson@test.local)

6. Add the member and click ok.

7. Open SQL Server Management Studio and add new group to SQL Server by going to the TOC of the instance connection and expand security. Right click on the Login folder and select new login.



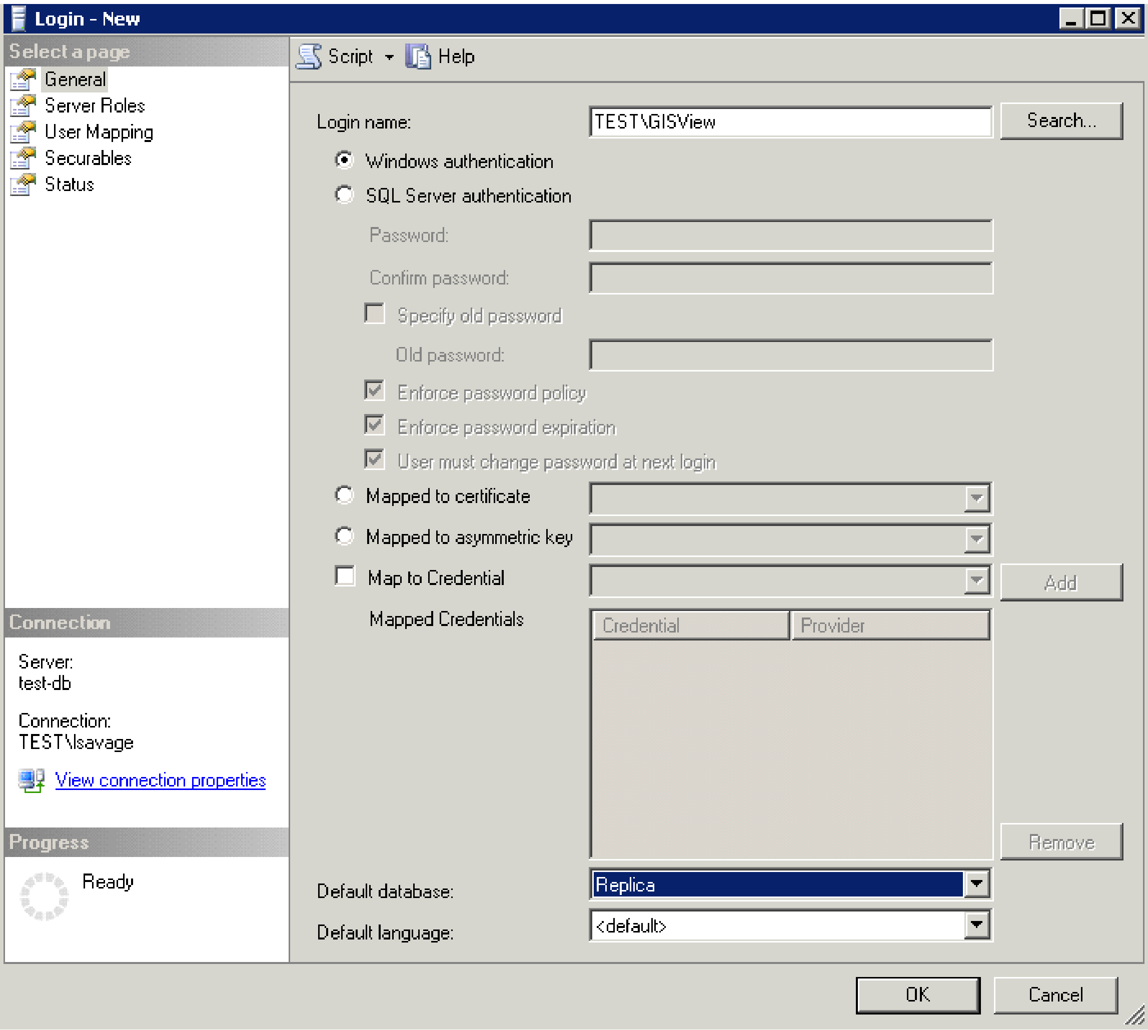
Notes:____

____

____

____

8. Click on Search and make sure you are connecting to your domain as the location and add groups as a selectable item from Object Types. Click ok when you've found the group. In this example, we're using test\GISView. Make sure the default database is selected to something other than master. Since these users are going to a view only group, we're assigning the group to the replica as the default database.



Login - New

Select a page: General, Server Roles, User Mapping, Securables, Status

Script | Help

Login name: TEST\GISView [Search...]

● Windows authentication

○ SQL Server authentication

Password:

Confirm password:

□ Specify old password

Old password:

☑ Enforce password policy

☑ Enforce password expiration

☑ User must change password at next login

○ Mapped to certificate

○ Mapped to asymmetric key

□ Map to Credential [Add]

Mapped Credentials

Credential	Provider

[Remove]

Default database: Replica

Default language: <default>

Connection

Server:
test-db

Connection:
TEST\lsavage

View connection properties

Progress

Ready

[OK] [Cancel]

Notes:____

____

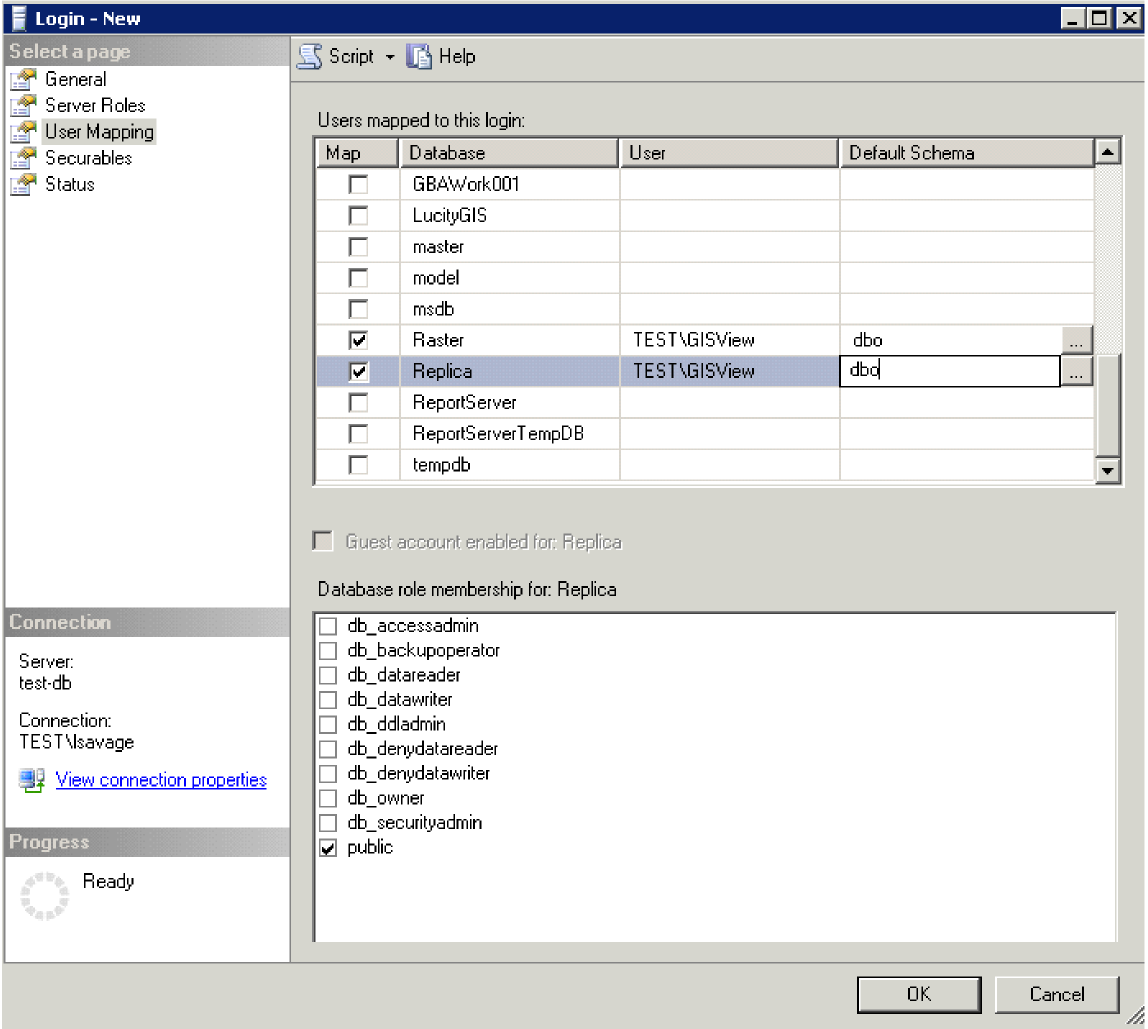
____

____

____

____

9. In the TOC of the Login – New dialog, click on user mapping. Check the boxes next to the databases you want the group to be assigned to. Just keep database role membership for <database> set to public. We'll let the Enterprise Geodatabase control privileges. Click ok.



Server:
test-db

Connection:
TEST\lsavage

Ready

Users mapped to this login:

Map	Database	User	Default Schema
□	GBAWork001		
□	LucityGIS		
□	master		
□	model		
□	msdb		
☑	Raster	TEST\GISView	dbo
☑	Replica	TEST\GISView	dbo
□	ReportServer		
□	ReportServerTempDB		
□	tempdb		

□ Guest account enabled for: Replica

Database role membership for: Replica

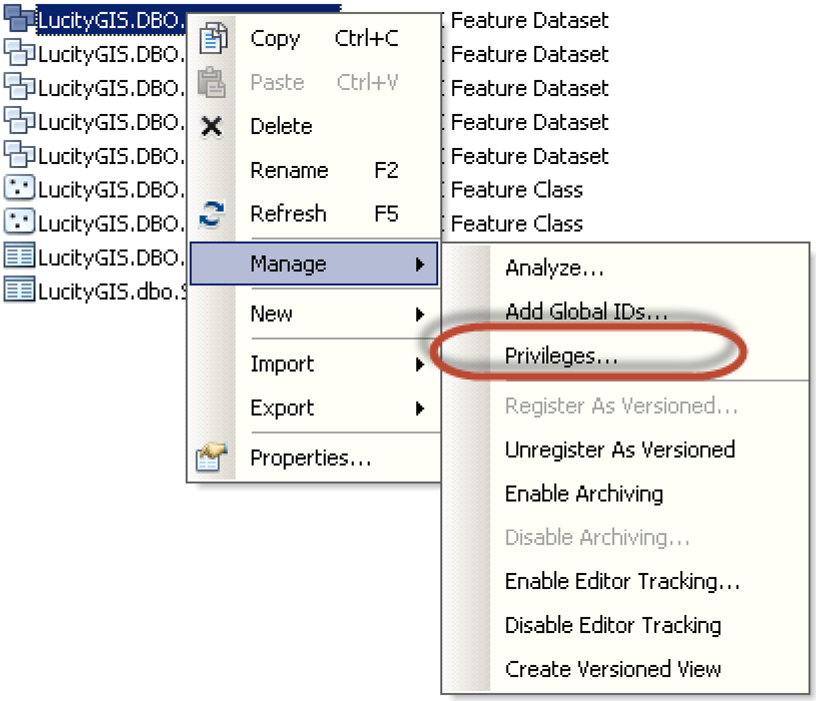
- □ db_accessadmin
- □ db_backupoperator
- □ db_datareader
- □ db_datawriter
- □ db_ddladmin
- □ db_denydatareader
- □ db_denydatawriter
- □ db_owner
- □ db_securityadmin
- ☑ public

Notes:____

Change/Add Privileges

In the previous exercise, we created two types of role scenarios. One is using the enterprise Geodatabase tools to create a role with multiple users which can be revoked per user. The other is allowing Active Directory Groups to be assigned like roles. Now, we are going to assign privileges to these two types of roles.

1. Go to ArcCatalog database connection as administrator and right click on the data you want to add privileges to. Remember, you can multiply select one object at a time (Feature Datasets, Feature Classes, and Tables). In this example, we'll be adding the active directory group we created. Select Privileges.



Notes:____

____

____

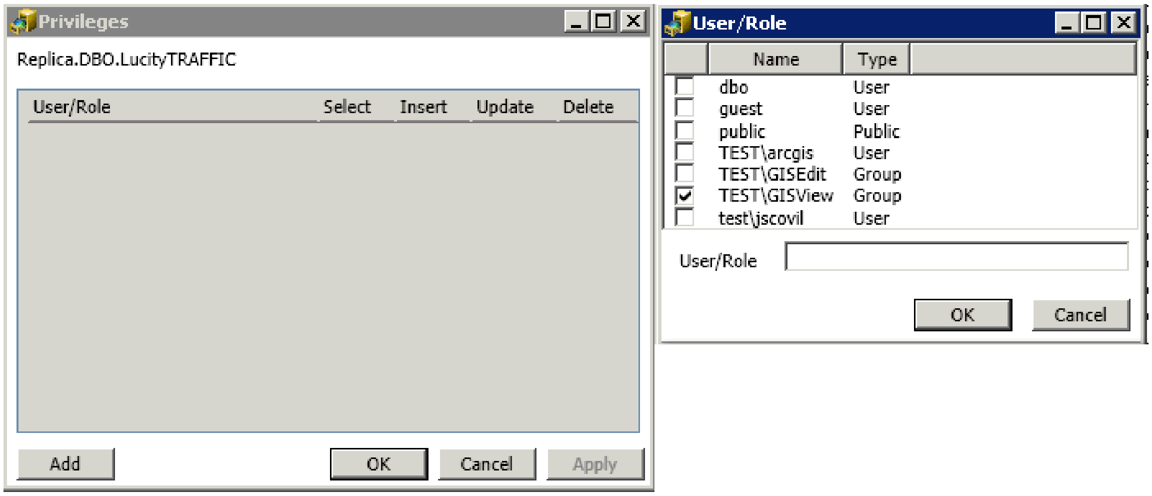
____

____

____

____

2. Click on the user/role you would like to add. Click ok.



Privileges

Replica.DBO.LucityTRAFFIC

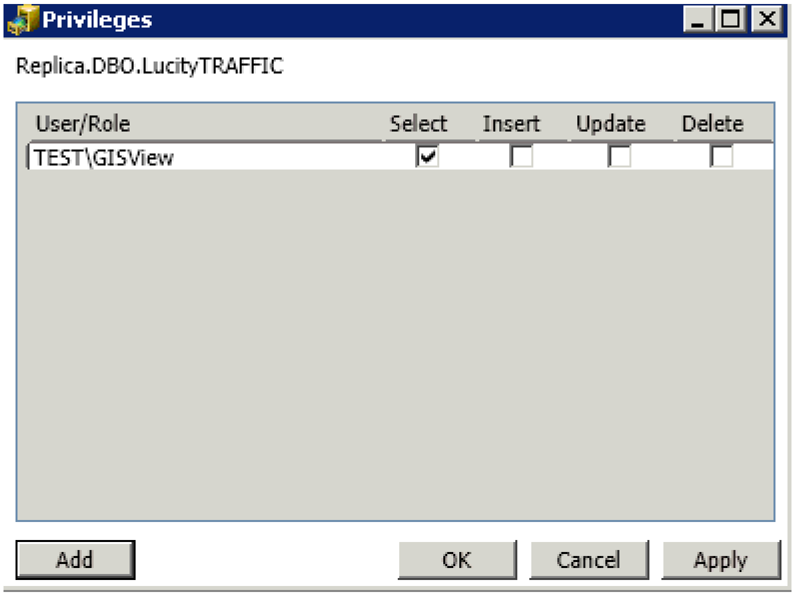
User/Role	Select	Insert	Update	Delete

User/Role

	Name	Type
□	dbo	User
□	guest	User
□	public	Public
□	TEST\arcgis	User
□	TEST\GISEdit	Group
☑	TEST\GISView	Group
□	test\jscovil	User

User/Role

3. Since the GIS View group is read-only, only keep the select box checked. Click ok.



Privileges

Replica.DBO.LucityTRAFFIC

User/Role	Select	Insert	Update	Delete
TEST\GISView	☑	□	□	□

Notes:____

____

____

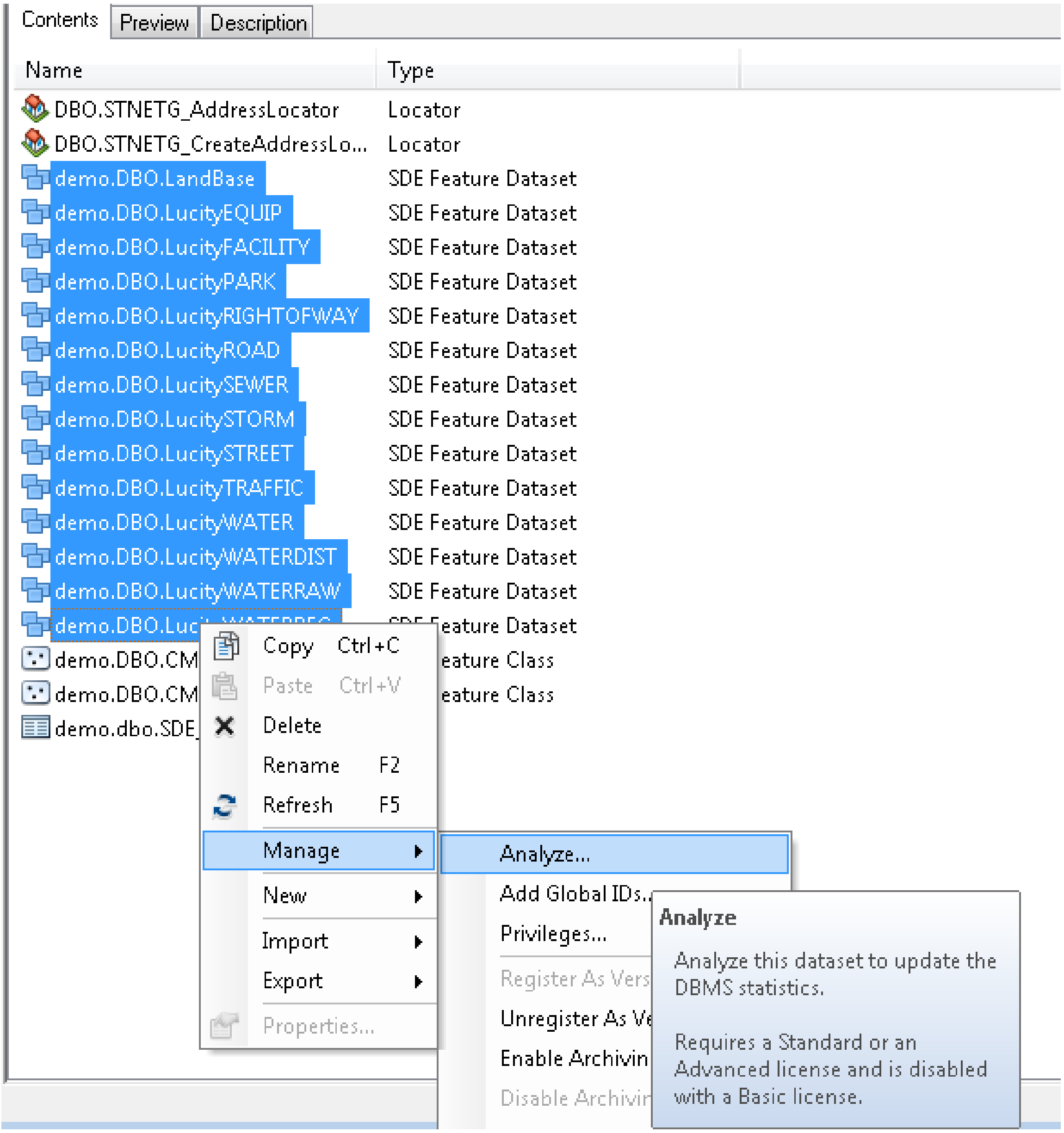
____

____

Analyze and Index

Analyze and index is something that you want to do on a regular basis if you creating and adding features to your database. This will allow for better faster access to your database. Make sure ArcGIS for Server services are stopped before rebuilding indexes.

- Go to your database connection and select on the objects you want to analyze and rebuilt index. Right click on the select items and go to manage/analyze. There is no feel good complete dialog given so let the wheel turn until complete.



Name	Type
DBO.STNETG_AddressLocator	Locator
DBO.STNETG_CreateAddressLo...	Locator
demo.DBO.LandBase	SDE Feature Dataset
demo.DBO.LucityEQUIP	SDE Feature Dataset
demo.DBO.LucityFACILITY	SDE Feature Dataset
demo.DBO.LucityPARK	SDE Feature Dataset
demo.DBO.LucityRIGHTOFWAY	SDE Feature Dataset
demo.DBO.LucityROAD	SDE Feature Dataset
demo.DBO.LucitySEWER	SDE Feature Dataset
demo.DBO.LucitySTORM	SDE Feature Dataset
demo.DBO.LucitySTREET	SDE Feature Dataset
demo.DBO.LucityTRAFFIC	SDE Feature Dataset
demo.DBO.LucityWATER	SDE Feature Dataset
demo.DBO.LucityWATERDIST	SDE Feature Dataset
demo.DBO.LucityWATERRAW	SDE Feature Dataset
demo.DBO.Luc	SDE Feature Dataset
demo.DBO.CM	eature Class
demo.DBO.CM	eature Class
demo.dbo.SDE_	

Analyze

Analyze this dataset to update the DBMS statistics.

Requires a Standard or an Advanced license and is disabled with a Basic license.

Compression

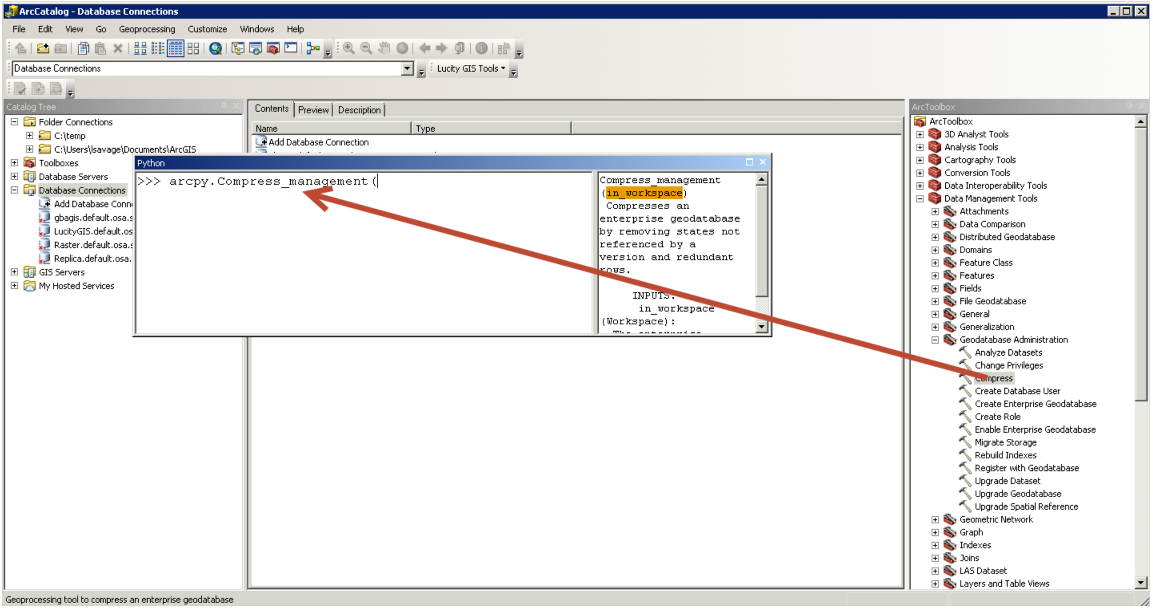
What is Compression?

Is a process of moving rows of data from the delta (add and delete) tables to the base tables of the feature classes. This is a versioning performance process.

- If your data is in default and you have the data versioned, you should compress.
- Performance and Data Integrity Tip: Analyze statistics and rebuilt indexes before and after compression
- Data Integrity Tip: Compress at least once a week
- Data Integrity Tip: Compress after large loads of data or major changes in data

There are several ways to compress your database. Below are the different ways to compress your database.

1. Using Python
   a. Open ArcToolbox and go to data management tools/Geodatabase Administration/Compress
   b. Open the Python dialog by going to the Geoprocessing Menu and left click on Python
   c. A dialog window will open for Python. Left click on the Compress tool in ArcToolbox and drag the tool within the Python window
   d. Notice that the Python dialog has been updated with the new function



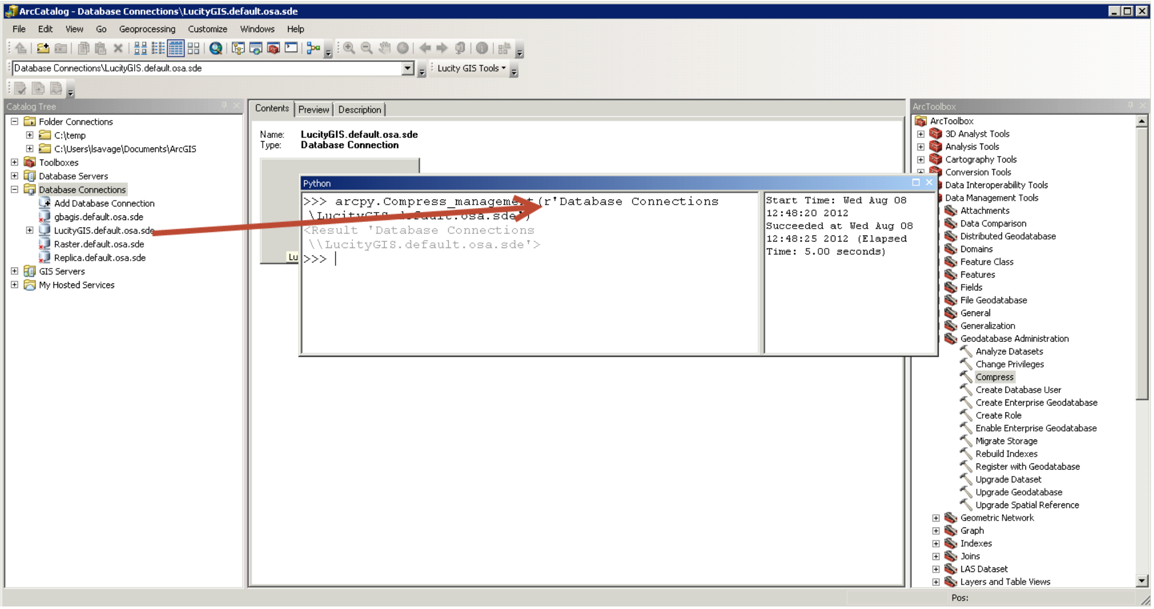
```
>>> arcpy.Compress_management(
```

```
Compress_management
(in_workspace)
 Compresses an
enterprise geodatabase
by removing states not
referenced by a
version and redundant
rows.

     INPUTS:
      in_workspace
(Workspace):
```

Geoprocessing tool to compress an enterprise geodatabase

   e. Now open the database connection you would like to compress. Make sure the connection has rights to perform a compress. Drag and drop the connection into the Python dialog

f. Hit the Enter button on the keyboard as this will execute the command. On the right, the dialog will show the start time and how long it took to complete the operation.



Name: LucityGIS.default.osa.sde
Type: Database Connection

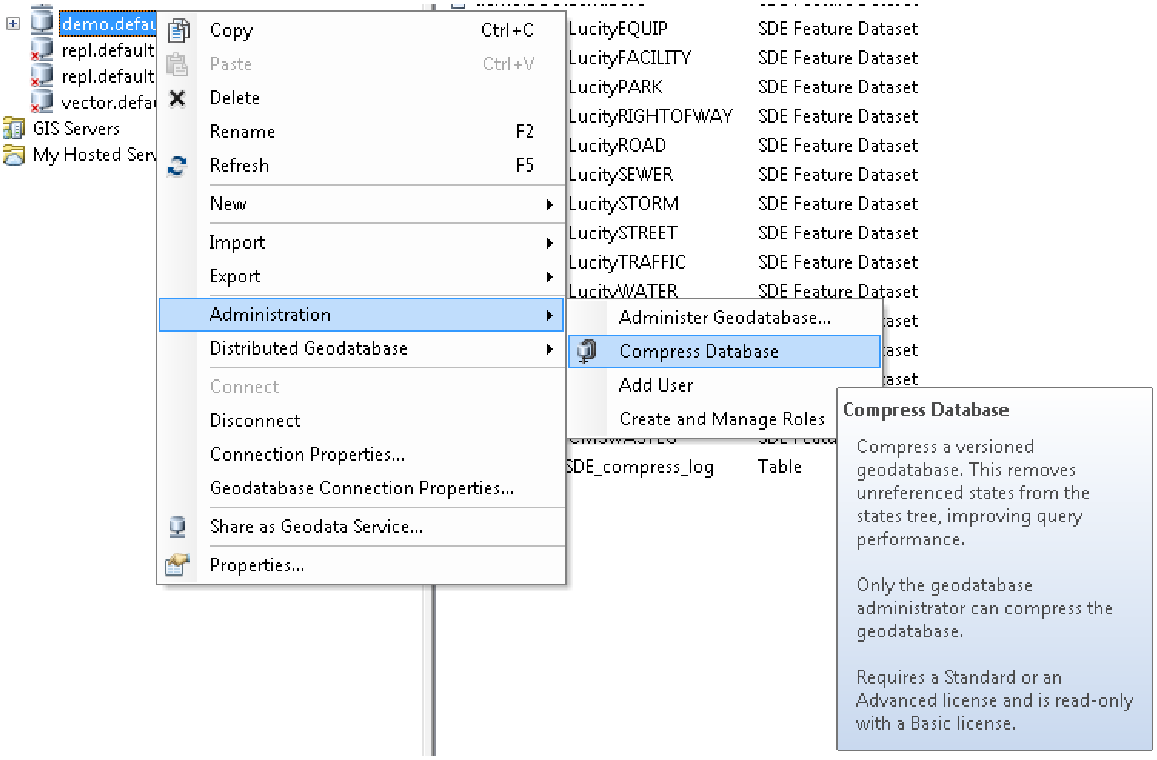
```
>>> arcpy.Compress_management(r'Database Connections
\LucityGIS.default.osa.sde')
<Result 'Database Connections
\\LucityGIS.default.osa.sde'>
>>> 
```

```
Start Time: Wed Aug 08
12:48:20 2012
Succeeded at Wed Aug 08
12:48:25 2012 (Elapsed
Time: 5.00 seconds)
```

2. Using the ArcCatalog Tool

a. Right Click on a database connection in the ArcCatalog TOC

b. Choose Administration/Compress Database


demo.defau
repl.default
repl.default
vector.defa
GIS Servers
My Hosted Serv
Copy Ctrl+C
Paste Ctrl+V
Delete
Rename F2
Refresh F5
New
Import
Export
Administration
Distributed Geodatabase
Connect
Disconnect
Connection Properties...
Geodatabase Connection Properties...
Share as Geodata Service...
Properties...
Administer Geodatabase...
Compress Database
Add User
Create and Manage Roles
LucityEQUIP SDE Feature Dataset
LucityFACILITY SDE Feature Dataset
LucityPARK SDE Feature Dataset
LucityRIGHTOFWAY SDE Feature Dataset
LucityROAD SDE Feature Dataset
LucitySEWER SDE Feature Dataset
LucitySTORM SDE Feature Dataset
LucitySTREET SDE Feature Dataset
LucityTRAFFIC SDE Feature Dataset
LucityWATER SDE Feature Dataset
SDE_compress_log Table

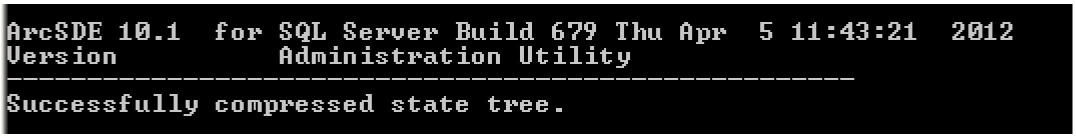
Compress Database

Compress a versioned geodatabase. This removes unreferenced states from the states tree, improving query performance.

Only the geodatabase administrator can compress the geodatabase.

Requires a Standard or an Advanced license and is read-only with a Basic license.

3. Command Line Compress; make sure whatever system connected has the command line installation of ArcSDE installed.
    a. Syntax: sdeversion -o compress -u <sa or admin> -p <password> -i sde:sqlserver:<instancename> -s <server> -D <database> -N



```
ArcSDE 10.1  for SQL Server Build 679 Thu Apr  5 11:43:21  2012
Version          Administration Utility
-----------------------------------------------------
Successfully compressed state tree.
```

Note: Compress process will only compress data that is in read-only mode. If a feature class is being edited, the compress process will bypass the feature class table and only compress non-edited data. If you are having problems with compression, please reconcile, post all versions. Afterwards, delete the versions and make sure all state locks are terminated as well as all users are disconnected. After all users are disconnected, perform the compression routine. This is better known as "Zero State Compression". Afterwards, recreate your versions and you'll have clean delta tables.

Notes:____

____

____

____

____

____

____

____

____

____

____

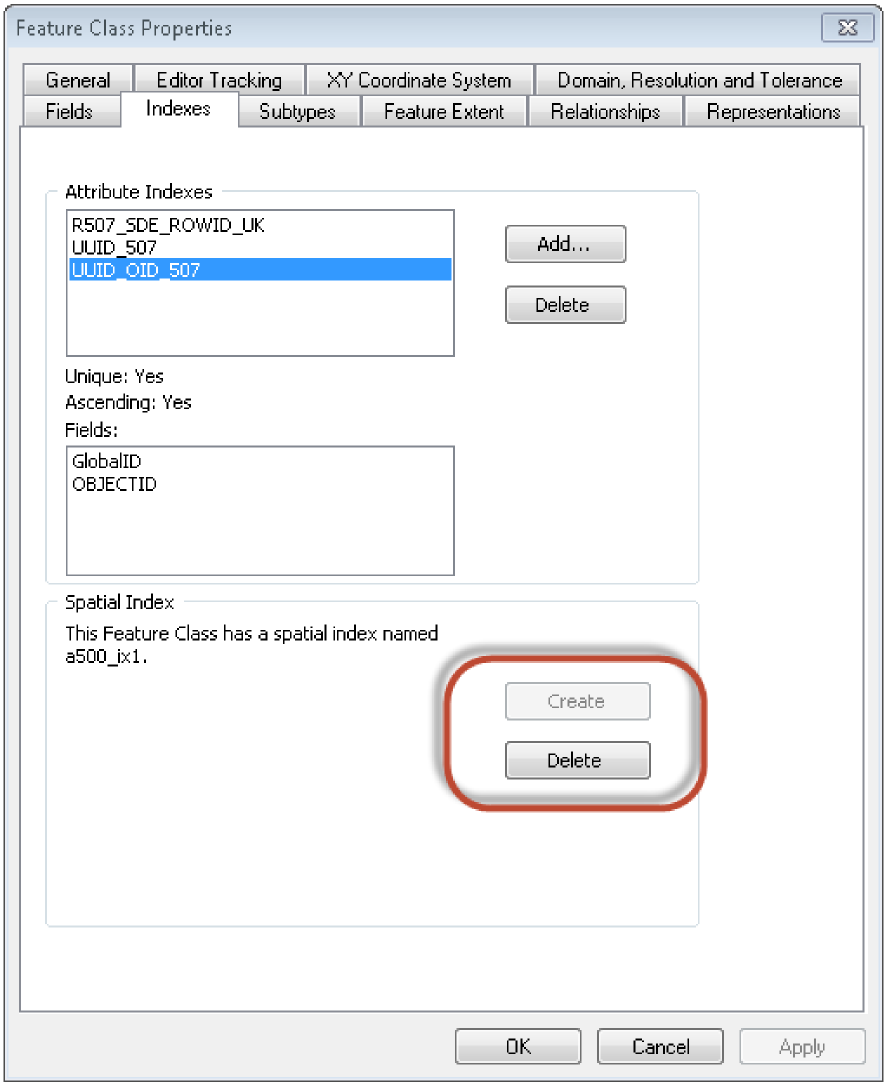
____

Spatial and Attribute Indexes

In ArcGIS v10.1 and higher, spatial indexes are managed for you in a sql table. You can delete and recreate the spatial index from the feature class properties. Spatial indexes allow your spatial data to be queried faster within a grid concept. Each grid has a size based on the density of your spatial data. In the past, this was a calculation that needed to be performed by the GIS DBA or GIS Analyst but is now automated for you. Kudos ESRI!

Spatial Indexes

1. If you are recalculating your feature extent and feel that you want to recreate your spatial index, right click on a feature class and choose properties.
2. When the feature class properties dialog appears, go to the indexes tab.
3. Click Delete and Create under Spatial Index at the bottom of the dialog. Click Apply.



Feature Class Properties

Attribute Indexes

R507_SDE_ROWID_UK
UUID_507
UUID_OID_507

Unique: Yes
Ascending: Yes
Fields:

GlobalID
OBJECTID

Spatial Index

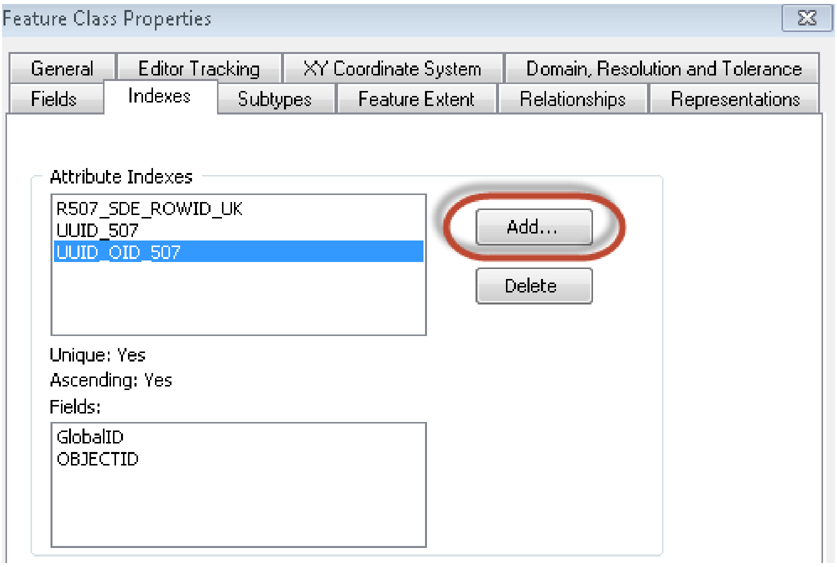
This Feature Class has a spatial index named a500_ix1.

4. This process will recreate the feature class spatial index table for the feature class.

Attribute Indexes

Attribute indexes are used to speed the performance of data queries used on a consistent basis. As an example, we are going to setup an attribute index for Lucity GIS data. Both the common ID and the Lucity AutoID will speed up your query performance capability for each feature class that is linked to Lucity when assigning an attribute index.

1. In the same dialog as the previous exercise (Feature Class Properties), please go to the indexes tab.
2. Click on the Add button next to Attribute Indexes.



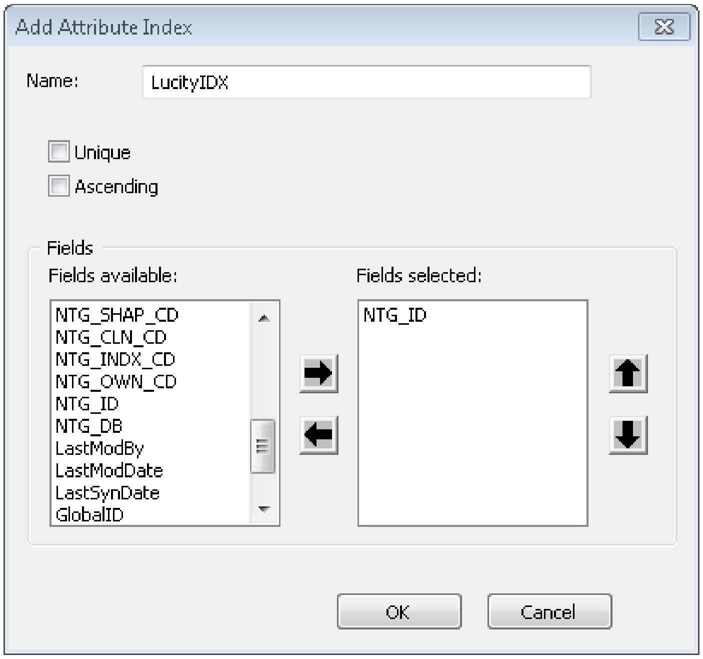
Attribute Indexes

R507_SDE_ROWID_UK
UUID_507
UUID_OID_507

Unique: Yes
Ascending: Yes
Fields:

GlobalID
OBJECTID

3. Create LucityIDX for the Name and add the NTG_ID which is linked to the Lucity AutoID.



Name: LucityIDX

☐ Unique
☐ Ascending

Fields

Fields available:

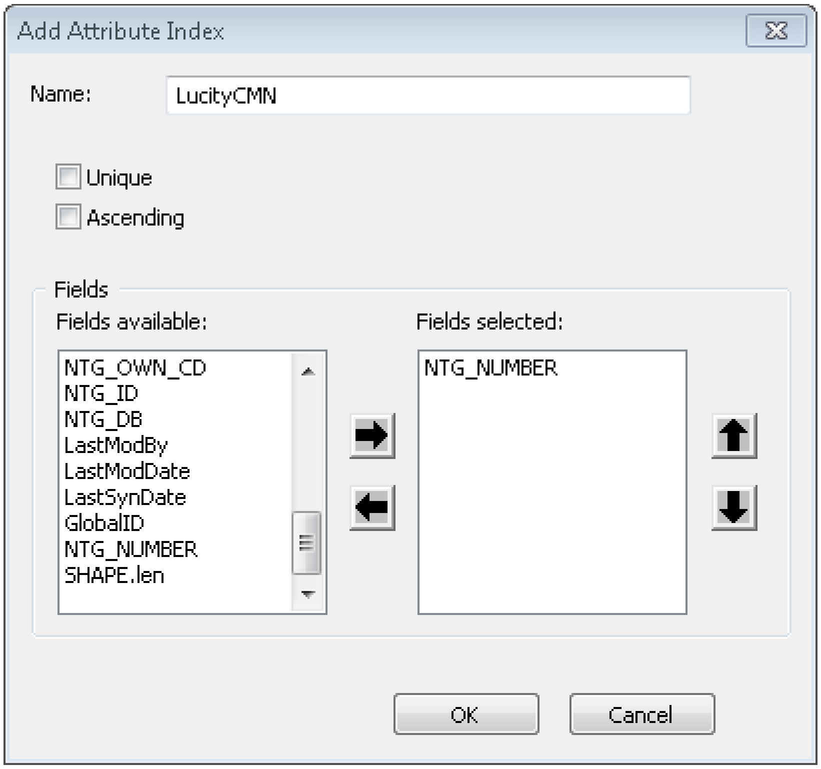
NTG_SHAP_CD
NTG_CLN_CD
NTG_INDX_CD
NTG_OWN_CD
NTG_ID
NTG_DB
LastModBy
LastModDate
LastSynDate
GlobalID

Fields selected:

NTG_ID

4. Click ok and Apply.

5. Complete the same operation again by adding LucityCMN index and associate it with the field NTG_NUMBER which is linked to the Lucity AltID.



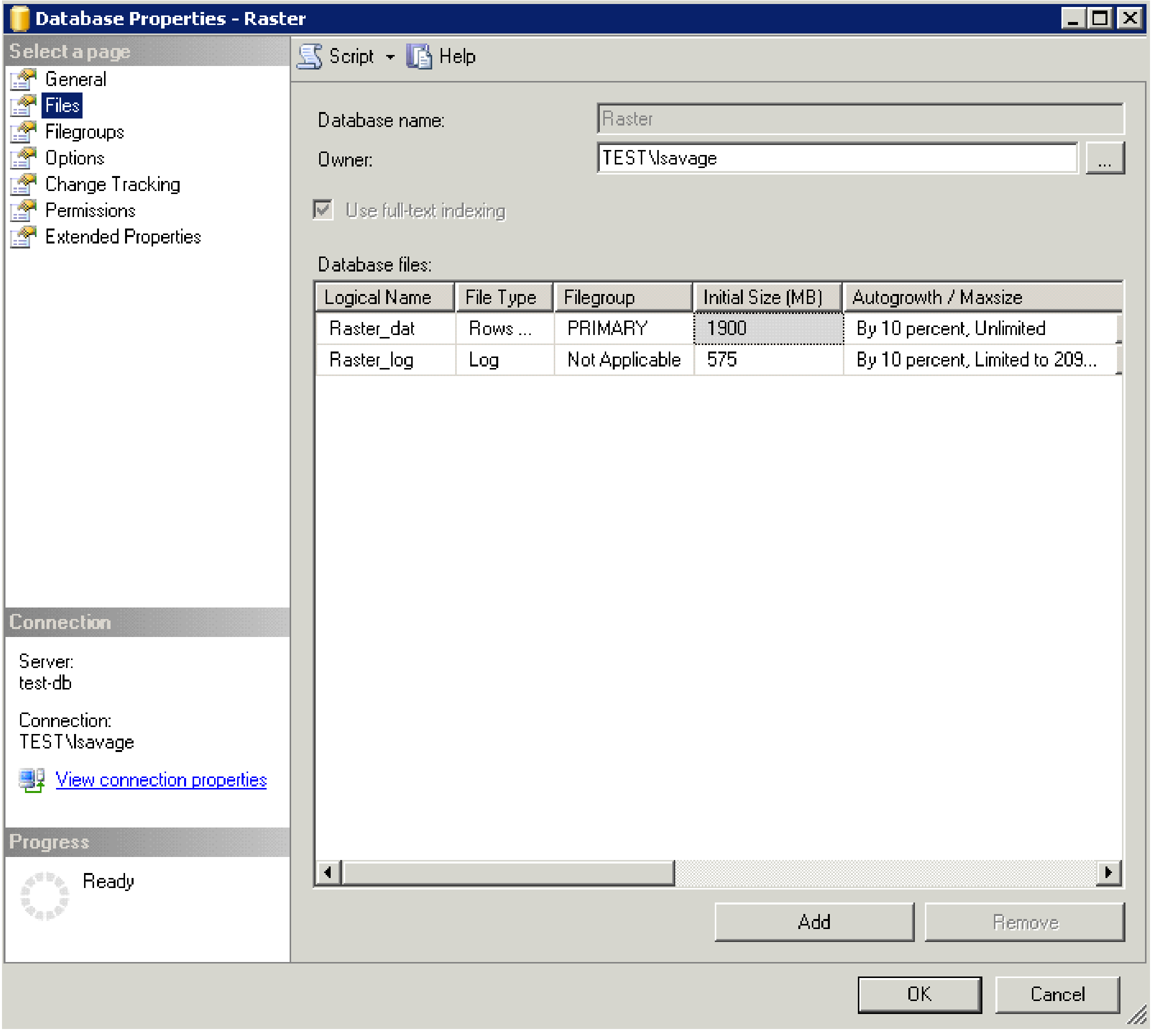
6. Click Ok and Apply.

Notes:____

Tune SQL Server space for Geodatabases

In terms of SQL Server space for Geodatabases, the functionality of the Geodatabase is key to how much space you will need. Recommend splitting up the vector and the raster data into two separate Geodatabases. Raster should be in its own database. This will allow for easy to manage backups and system configuration.

1. When loading Raster data calculate the decompressed size of the image and set this as the file size for the initial load. The logfile can be somewhat small as ESRI has changed the transaction of the loads from one long transaction to iterative raster load transactions. A good rule of thumb, create logfiles for Raster 1/3 of the size of the Raster database. For the data file size, find out the decompression size of the complete mosaic raster data or single image size as a collection. You can choose to use the new mosaic option for raster as this will create a virtual mosaic dataset like the terrain datasets but that's totally up to you as the administrator.



Database name: Raster

Owner: TEST\lsavage

☑ Use full-text indexing

Database files:

Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth / Maxsize
Raster_dat	Rows ...	PRIMARY	1900	By 10 percent, Unlimited
Raster_log	Log	Not Applicable	575	By 10 percent, Limited to 209...

Connection

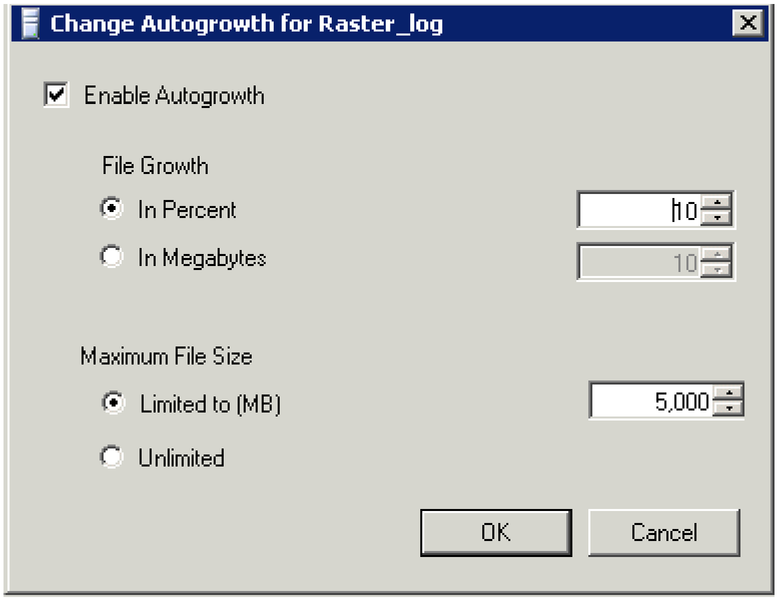
Server:
test-db

Connection:
TEST\lsavage

Progress

Ready

2. For log files, don't let the log files use unlimited file growth as a looped transaction could bring your server down.



Change Autogrowth for Raster_log

☑ Enable Autogrowth

File Growth

● In Percent 10

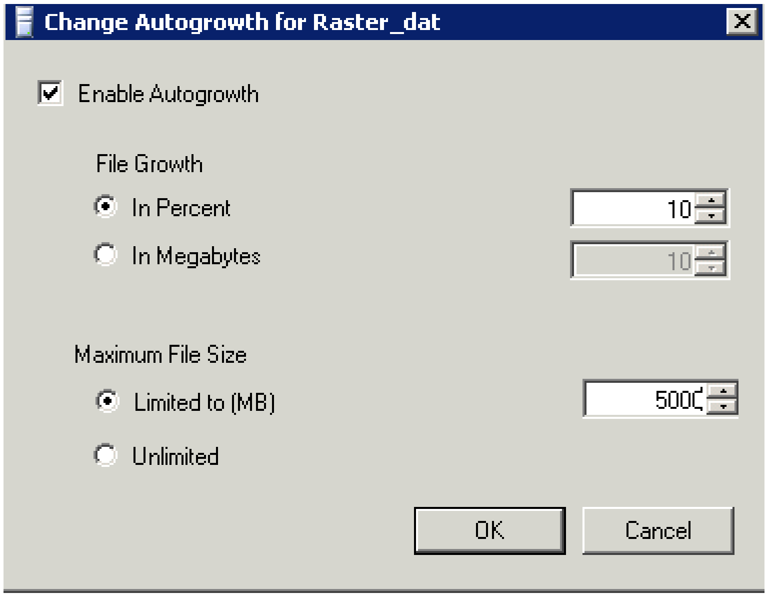
○ In Megabytes 10

Maximum File Size

● Limited to (MB) 5,000

○ Unlimited

3. Do the same for the data filegroup.



Change Autogrowth for Raster_dat

☑ Enable Autogrowth

File Growth

● In Percent 10

○ In Megabytes 10

Maximum File Size

● Limited to (MB) 5000

○ Unlimited

4. Most Raster datasets are 30 GB to 70 GB once loaded in the Enterprise Geodatabase. You may want to compensate for the increase. It's faster to pre-allocate space for Raster before you load. Don't allow Raster databases to get too large. As a recommendation, you could split up Raster databases into separate databases maxing out each one into 300GB databases.

5. After loading, analyze through ArcCatalog and assign permissions. Make sure you backup, truncate log file and shrink the database before making it available to the public.

6. If large loads for vector data or a Geodatabase that is used as an editing Geodatabase, perform similar operations.

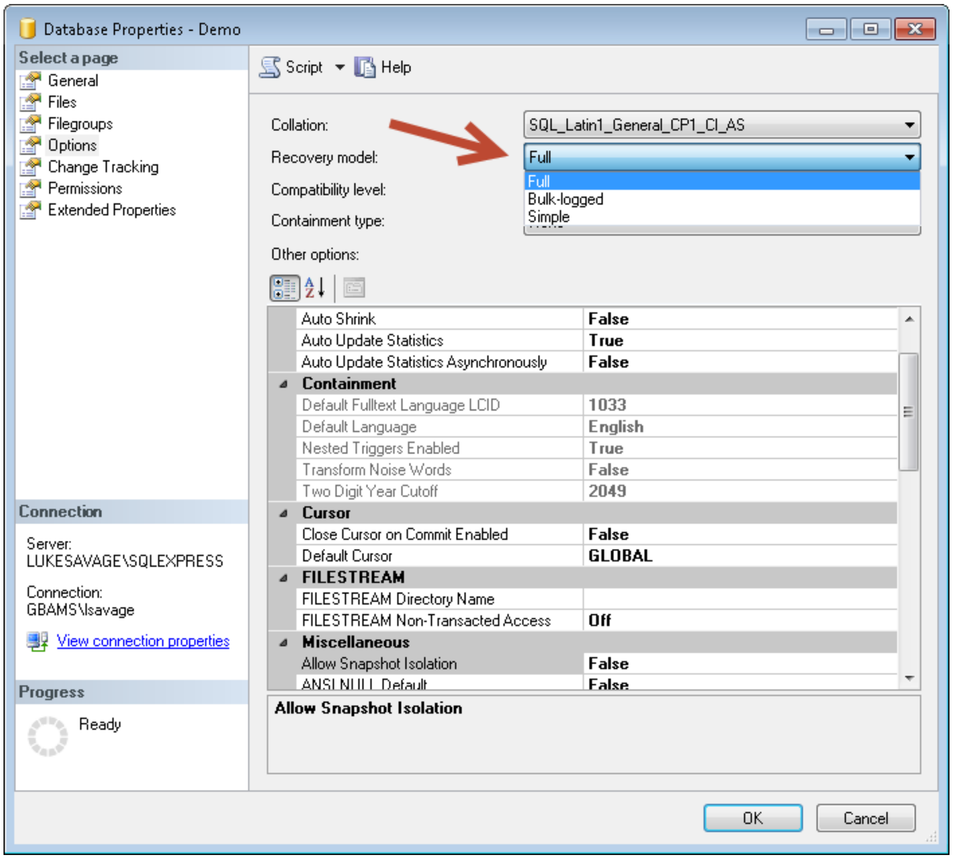
7. Set your growth to a manageable amount within your data storage. For Vector, you can use percentage. For Raster, you will be adding large amounts of data in one transaction. You should consider size growth rather than percentage.

Backups

If you want to create backups using the SQL Server maintenance wizard, this is an easy step by step process. However, in our example below, we will be touching on creating backups from SQL Script, using SQL Commands and windows task manager. Command line seems a bit quicker when running through each operation. Below is the information about each batch script. In order to run the complete procedure below, you will need to have each database backup recovery model setting turned to 'Full'. Simple model is good for a development environment but only is as good as the last backup. For cleanup, please use a file copy and delete batch routine to reset the backups as backups will not overwrite in sql script. You can use a simple copy and delete command in command line.

Note: when you want 99% assurance of your enterprise geodatabase backups, simply backup to a known location both offsite and onsite as a file geodatabase.

Enterprise Geodatabase Backups



Collation: SQL_Latin1_General_CP1_CI_AS

Recovery model: Full

Compatibility level:

Containment type:

Auto Shrink	False
Auto Update Statistics	True
Auto Update Statistics Asynchronously	False
Containment	
Default Fulltext Language LCID	1033
Default Language	English
Nested Triggers Enabled	True
Transform Noise Words	False
Two Digit Year Cutoff	2049
Cursor	
Close Cursor on Commit Enabled	False
Default Cursor	GLOBAL
FILESTREAM	
FILESTREAM Directory Name	
FILESTREAM Non-Transacted Access	Off
Miscellaneous	
Allow Snapshot Isolation	False
ANSI NULL Default	False

Allow Snapshot Isolation

Connection

Server:
LUKESAVAGE\SQLEXPRESS

Connection:
GBAMS\lsavage

View connection properties

Progress

Ready

Full Backup: This will backup the entire database.

Intermediate Backups: Captures changes to the database since the last full backup (also known as differential backups.

Truncate Logs: Truncates logs to free up space

Shrink: Shrinks data and log files.

Analyze and Index: Analyze/update statistics, and updates indexes

1. Create a full backup script. For every database, create a separate action.

```
BACKUP DATABASE [DEMO] TO  DISK = N'C:\backup\SQLEXPRESS\DEMO.bak' WITH NOFORMAT, NOINIT,
RETAINDAYS = 13,  NAME = N'DEMO-Full Database Backup', SKIP, NOREWIND, NOUNLOAD,  STATS = 10

GO
```

2. Create an intermediate or differential script.

```
BACKUP DATABASE [DEMO] TO  DISK = N'C:\backup\SQLEXPRESS\DEMO.bak' WITH DIFFERENTIAL,
NOFORMAT, NOINIT, RETAINDAYS = 13,  NAME = N'DEMO-Full Database Backup', SKIP, NOREWIND,
NOUNLOAD,  STATS = 10

GO
```

3. Create a backup Log script.

```
BACKUP LOG [DEMO] TO  DISK = N'C:\backup\SQLEXPRESS\DEMO.bak' WITH NOFORMAT, NOINIT,
RETAINDAYS = 13,  NAME = N'DEMO-Transaction Log  Backup', SKIP, NOREWIND, NOUNLOAD,  STATS = 10

GO
```

4. Create a shrink script.

```
USE [DEMO]

GO

DBCC SHRINKFILE (N'DEMO_Log' , 0, TRUNCATEONLY)

GO
```

5. Create an update stats and index script.

```
EXEC sp_updatestats

go

EXEC sp_MSforeachtable @command1='print "?"', @command2='set QUOTED_IDENTIFIER ON;ALTER INDEX
ALL ON ? REBUILD WITH (FILLFACTOR=90,ONLINE=OFF)'
```

We will be calculating an index of 90% capacity for fillfactor (which gives 10% growth for data indexes). Also, we will retain the backups for 13 days and starting over on the 14th day. For each script we will create a batch file to call the .sql files.

6. Create batch routines under root of c:\.
    - Full Backup batch file (example: full_backup.bat)

```
cd\

sqlcmd -S lukesavage\sqlexpress -i c:\backup\SQLEXPRESS\full_backup.sql -o
c:\backup\SQLEXPRESS\full_backup_log.txt
```

```
exit
```

- Intermediate or differential backup (example: intermediate_backup.bat)

```
cd\

sqlcmd -S lukesavage\sqlexpress -i c:\backup\SQLEXPRESS\intermediate_backup.sql -o
c:\backup\SQLEXPRESS\inter_backup_log.txt

exit
```

- Backup Logs and Truncate (example: backup_logs.bat)

```
cd\

sqlcmd -S lukesavage\sqlexpress -i c:\backup\SQLEXPRESS\truncate_logs.sql -o
c:\backup\SQLEXPRESS\truncate_log.txt

exit
```

- Shrink Data and Log files (example: shrink.bat)

```
cd\

sqlcmd -S lukesavage\sqlexpress -i c:\backup\SQLEXPRESS\shrink.sql -o
c:\backup\SQLEXPRESS\shrink_log.txt

exit
```

- Update Stats and Index batch file (example: update_stats_index.bat)

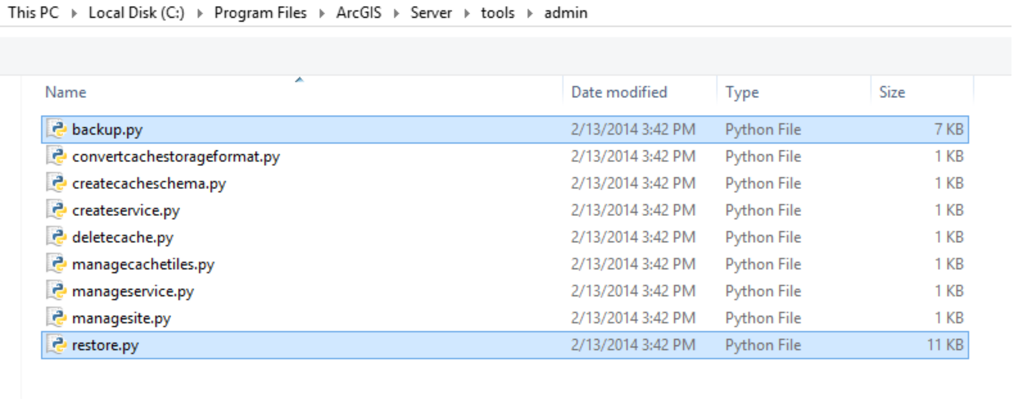
```
cd\

sqlcmd -S lukesavage\sqlexpress -d Demo -i c:\backup\SQLEXPRESS\update_stats_index.sql -o
c:\backup\SQLEXPRESS\VECTOR_update_stats_index_log.txt

exit
```

7. Create Windows Scheduled tasks for each backup file.
    - Full Backup: running once a week on a Saturday Early Morning
    - Differential or Intermediate backup: Running once a day except Saturday at the same time as the Full Backup
    - Backup Logs: Run once a day or twice a day depending on your workload and editing. Make sure you try to backup logs during lower workload hours.
    - Shrink Data and Logs: Run right after the backup log routines
    - Analyze, update statistics and indexes: Run once a day and/or before and after backups. Most will run this once a day after backups occur.
    - Create backup copy and cleanup procedures using command line scripts or delete utilities that can be scheduled

Backup ArcGIS for Server Site

Provided with ArcGIS for Server, you can now backup your site. This backup only accounts for the site configuration not the data. If you have caches, handle the backup of the caches using another copy or backup utility.

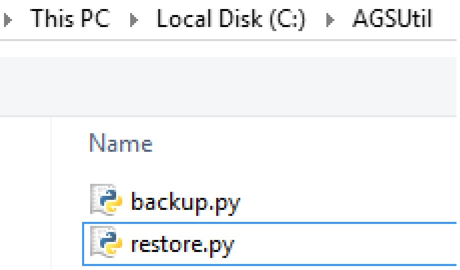
1. On your ArcGIS for Server, go to C:\Program Files\ArcGIS\Server\tools\admin and copy the two python script utilities.



This PC ▸ Local Disk (C:) ▸ Program Files ▸ ArcGIS ▸ Server ▸ tools ▸ admin

Name	Date modified	Type	Size
backup.py	2/13/2014 3:42 PM	Python File	7 KB
convertcachestorageformat.py	2/13/2014 3:42 PM	Python File	1 KB
createcacheschema.py	2/13/2014 3:42 PM	Python File	1 KB
createservice.py	2/13/2014 3:42 PM	Python File	1 KB
deletecache.py	2/13/2014 3:42 PM	Python File	1 KB
managecachetiles.py	2/13/2014 3:42 PM	Python File	1 KB
manageservice.py	2/13/2014 3:42 PM	Python File	1 KB
managesite.py	2/13/2014 3:42 PM	Python File	1 KB
restore.py	2/13/2014 3:42 PM	Python File	11 KB

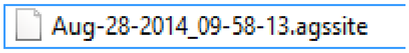
2. Create a new folder in a known location that is easy to get to in command line. As an example, I created an agsutil folder under the root of c:\.
3. Paste them in the new location.



▸ This PC ▸ Local Disk (C:) ▸ AGSUtil

Name
backup.py
restore.py

4. Go to the command prompt as an administrator, run the backup.py script. Make sure you are connecting to python's home directory or set python.exe as a path. Below is an example after setting python.exe path in environment variables.
   a. Python.exe c:\agsutil\backup.py –u <ags admin> -p <password> -s http://<localhost or servername:6080> -f c:\backup\ags
   b. After running, below is an example of the output.

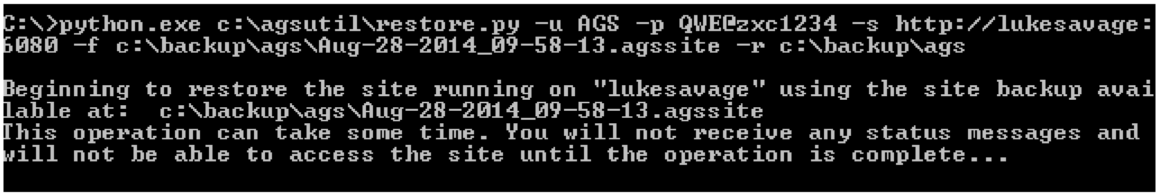


Aug-28-2014_09-58-13.agssite

Restore ArcGIS for Server Site

Provided with ArcGIS for Server, you can restore a backup of your site.

5. Go to command prompt and restore your site using the backup and restore.py script. Below is an example.
    a. Python.exe c:\agsutil\restore.py –u <username> -p <password> -s <localhost or servername>:6080 –f c:\backup\ags\Aug-29-2014_09-58-13.agssite –r c:\backup\ags
    b. This procedure will take some time and your site will be down until complete.



```
C:\>python.exe c:\agsutil\restore.py -u AGS -p QWE@zxc1234 -s http://lukesavage:
6080 -f c:\backup\ags\Aug-28-2014_09-58-13.agssite -r c:\backup\ags

Beginning to restore the site running on "lukesavage" using the site backup avai
lable at:  c:\backup\ags\Aug-28-2014_09-58-13.agssite
This operation can take some time. You will not receive any status messages and
will not be able to access the site until the operation is complete...
```

6. It is important that you reregister your web adapter to ArcGIS for Server after the restore is complete.

Automated Maintenance Scripts

With the release of ArcGIS v10.1 and higher, you have the capability to create python scripts to analyze, update statistics, and update index files. However, in this example we will show how to use ArcSDE and SQL Server command line.

SQL Server and ArcSDE Scripts

1. Install ArcSDE with Command line tools (**ArcGIS 10.2 is the last version to include command line tools. Post 10.2, plan to write automation scripts in python**)
2. Go to SQL Server Studio Management and open designer.
3. Create query. In this query, it will analyze and update statistics and rebuild indexes with a fill factor of 90%, which means you have 10% growth for indexes.

```
EXEC sp_updatestats

go

EXEC sp_MSforeachtable @command1='print "?"', @command2='set QUOTED_IDENTIFIER ON;ALTER
INDEX ALL ON ? REBUILD WITH (FILLFACTOR=90,ONLINE=OFF)'
```

4. Save query as update_stats_index.sql in the root of c:\
5. Create a Batch file that contains the following script. Notice the output log file created so you can keep track of your operations and if they fail troubleshoot the issue effectively. Items in Red signify zero state compression where all data is pushed to base tables. If zero state compression is not needed, delete the lines in Red below.
6. Save as <name>_state_compression.bat in the root of c:\.

Note: In a versioned environment, recommend zero state compression at least once every 6 months for lightweight editing. Heavy data editing environment zero state compression should be accomplished every quarter. Normal compression should be taking place once a week. This is to ensure data integrity long term.

```
sdemon -o kill -t all -p <sa password> -i sde:sqlserver:<instance> -s <server name> -D <geodatabase1> -N

sdemon -o kill -t all -p <sa password> -i sde:sqlserver:<instance> -s <server name> -D <geodatabase2> -N

sqlcmd -S localhost -U sa -P <sa password> -D <geodatabase1> -i c:\update_stats_index.sql -o c:\update_stats_preindex_log.txt

sqlcmd -S localhost -U sa -P <sa password> -D <geodatabase2> -i c:\update_stats_index.sql -o c:\update_stats_preindex_log.txt

sdeversion -o delete -i sde:sqlserver:<instance> -V dbo.edit -u sa -p <sa password> -s <server name> -D <geodatabase1> -N

sdeversion -o compress -u sa -p <sa password> -i sde:sqlserver:<instance> -s <server name> -D <geodatabase1> -N

sdeversion -o compress -u sa -p <sa password> -i sde:sqlserver:<instance> -s <server name> -D <geodatabase2> -N

sqlcmd -S localhost -U sa -P <sa password> -D <geodatabase1> -i c:\update_stats_index.sql -o c:\update_stats_index_postlog.txt

sqlcmd -S localhost -U sa -P <sa password> -D <geodatabase2> -i c:\update_stats_index.sql -o c:\update_stats_index_postlog.txt

sdeversion -o create -i sde:sqlserver:<instance> -V edit -P dbo.default -A public -u sa -p <sa password> -s <server name> -D <geodatabase1>

sqlcmd -S localhost -U sa -P <sa password> -D <geodatabase1> -i c:\update_stats_index.sql -o c:\update_stats_index_log.txt
```

7. Add a windows task in windows task scheduler on the server where ArcSDE binaries and command line tools reside.

Notes:____

____

____

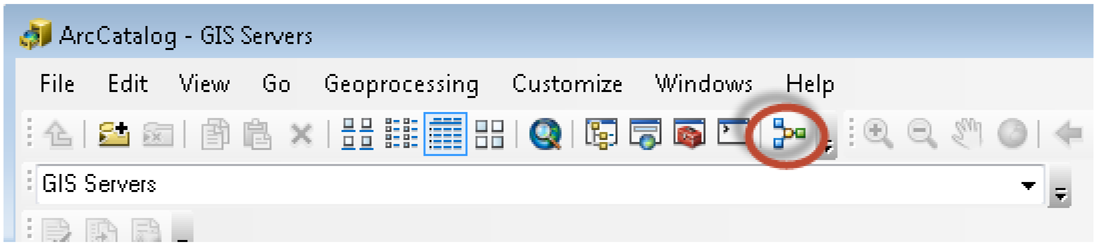
____

____

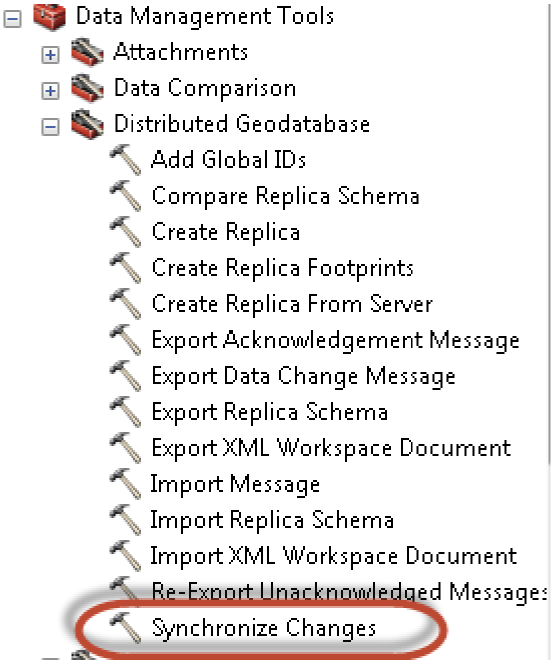
____

Automate Replica Synchronization using Python

1. Go to ArcCatalog and open model builder. This is the Garfield way of creating python scripts for the non-savvy coder.



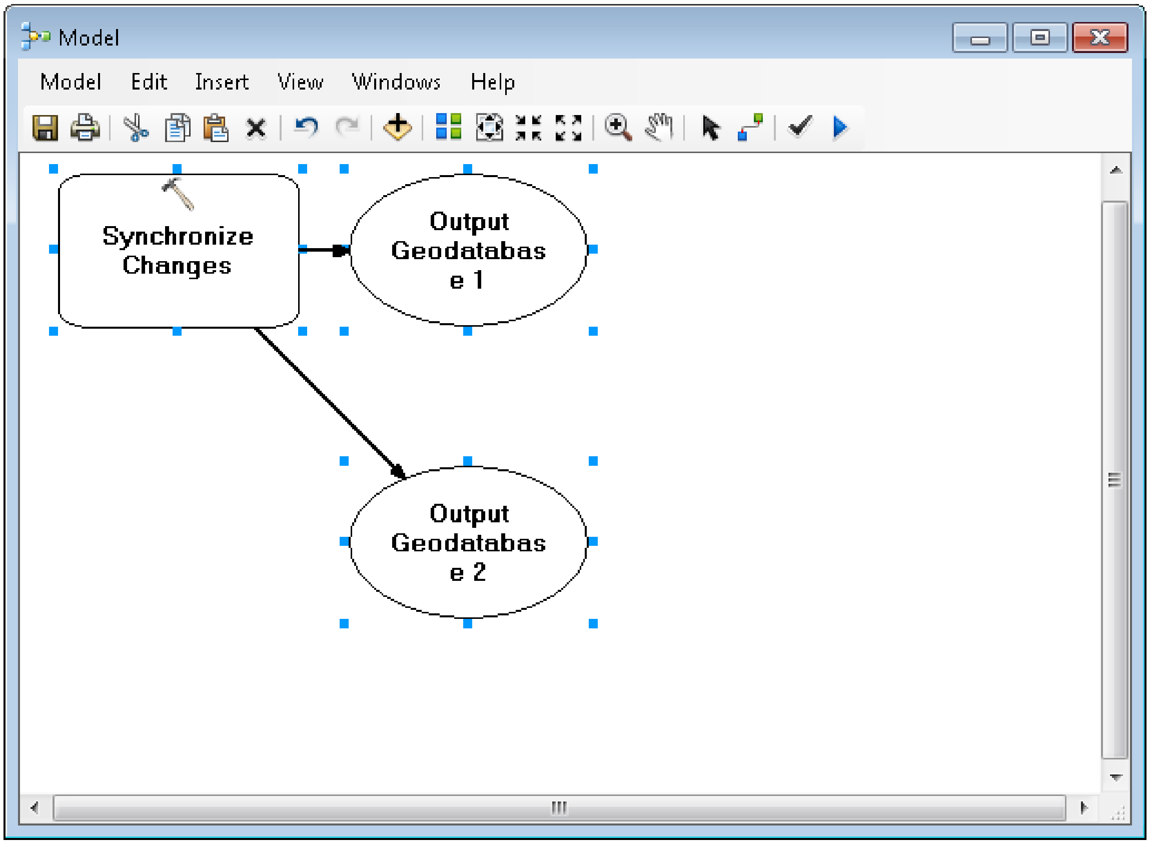
2. In Model Builder, open arctoolbox if not already open and go to Distributed Geodatabases/Synchronize Changes



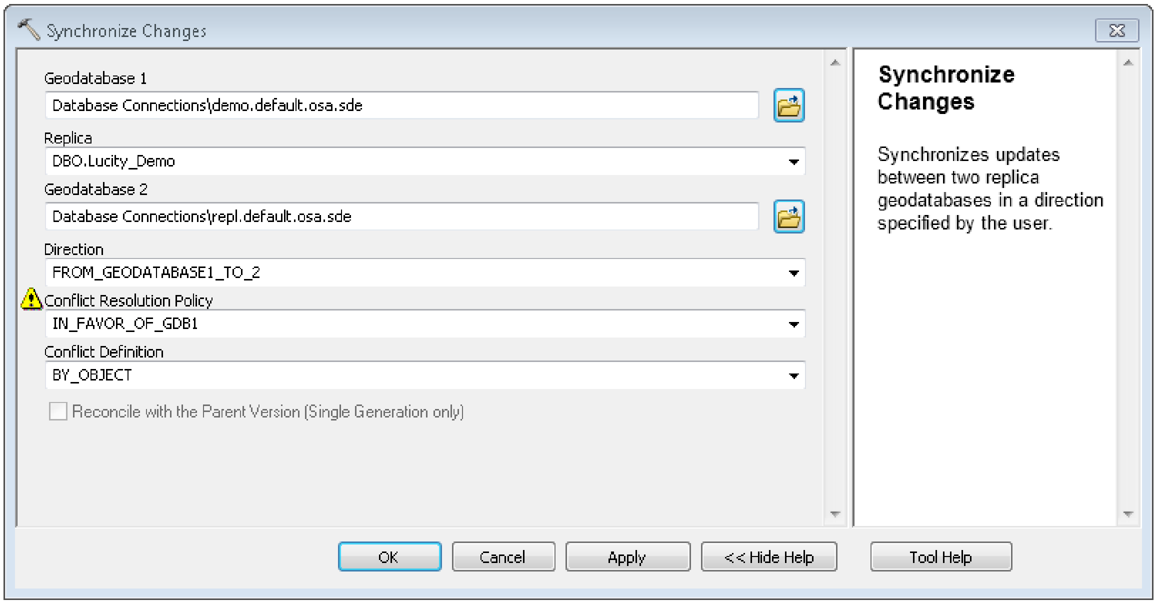
- Data Management Tools
  - Attachments
  - Data Comparison
  - Distributed Geodatabase
    - Add Global IDs
    - Compare Replica Schema
    - Create Replica
    - Create Replica Footprints
    - Create Replica From Server
    - Export Acknowledgement Message
    - Export Data Change Message
    - Export Replica Schema
    - Export XML Workspace Document
    - Import Message
    - Import Replica Schema
    - Import XML Workspace Document
    - Re-Export Unacknowledged Messages
    - Synchronize Changes

Notes:

3. Drag and drop the tool into Model Builder.



4. Once tool is in Model Builder, double click on Synchronize Changes box to open the dialog.
5. Fill in the information appropriately. For reference, please visit the Replica section.



Synchronize Changes

Geodatabase 1
Database Connections\demo.default.osa.sde

Replica
DBO.Lucity_Demo

Geodatabase 2
Database Connections\repl.default.osa.sde

Direction
FROM_GEODATABASE1_TO_2

Conflict Resolution Policy
IN_FAVOR_OF_GDB1

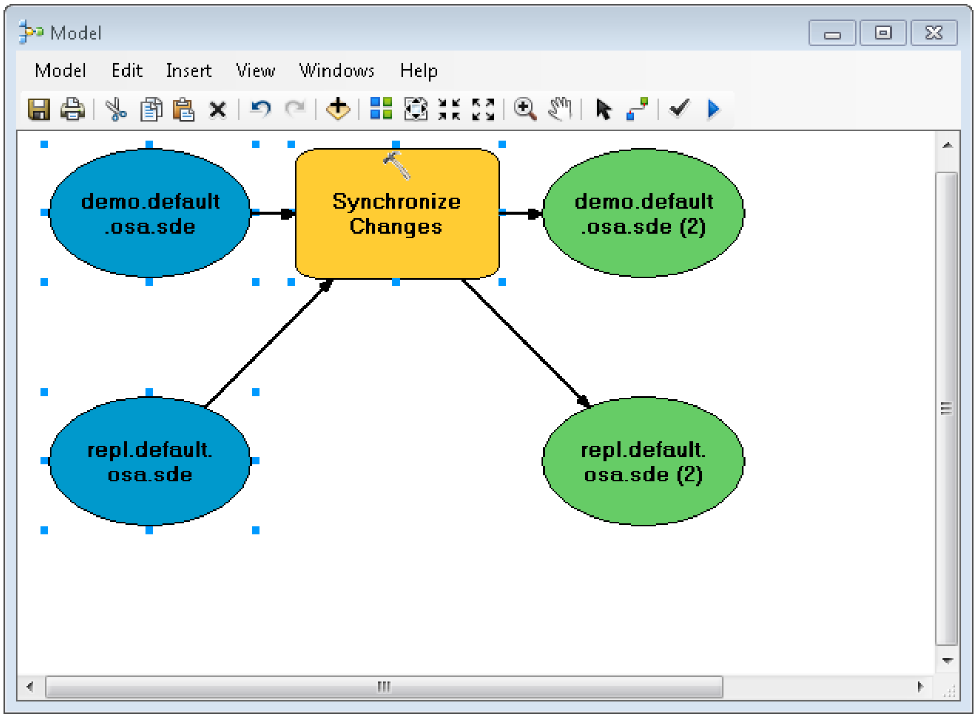
Conflict Definition
BY_OBJECT

□ Reconcile with the Parent Version (Single Generation only)

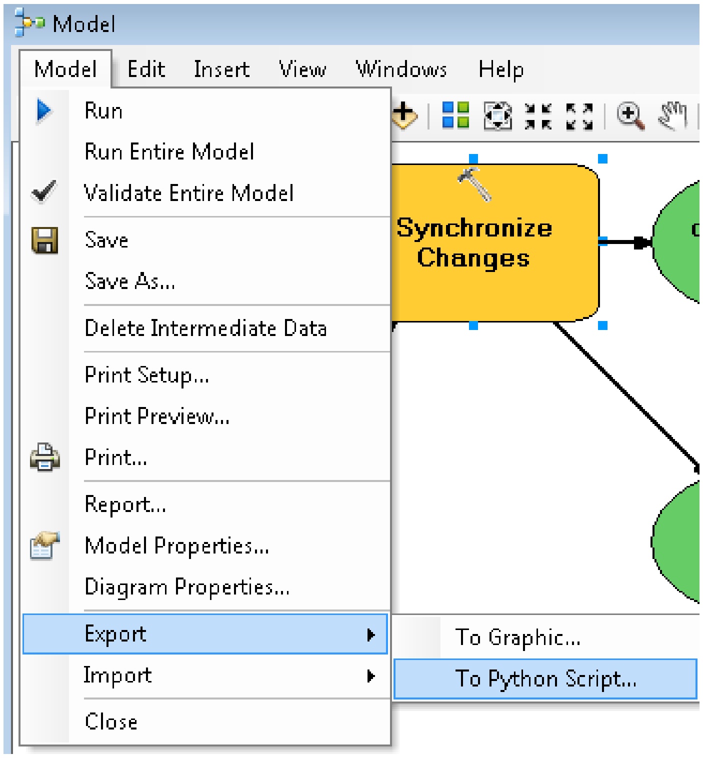
Synchronize Changes

Synchronizes updates between two replica geodatabases in a direction specified by the user.

6. Click ok. Notice colors change in the dialog. Run the model to test by clicking on the run button. ▶



7. After successfully testing the model, go to the model menu and click on export/To Python Script.

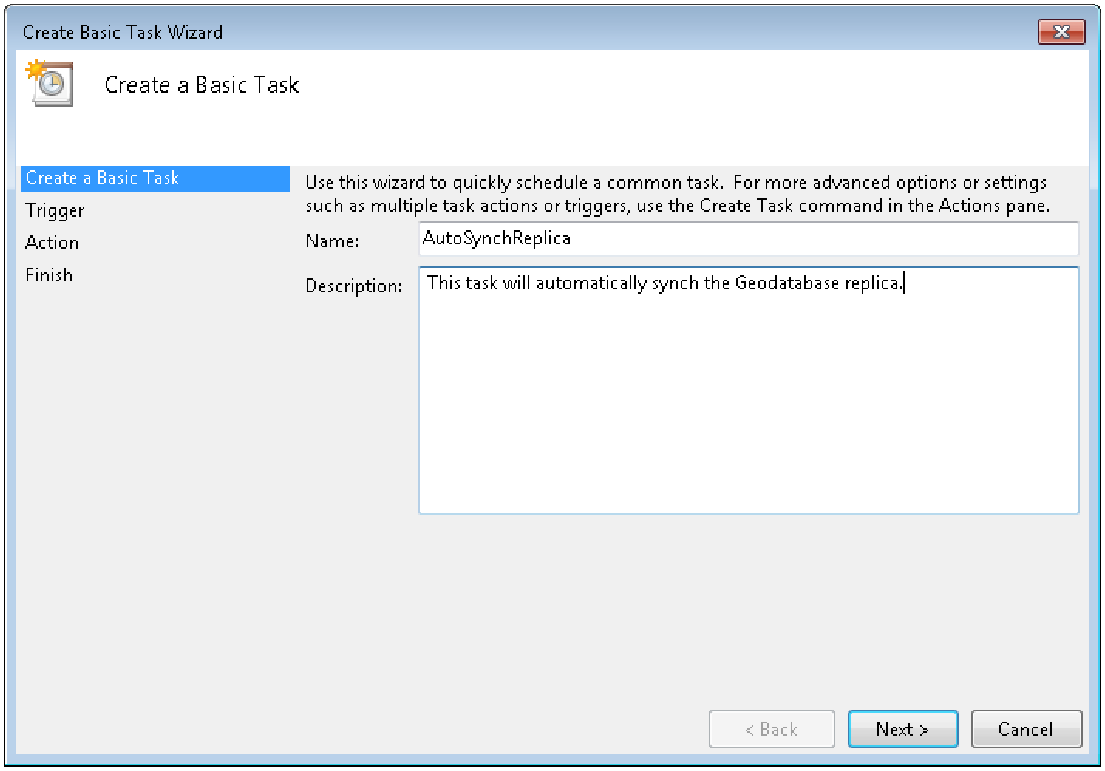


8. Save in location that has Python and ArcCatalog installed. For example, auto_synch.py under the root of c:\.
9. Edit the Python script so that it references arcinfo rather than arceditor. This can produce errors in your script. Also, add the RED content below to enable logging when running this script.

```
# Set the necessary product code
import arcinfo
# Import arcpy module
import arcpy
# Record Logs
arcpy.SetLogHistory(True)
# Local variables:
demo_default_osa_sde = "Database Connections\\demo.default.osa.sde"
repl_default_osa_sde = "Database Connections\\repl.default.osa.sde"
# Process: Synchronize Changes
arcpy.SynchronizeChanges_management(demo_default_osa_sde, "DBO.Lucity_Demo",
repl_default_osa_sde, "FROM_GEODATABASE1_TO_2", "IN_FAVOR_OF_GDB1", "BY_OBJECT",
"DO_NOT_RECONCILE")
```

Notes:

10. Go to Windows Task Scheduler and create a basic task.



Create Basic Task Wizard

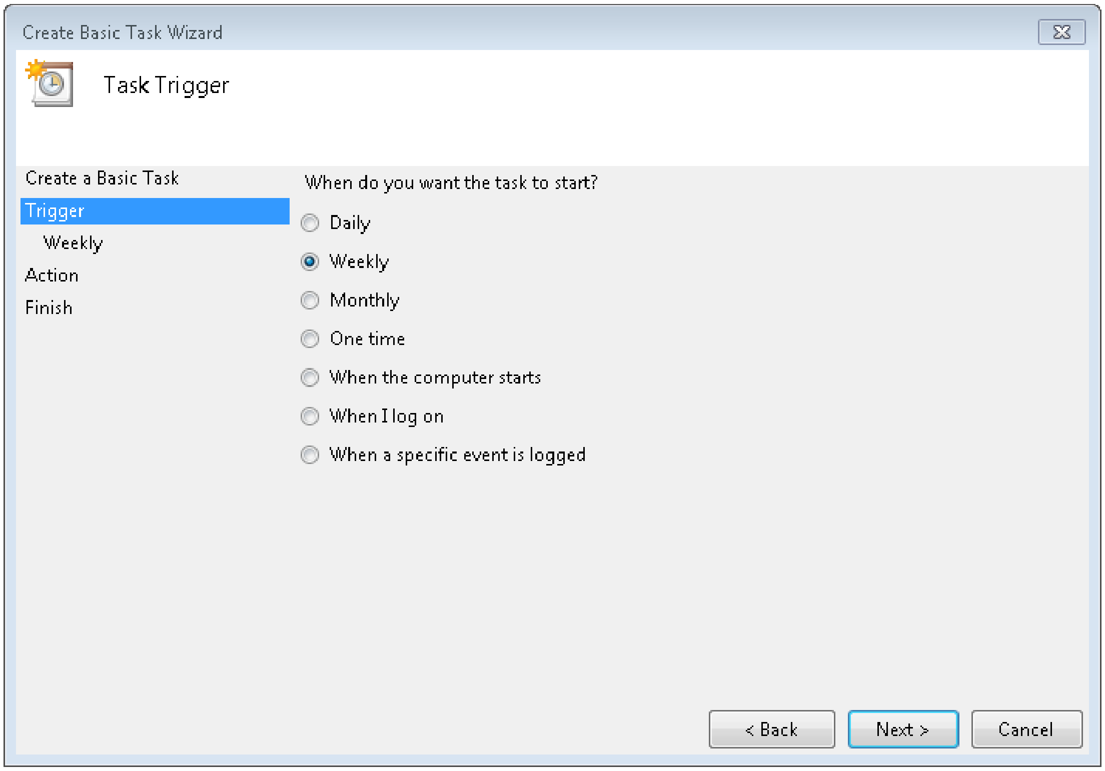
Create a Basic Task

Use this wizard to quickly schedule a common task. For more advanced options or settings such as multiple task actions or triggers, use the Create Task command in the Actions pane.

Name: AutoSynchReplica

Description: This task will automatically synch the Geodatabase replica.

11. Click next and schedule the synchronization of the replica to whenever you want the synchronization to take place. Usually each day or once a week will fit most organizations. In this example we are going to setup a weekly schedule. Click next.



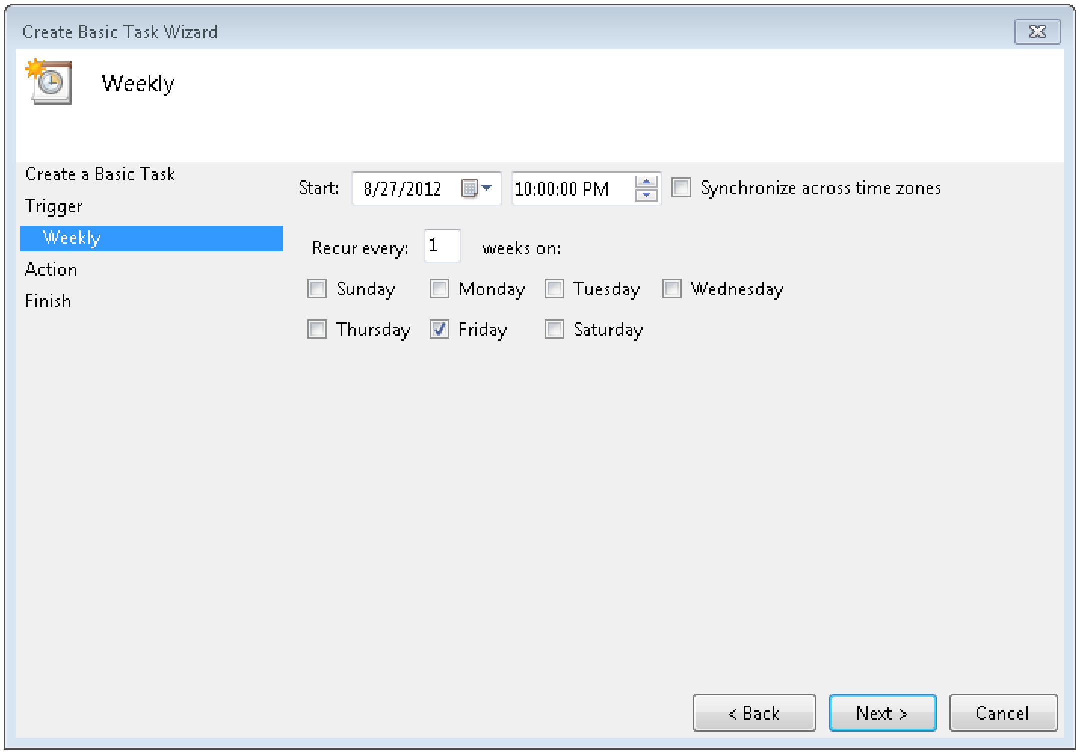
Create Basic Task Wizard

Task Trigger

When do you want the task to start?

○ Daily
● Weekly
○ Monthly
○ One time
○ When the computer starts
○ When I log on
○ When a specific event is logged

12. In the timeliness of this schedule, I want this to fire off before the backups have occurred so that my databases stay in tune. In my backup routines, I may have a performance maintenance script that analyzes, updates statistics and indexes as well as backups the transaction logs and shrinks the database. Therefore, in this example we'll set the synchronization to occur after I leave for the day on Friday. Click next.



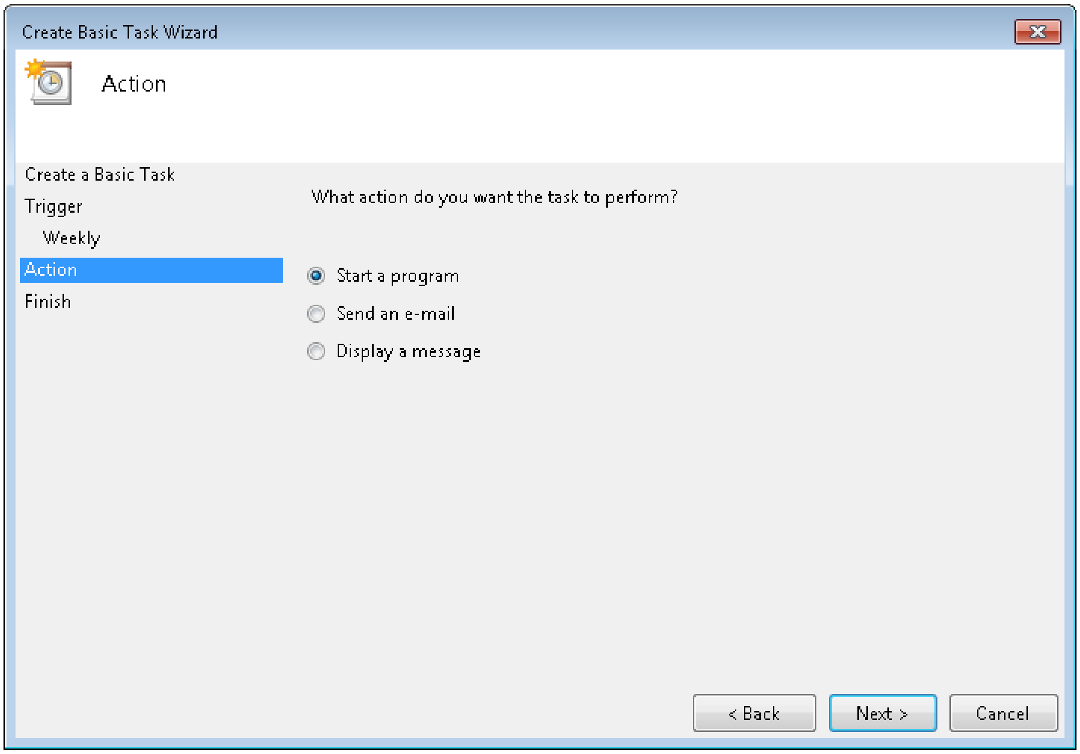
Start: 8/27/2012 10:00:00 PM □ Synchronize across time zones

Recur every: 1 weeks on:

□ Sunday □ Monday □ Tuesday □ Wednesday

□ Thursday ☑ Friday □ Saturday

13. Choose start a program and click next.



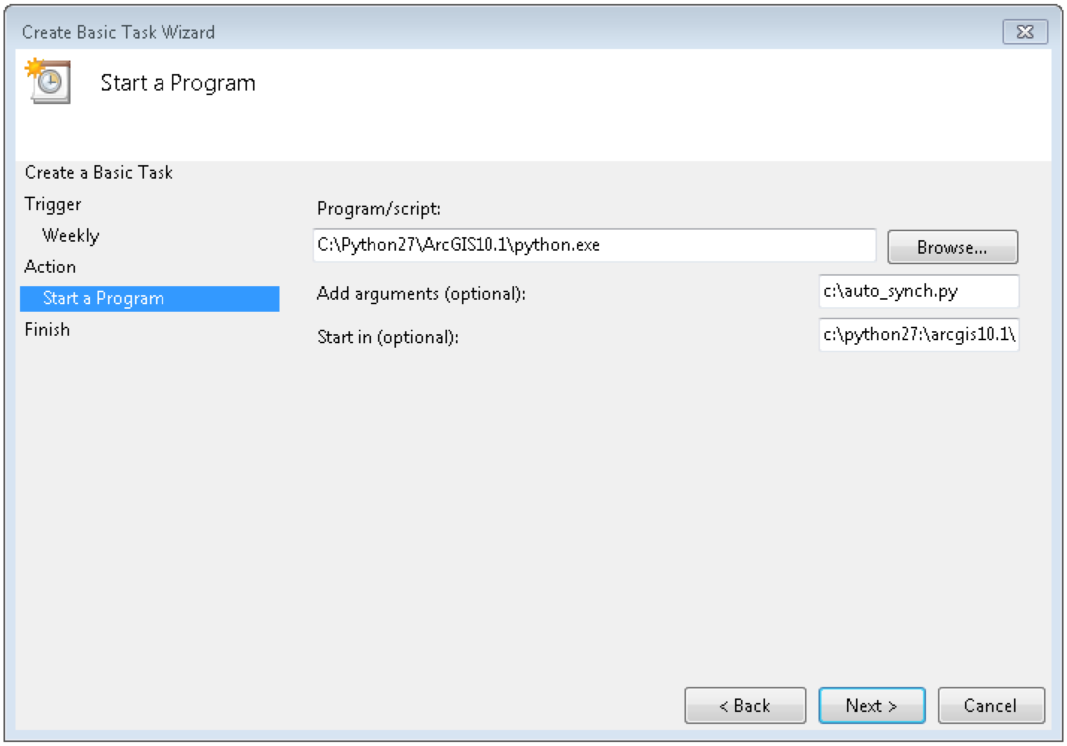
What action do you want the task to perform?

● Start a program

○ Send an e-mail

○ Display a message

14. In Windows Vista and above, we need to add the python program as the script so that it knows what executable to use. In Add arguments (optional):, you need to add the location of your saved python script. Also, in 'Start in (optional), you need to add the location folder of the python executable. Once complete, click next.

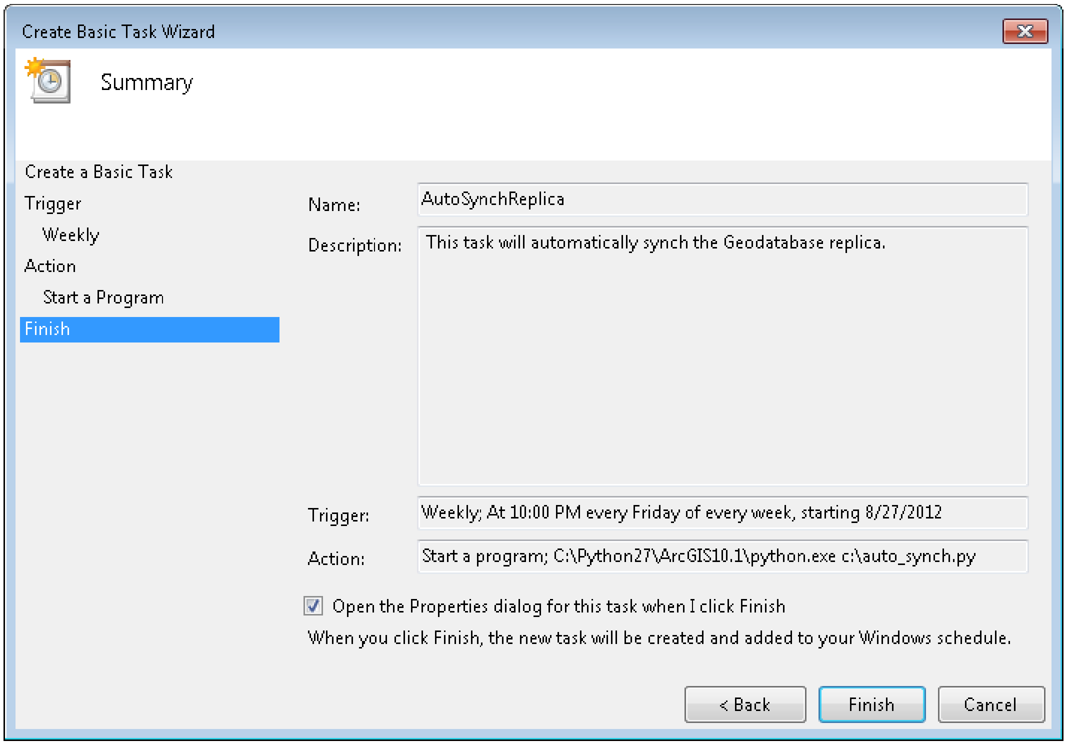


Program/script: C:\Python27\ArcGIS10.1\python.exe

Add arguments (optional): c:\auto_synch.py

Start in (optional): c:\python27:\arcgis10.1\

15. Click the 'Open the Properties dialog for this task when I click Finish' checkbox to verify administrative user to run this task and its credentials. Click Finish.



Name: AutoSynchReplica

Description: This task will automatically synch the Geodatabase replica.

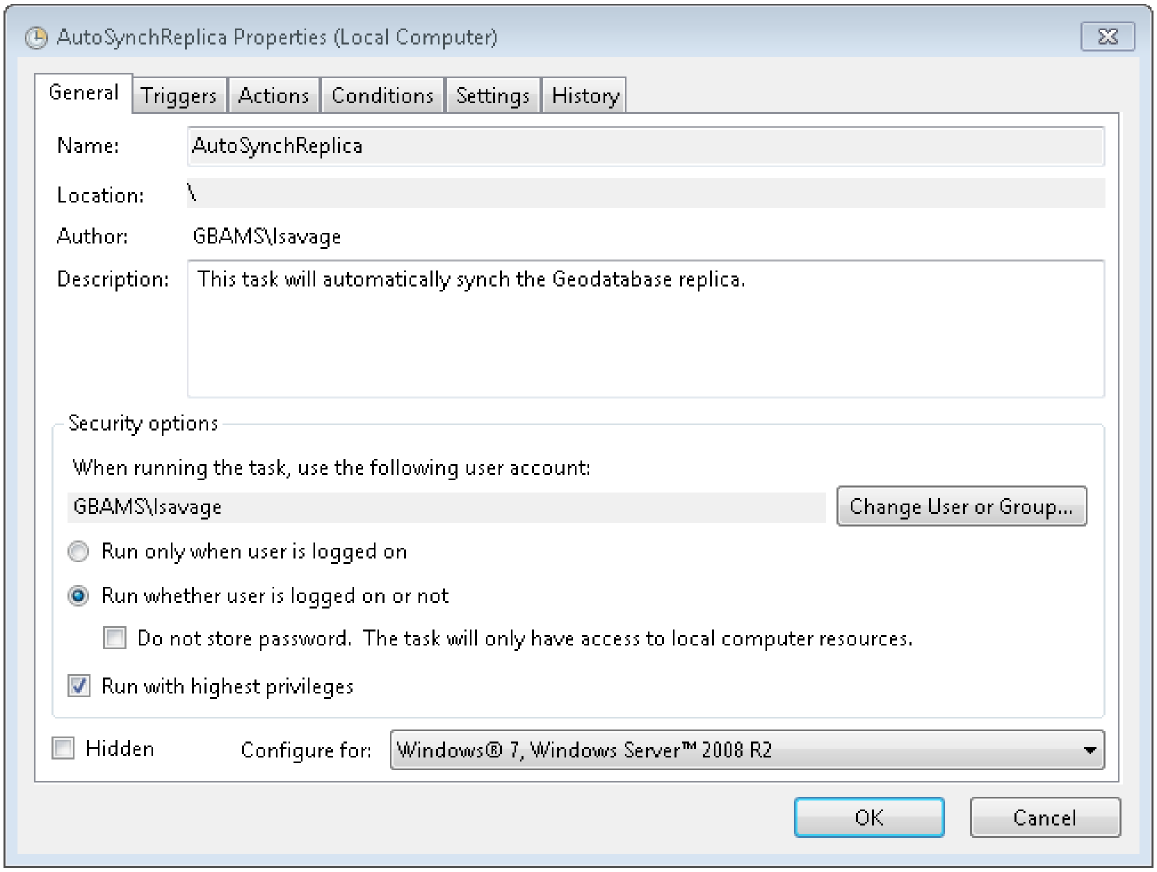
Trigger: Weekly; At 10:00 PM every Friday of every week, starting 8/27/2012

Action: Start a program; C:\Python27\ArcGIS10.1\python.exe c:\auto_synch.py

☑ Open the Properties dialog for this task when I click Finish

When you click Finish, the new task will be created and added to your Windows schedule.

16. Check the ‘Run whether user is logged on or not’ radio button and check the checkbox next to ‘Run with highest privileges’.



Name: AutoSynchReplica

Location: \

Author: GBAMS\lsavage

Description: This task will automatically synch the Geodatabase replica.

Security options

When running the task, use the following user account:

GBAMS\lsavage

○ Run only when user is logged on

● Run whether user is logged on or not

□ Do not store password. The task will only have access to local computer resources.

☑ Run with highest privileges

□ Hidden

Configure for: Windows® 7, Windows Server™ 2008 R2

17. Test the task by right clicking on the new task and select run.
18. Check the ArcToolBox results history location by going to the following location.

C:\Users\<username>\AppData\Roaming\ESRI\Desktop10.2\ArcToolbox\History

Notes:

Options

Versioning

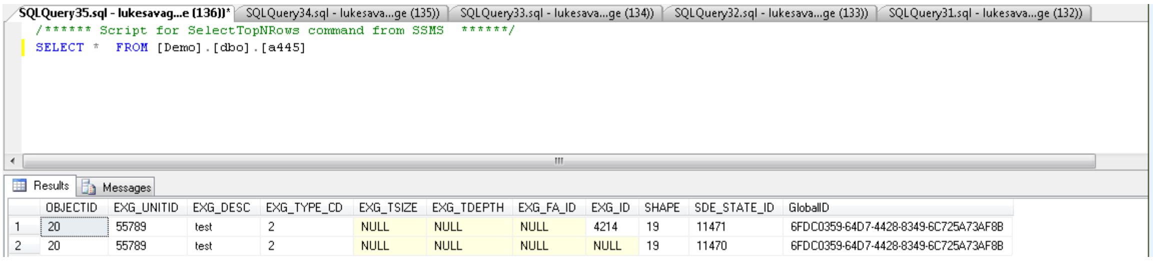
Is an alternative state of the database where you can make edits and changes that will not affect the base tables. When complete and edits are ready, the parent table will be reconciled and posted with the child version. Within the Geodatabase, there are two tables that store changes to the base data. These are called delta tables which are known as Add and Delete tables.

What are A and D tables?

- A tables are add tables. Anytime you add a record or change a record, the changes are added to the A table.
- D tables are delete tables. Anytime you delete a record, the delete rows are added to the D table.
- A and D tables are numbered based on the registration_id in the SDE_Table_Registry table.

Example of finding an Add Table.

If you have made an edit to a feature class, you need to get the registration_id for that feature class and go to the appropriate A table. This is stored in the sde_table_registry in SQL Server. For child versions, they are subsequent IDs such as default version would be the actual registration_id (A144) and the child version would be (A145). Same applies to the D tables.



SQLQuery35.sql - lukesavag...e (136))*
SQLQuery34.sql - lukesava...ge (135))
SQLQuery33.sql - lukesava...ge (134))
SQLQuery32.sql - lukesava...ge (133))
SQLQuery31.sql - lukesava...ge (132))
Results
Messages

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT *  FROM [Demo].[dbo].[a445]
```

	OBJECTID	EXG_UNITID	EXG_DESC	EXG_TYPE_CD	EXG_TSIZE	EXG_TDEPTH	EXG_FA_ID	EXG_ID	SHAPE	SDE_STATE_ID	GlobalID
1	20	55789	test	2	NULL	NULL	NULL	4214	19	11471	6FDC0359-64D7-4428-8349-6C725A73AF8B
2	20	55789	test	2	NULL	NULL	NULL	NULL	19	11470	6FDC0359-64D7-4428-8349-6C725A73AF8B

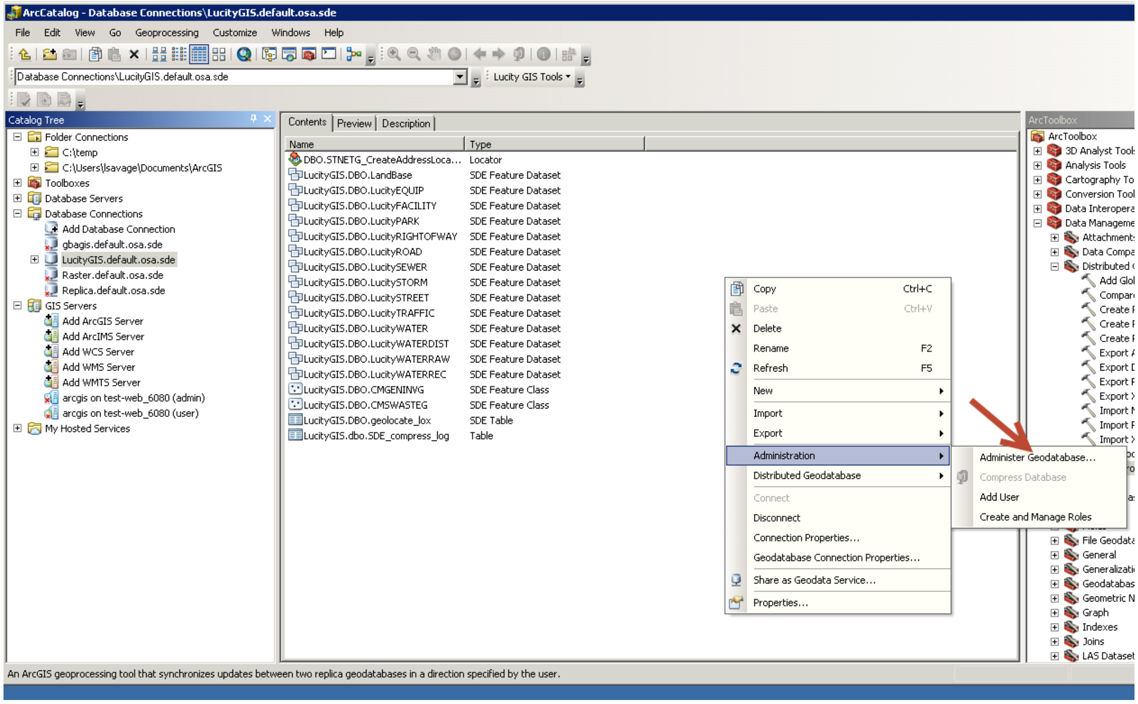
When Does Versioning Make Sense?

The option to moving edits to base tables is a scary thing for GIS people. Why?

- You can edit simple data only—points, lines, polygons, annotation, and relationships. You cannot edit a feature class in a topology, geometric network, or terrain.
- You cannot archive changes for the dataset.
- You cannot replicate the dataset.
- When you edit the DEFAULT version or post a version to the DEFAULT, you do not have the ability to resolve conflicts, so it is possible to overwrite another user's edits.

How to Create a Version

1. Right click on a Database Connection link in ArcCatalog TOC or right click in the white space in the contents tab in ArcCatalog

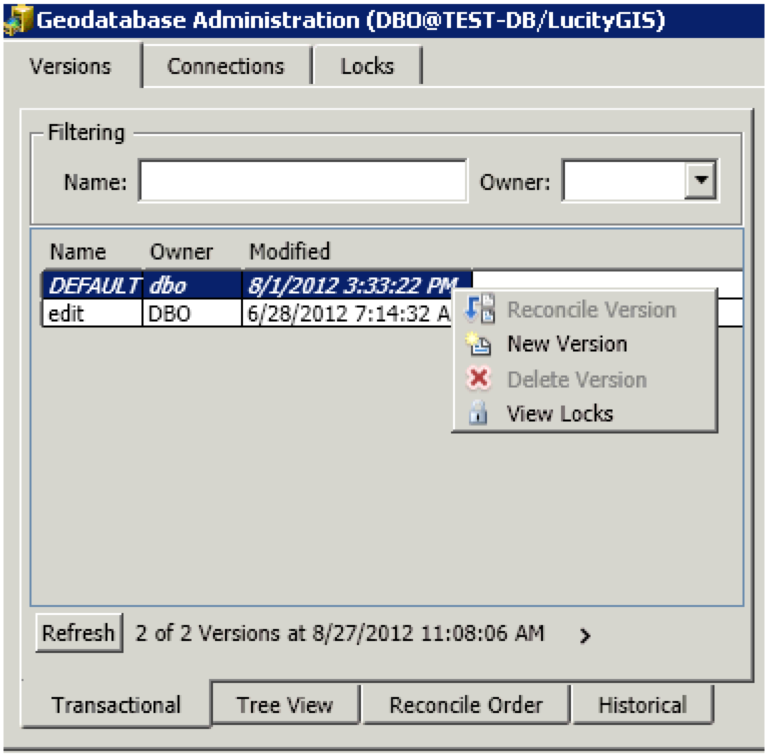


Name	Type
DBO.STNETG_CreateAddressLoca...	Locator
LucityGIS.DBO.LandBase	SDE Feature Dataset
LucityGIS.DBO.LucityEQUIP	SDE Feature Dataset
LucityGIS.DBO.LucityFACILITY	SDE Feature Dataset
LucityGIS.DBO.LucityPARK	SDE Feature Dataset
LucityGIS.DBO.LucityRIGHTOFWAY	SDE Feature Dataset
LucityGIS.DBO.LucityROAD	SDE Feature Dataset
LucityGIS.DBO.LucitySEWER	SDE Feature Dataset
LucityGIS.DBO.LucitySTORM	SDE Feature Dataset
LucityGIS.DBO.LucitySTREET	SDE Feature Dataset
LucityGIS.DBO.LucityTRAFFIC	SDE Feature Dataset
LucityGIS.DBO.LucityWATER	SDE Feature Dataset
LucityGIS.DBO.LucityWATERDIST	SDE Feature Dataset
LucityGIS.DBO.LucityWATERRAW	SDE Feature Dataset
LucityGIS.DBO.LucityWATERREC	SDE Feature Dataset
LucityGIS.DBO.CMGENINVG	SDE Feature Class
LucityGIS.DBO.CMSWASTEG	SDE Feature Class
LucityGIS.DBO.geolocate_lox	SDE Table
LucityGIS.dbo.SDE_compress_log	Table

An ArcGIS geoprocessing tool that synchronizes updates between two replica geodatabases in a direction specified by the user.

2. Click on Administration/Administrator Geodatabase.

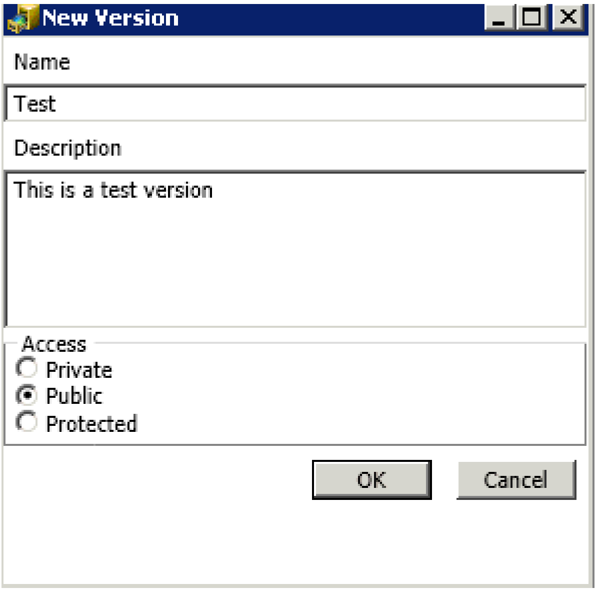
3. Right click on the Default version and select new version



Name	Owner	Modified
DEFAULT	dbo	8/1/2012 3:33:22 PM
edit	DBO	6/28/2012 7:14:32 A

2 of 2 Versions at 8/27/2012 11:08:06 AM

4. Create a Name for the new version and select public. Click Ok.



New Version

Name

Test

Description

This is a test version

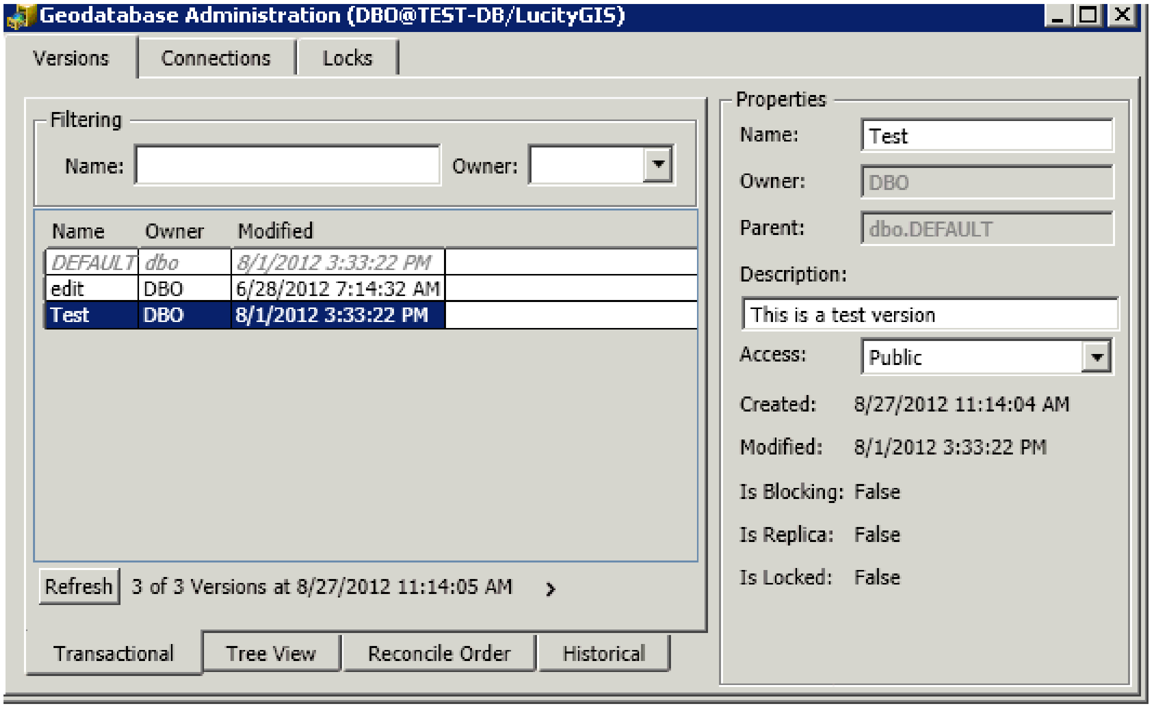
Access

○ Private

● Public

○ Protected

5. What's the difference between Access types
   - Private: Created user eyes only; no one has access unless owner of the version
   - Public: Everyone can see the version and create versions from. If you have edit privileges, you can edit the version.
   - Protected: Everyone can see the version but cannot edit the version unless owner. Everyone can create versions from.
6. This is now the child of Default and is listed in the Geodatabase Administration dialog.



Geodatabase Administration (DBO@TEST-DB/LucityGIS)

Filtering

Name:

Owner:

Name	Owner	Modified
DEFAULT	*dbo*	*8/1/2012 3:33:22 PM*
edit	DBO	6/28/2012 7:14:32 AM
Test	DBO	8/1/2012 3:33:22 PM

3 of 3 Versions at 8/27/2012 11:14:05 AM

Properties

Name: Test

Owner: DBO

Parent: dbo.DEFAULT

Description:

This is a test version

Access: Public

Created: 8/27/2012 11:14:04 AM

Modified: 8/1/2012 3:33:22 PM

Is Blocking: False

Is Replica: False

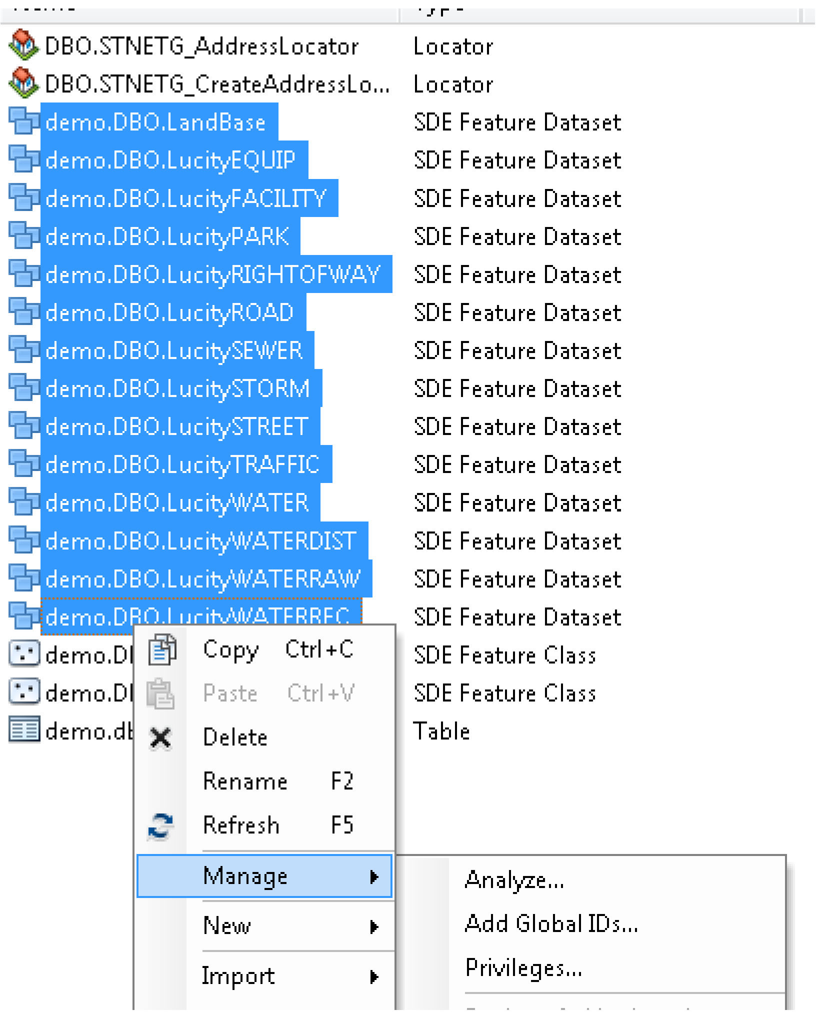
Is Locked: False

Replica

A replica is a copy of a database using GUID attributes that synchronize from the parent to the child databases or vice versa. There are several ways to create a replica but we will be creating the most likely replica used which is a one-way replica. One way replicas are easy to setup and are handy for the administrators to offload view only users off of the production database while performing synchronizations from time to time. Keeping the database in synch when production changes have been approved is a nice way to distribute your data to users while retaining data integrity, security and flexibility.

1. Prepare your data for a replica. Go to your database and right click on Feature Datasets, and go to manage/add global ids. Global Ids are necessary for the Geodatabase to keep track of changes in each database that is a part of a replica.

Note: Data must be registered as versioned

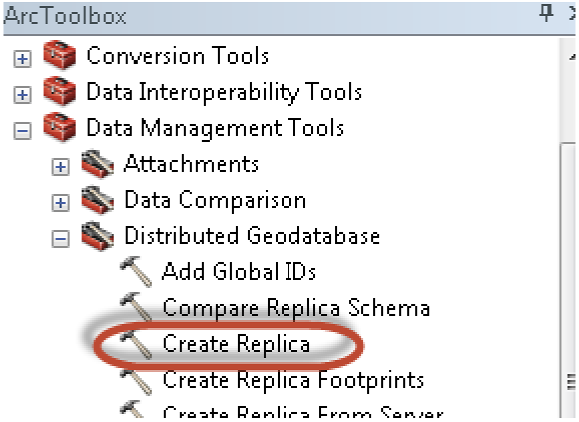


DBO.STNETG_AddressLocator	Locator
DBO.STNETG_CreateAddressLo...	Locator
demo.DBO.LandBase	SDE Feature Dataset
demo.DBO.LucityEQUIP	SDE Feature Dataset
demo.DBO.LucityFACILITY	SDE Feature Dataset
demo.DBO.LucityPARK	SDE Feature Dataset
demo.DBO.LucityRIGHTOFWAY	SDE Feature Dataset
demo.DBO.LucityROAD	SDE Feature Dataset
demo.DBO.LucitySEWER	SDE Feature Dataset
demo.DBO.LucitySTORM	SDE Feature Dataset
demo.DBO.LucitySTREET	SDE Feature Dataset
demo.DBO.LucityTRAFFIC	SDE Feature Dataset
demo.DBO.LucityWATER	SDE Feature Dataset
demo.DBO.LucityWATERDIST	SDE Feature Dataset
demo.DBO.LucityWATERRAW	SDE Feature Dataset
demo.DBO.LucityWATERREC	SDE Feature Dataset
demo.DI	SDE Feature Class
demo.DI	SDE Feature Class
demo.dt	Table

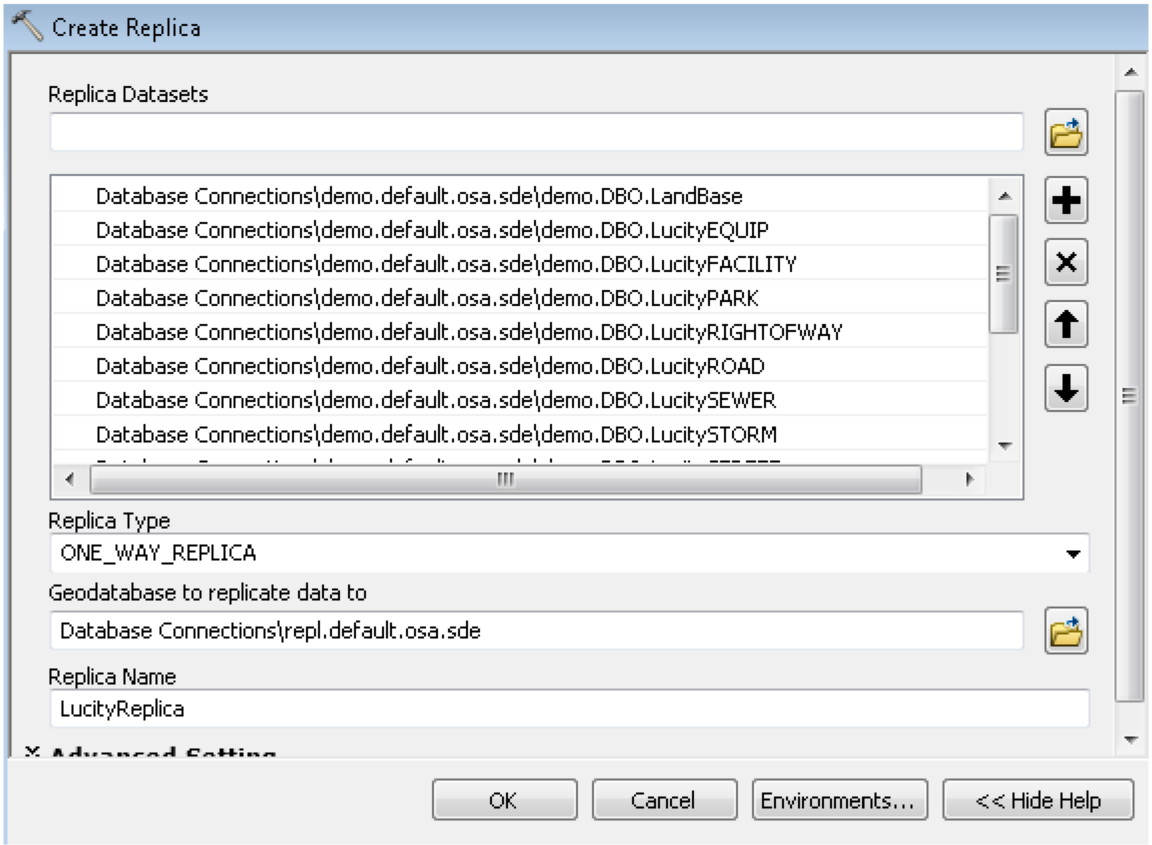
2. Do this for stand-alone Feature Classes as well.

3. To create a replica, go to ArcToolbox/Data Management Tools/Distributed Geodatabase and expand.

4. Click on Create Replica



5. Browse or drag and drop data you want to replicate from into the Create Replica dialog window. Change the Replica Type to one-way replica. Add the replicated database connection to replicate the data to. Give the Replica a name.



Replica Datasets

Database Connections\demo.default.osa.sde\demo.DBO.LandBase
Database Connections\demo.default.osa.sde\demo.DBO.LucityEQUIP
Database Connections\demo.default.osa.sde\demo.DBO.LucityFACILITY
Database Connections\demo.default.osa.sde\demo.DBO.LucityPARK
Database Connections\demo.default.osa.sde\demo.DBO.LucityRIGHTOFWAY
Database Connections\demo.default.osa.sde\demo.DBO.LucityROAD
Database Connections\demo.default.osa.sde\demo.DBO.LucitySEWER
Database Connections\demo.default.osa.sde\demo.DBO.LucitySTORM

Replica Type
ONE_WAY_REPLICA

Geodatabase to replicate data to
Database Connections\repl.default.osa.sde

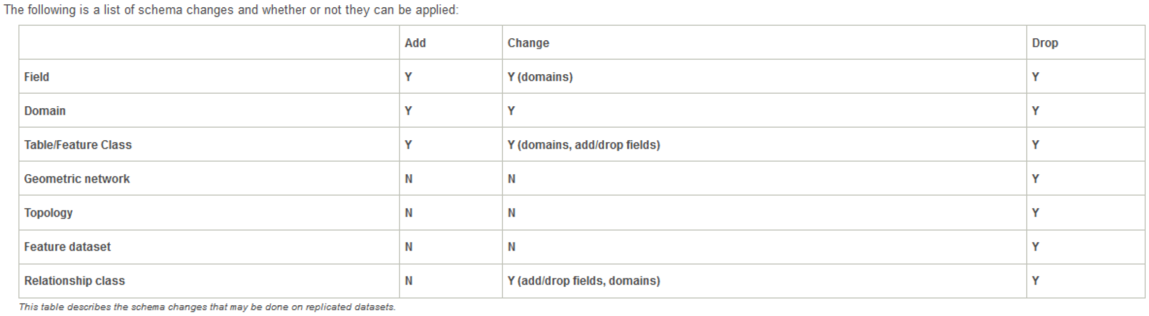
Replica Name
LucityReplica

6. The advanced settings allow you to change the behavior of your replica. Usually, you would except the defaults and click ok.

7. Once replica is in place, you can assign privileges.

Apply Schema Changes to Replica

In order to update the schema in a replica, there are three tools that you can use to update the schema. However, there are certain instances where you will need to unregister the replica, delete the replica feature class/tables/datasets and recreate the replica in order for the changes to push down from the parent to the child. Below is a matrix from Esri's ArcGIS Resources page for schema changes. It describes what is supported with the update process and what is not supported.



The following is a list of schema changes and whether or not they can be applied:

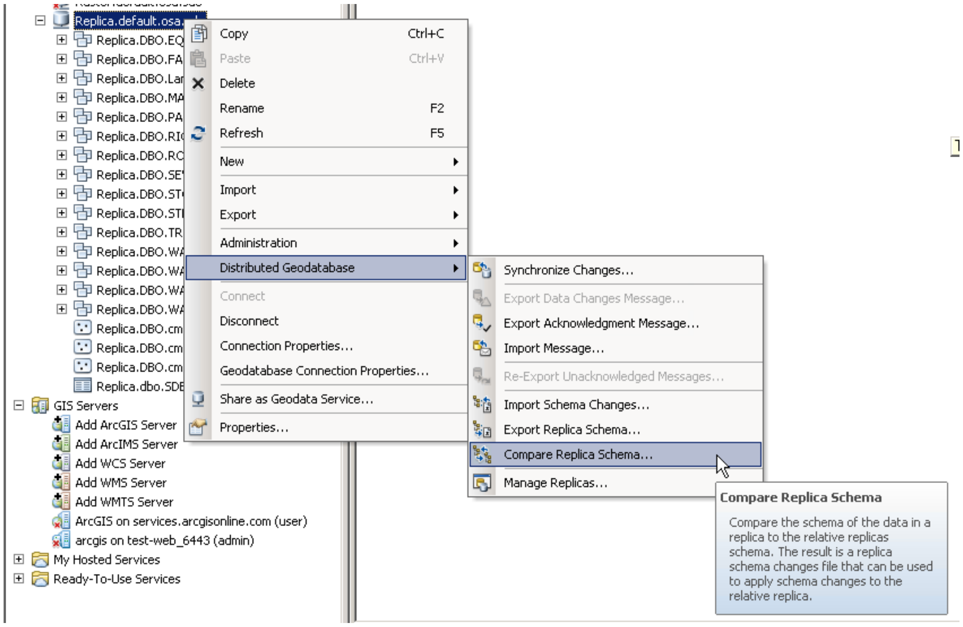
	Add	Change	Drop
Field	Y	Y (domains)	Y
Domain	Y	Y	Y
Table/Feature Class	Y	Y (domains, add/drop fields)	Y
Geometric network	N	N	Y
Topology	N	N	Y
Feature dataset	N	N	Y
Relationship class	N	Y (add/drop fields, domains)	Y

This table describes the schema changes that may be done on replicated datasets.

Steps to complete

- Compare Schema in Parent
- Import Schema into Child

1. Open ArcCatalog.
2. Right Click on the Replica database and choose distributed geodatabase/compare replica schema

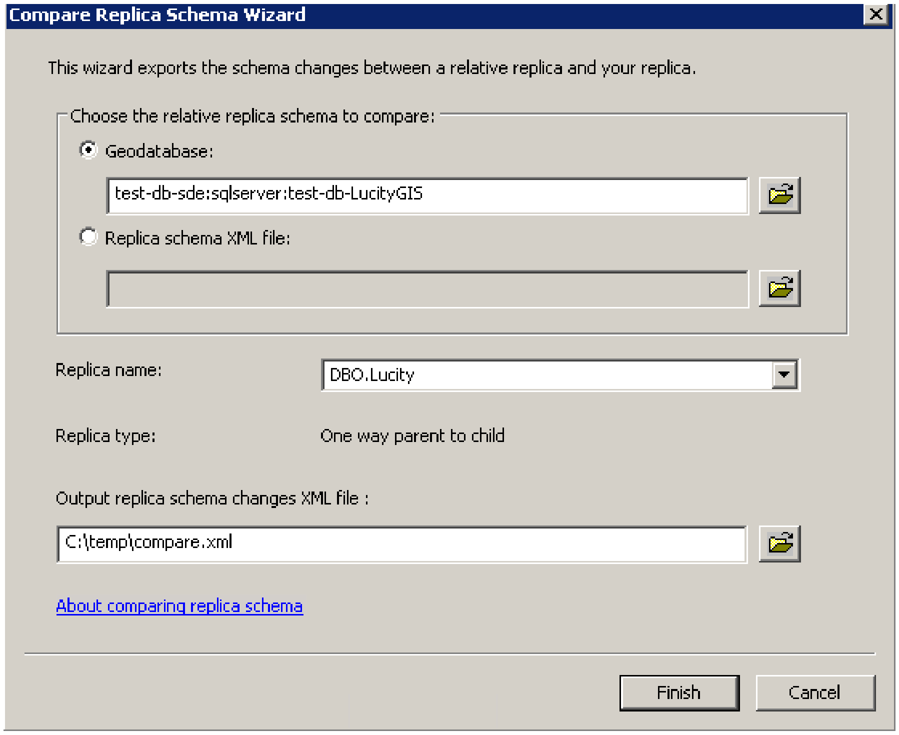


Compare Replica Schema

Compare the schema of the data in a replica to the relative replicas schema. The result is a replica schema changes file that can be used to apply schema changes to the relative replica.

3. Browse to the parent geodatabase
4. Choose Replica Name if you have multiple replicas

5. Choose a location to save the replica schema changes xml file



Compare Replica Schema Wizard

This wizard exports the schema changes between a relative replica and your replica.

Choose the relative replica schema to compare:

● Geodatabase: test-db-sde:sqlserver:test-db-LucityGIS

○ Replica schema XML file:

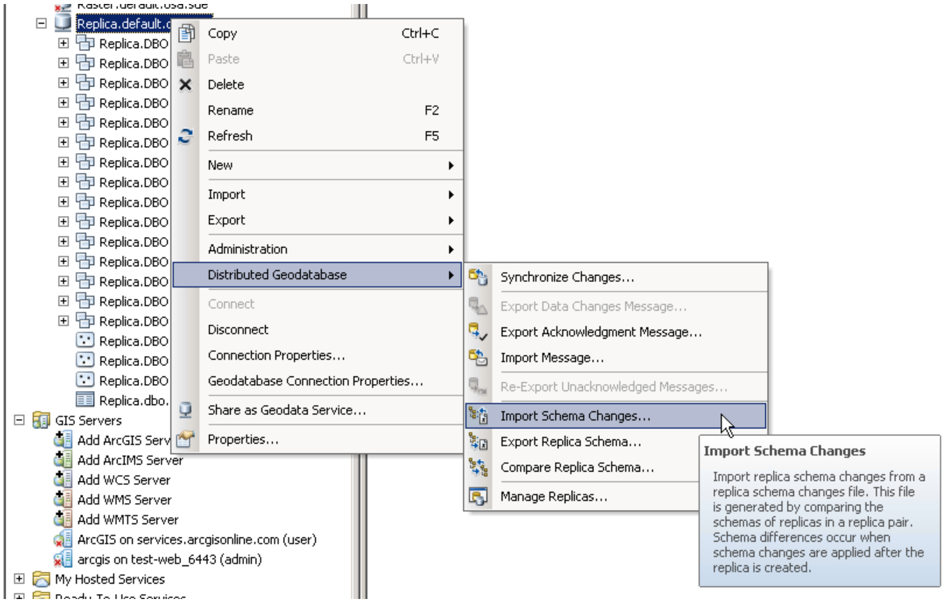
Replica name: DBO.Lucity

Replica type: One way parent to child

Output replica schema changes XML file : C:\temp\compare.xml

About comparing replica schema

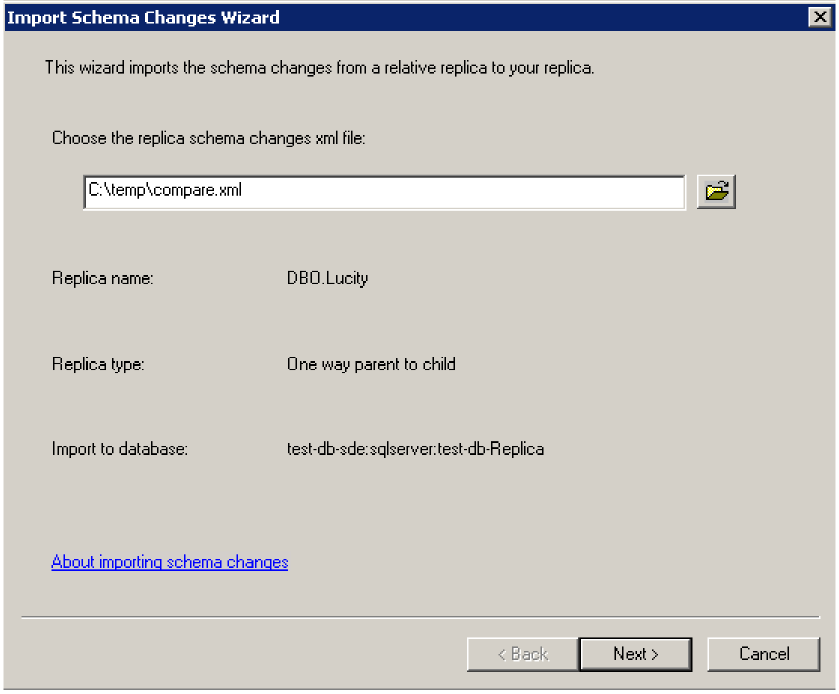
6. After complete, right click on the Replica database and choose distributed geodatabase/import replica schema



Import Schema Changes

Import replica schema changes from a replica schema changes file. This file is generated by comparing the schemas of replicas in a replica pair. Schema differences occur when schema changes are applied after the replica is created.

7. Locate the schema change xml file generated and add to the import schema changes wizard



Import Schema Changes Wizard

This wizard imports the schema changes from a relative replica to your replica.

Choose the replica schema changes xml file:

C:\temp\compare.xml

Replica name: DBO.Lucity

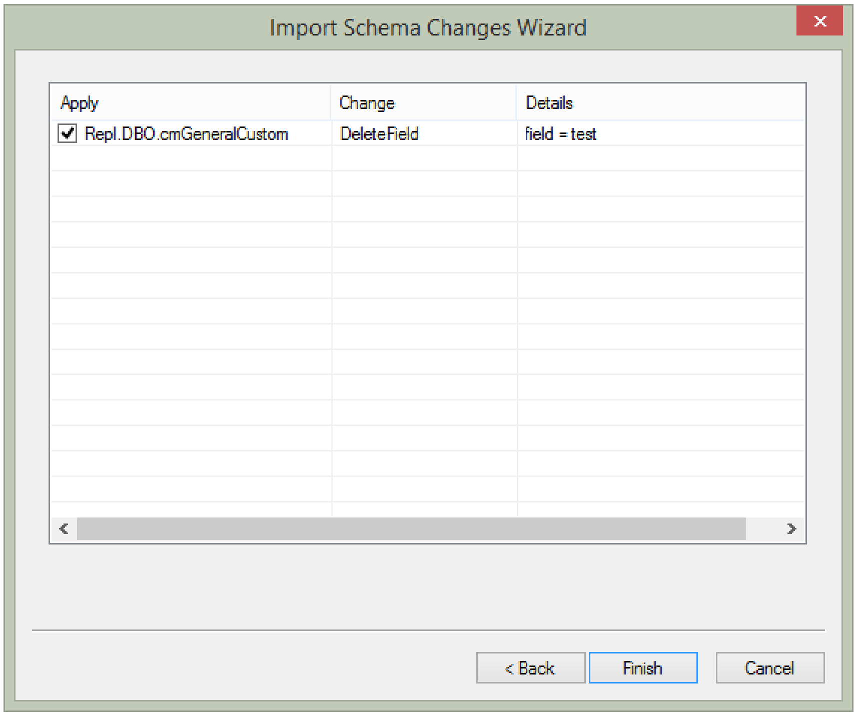
Replica type: One way parent to child

Import to database: test-db-sde:sqlserver:test-db-Replica

About importing schema changes

8. Review the information and click next

9. In the next screen, you should see the changes to the schema to apply. In this scenario, we deleted a field called ‘test; short int’.



Import Schema Changes Wizard

Apply	Change	Details
☑ Repl.DBO.cmGeneralCustom	DeleteField	field = test

10. Click Finish and Synchronize Changes.