Automating Administration Tasks Using Python

SEE WHAT OTHERS CAN'T

Mohammad Ashraf Dar
gistec

gisworx

Session Roadmap

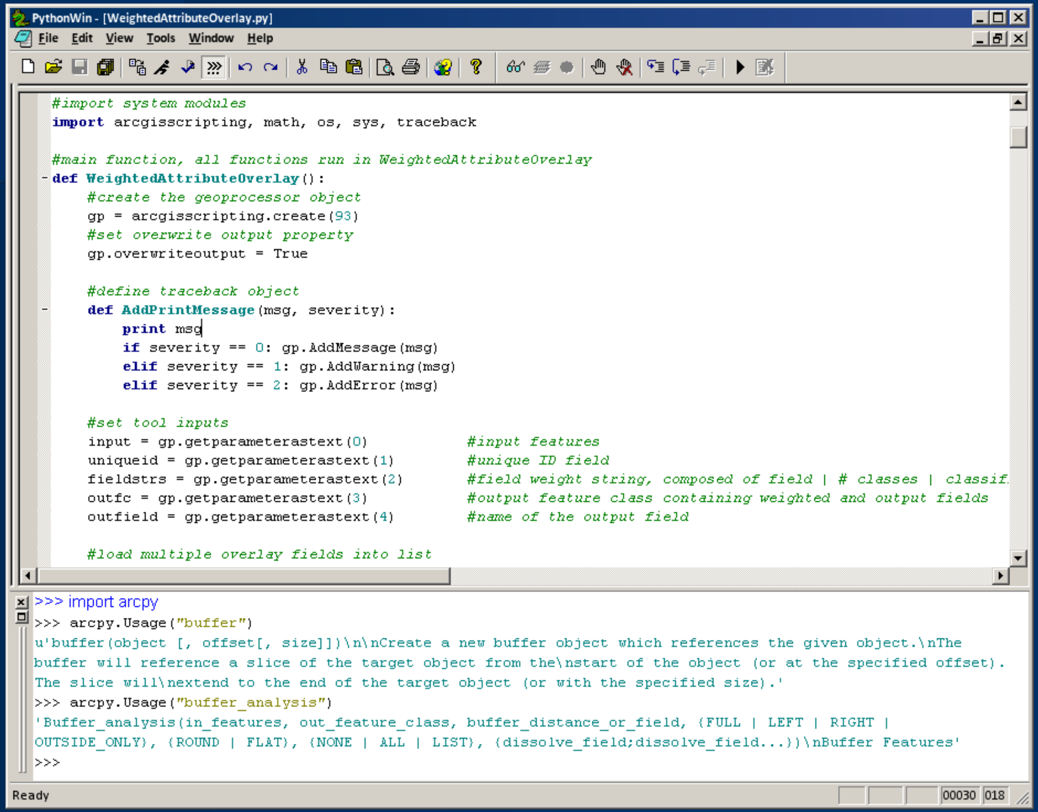
Session is divided into three parts

- **Part 1: Types of Administrators**
- **Part 2: Geodatabase Creation**
- **Part 3: Version Management**

gisworx

Python

- Free
- Simple and easy to learn
- Easy to maintain
- Wide-acceptance
- Modular
- Cross platform
- Scheduling
- Documentation of workflows


PythonWin - [WeightedAttributeOverlay.py]
File Edit View Tools Window Help

```
#import system modules
import arcgisscripting, math, os, sys, traceback

#main function, all functions run in WeightedAttributeOverlay
def WeightedAttributeOverlay():
    #create the geoprocessor object
    gp = arcgisscripting.create(93)
    #set overwrite output property
    gp.overwriteoutput = True

    #define traceback object
    def AddPrintMessage(msg, severity):
        print msg
        if severity == 0: gp.AddMessage(msg)
        elif severity == 1: gp.AddWarning(msg)
        elif severity == 2: gp.AddError(msg)

    #set tool inputs
    input = gp.getparameterastext(0)           #input features
    uniqueid = gp.getparameterastext(1)        #unique ID field
    fieldstrs = gp.getparameterastext(2)       #field weight string, composed of field | # classes | classif
    outfc = gp.getparameterastext(3)           #output feature class containing weighted and output fields
    outfield = gp.getparameterastext(4)        #name of the output field

    #load multiple overlay fields into list
```

```
>>> import arcpy
>>> arcpy.Usage("buffer")
u'buffer(object [, offset[, size]])\n\nCreate a new buffer object which references the given object.\nThe
buffer will reference a slice of the target object from the\nstart of the object (or at the specified offset).
The slice will\nextend to the end of the target object (or with the specified size).'
>>> arcpy.Usage("buffer_analysis")
'Buffer_analysis(in_features, out_feature_class, buffer_distance_or_field, {FULL | LEFT | RIGHT |
OUTSIDE_ONLY}, {ROUND | FLAT}, {NONE | ALL | LIST}, {dissolve_field;dissolve_field...})\nBuffer Features'
>>>
```

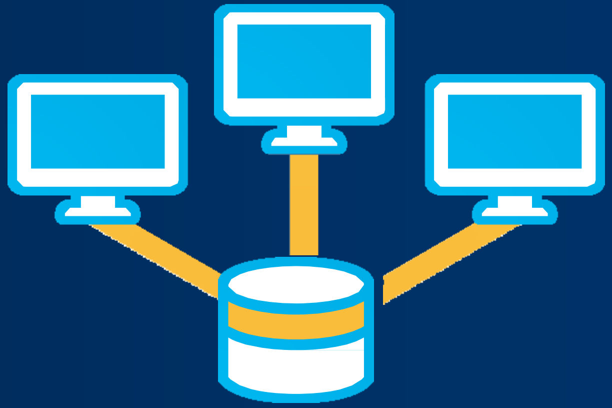
Data Sources

- File Geodatabases
  - System files in a file folder

- Enterprise Geodatabases
  - Oracle, SQL Server, PostgreSQL,
  - DB2, SAP HANA

- Enterprise Databases
  - Altibase, Demang, Netezza, Teradata

- Other systems
  - Hadoop

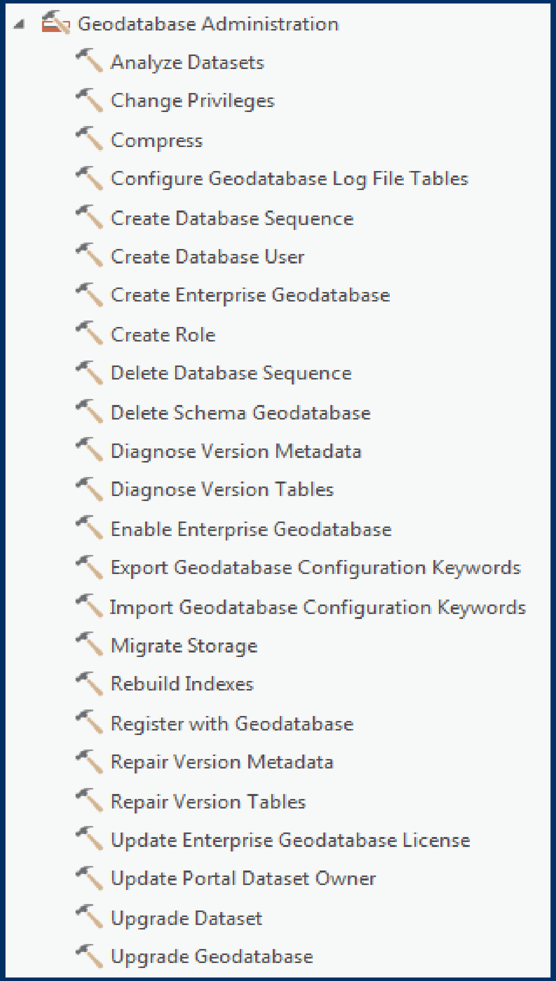


gisworx

Administration in Desktop and Server

- ArcGIS Desktop
  - GUI Tools
  - GP Tools

- ArcGIS for Server
  - GP Tools / Python



Geodatabase Administration
- Analyze Datasets
- Change Privileges
- Compress
- Configure Geodatabase Log File Tables
- Create Database Sequence
- Create Database User
- Create Enterprise Geodatabase
- Create Role
- Delete Database Sequence
- Delete Schema Geodatabase
- Diagnose Version Metadata
- Diagnose Version Tables
- Enable Enterprise Geodatabase
- Export Geodatabase Configuration Keywords
- Import Geodatabase Configuration Keywords
- Migrate Storage
- Rebuild Indexes
- Register with Geodatabase
- Repair Version Metadata
- Repair Version Tables
- Update Enterprise Geodatabase License
- Update Portal Dataset Owner
- Upgrade Dataset
- Upgrade Geodatabase

gisworx

Types of administrators

- ❑ **Database administrator** (DBA)
- ❑ **Geodatabase administrator** (sde)
- ❑ **Dataset administrator** (a.k.a data owner)

gisworx

Types of administrators

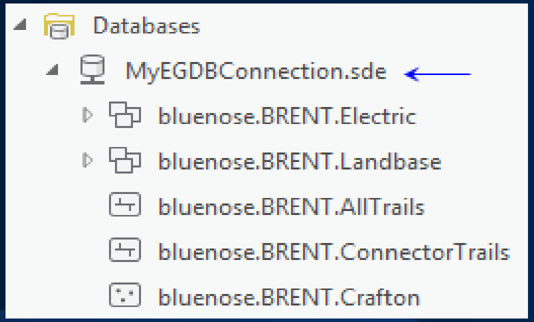
Database Administrator (DBA)	Geodatabase Administrator (sde user)	Dataset Administrator (data owner)
• Instance level admin • User management • Database backup • Performance monitoring	• Creates the geodatabase repository • View and reconcile any version • Performs Compress • Configuration keyword maintenance	• Grant privileges on data that they own • Modifying schema • Database statistics and index maintenance • Enabling geodatabase behavior on data tables

gisworx

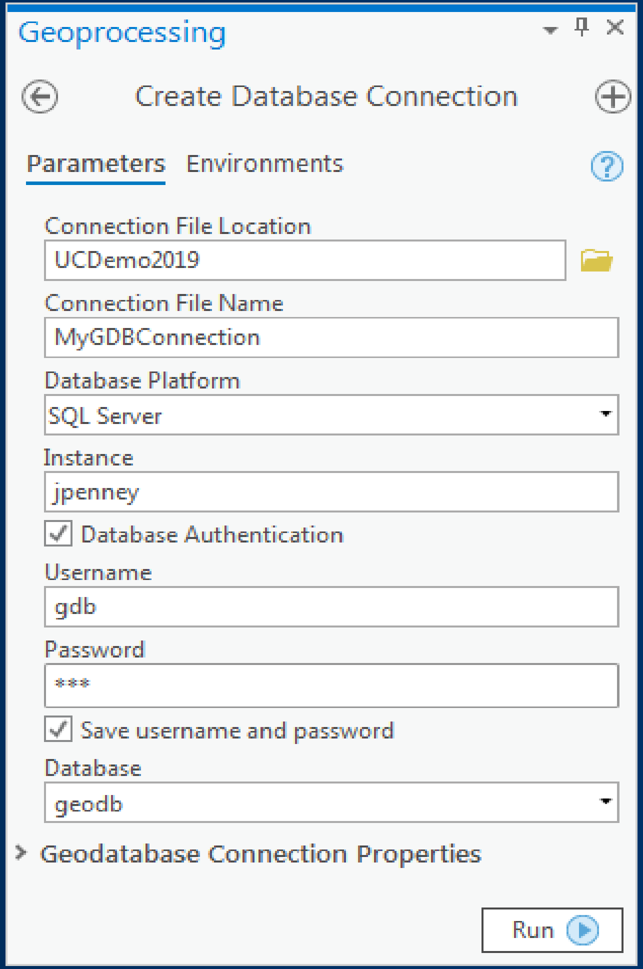
Connecting to an enterprise geodatabase

Connect to the geodatabase

- Create Database Connection GP tool
- Connection files are used by all admins and users
- Can use database or OS authentication
- Connection to a specific version



Databases
- MyEGDBConnection.sde
  - bluenose.BRENT.Electric
  - bluenose.BRENT.Landbase
  - bluenose.BRENT.AllTrails
  - bluenose.BRENT.ConnectorTrails
  - bluenose.BRENT.Crafton



Geoprocessing

Create Database Connection

Connection File Location: UCDemo2019
Connection File Name: MyGDBConnection
Database Platform: SQL Server
Instance: jpenney
☑ Database Authentication
Username: gdb
Password: ***
☑ Save username and password
Database: geodb

gisworx

Geodatabase Creation

Performed by Database Administrator (DBA)

Geodatabase Creation
- Create Enterprise Geodatabase GP tool
  - SQL Server, PostgreSQL, Oracle, etc.
  - License File required

Create Roles in the Geodatabase
- Create Role GP tool
  - Easy to assign/revoke privileges to a group of users

Create Users in the Geodatabase
- Create Database User GP tool
  - Assign to a role when creating a new user
  - Can be database or OS authentication

gisworx

Geodatabase Creation

Performed by Dataset Administrator (data owner)

Create or Load Data into the Geodatabase

- Create Table, Create Feature Class, etc.
- Import XML Workspace, Copy, Feature Class to Geodatabase, etc.

Manage Privileges

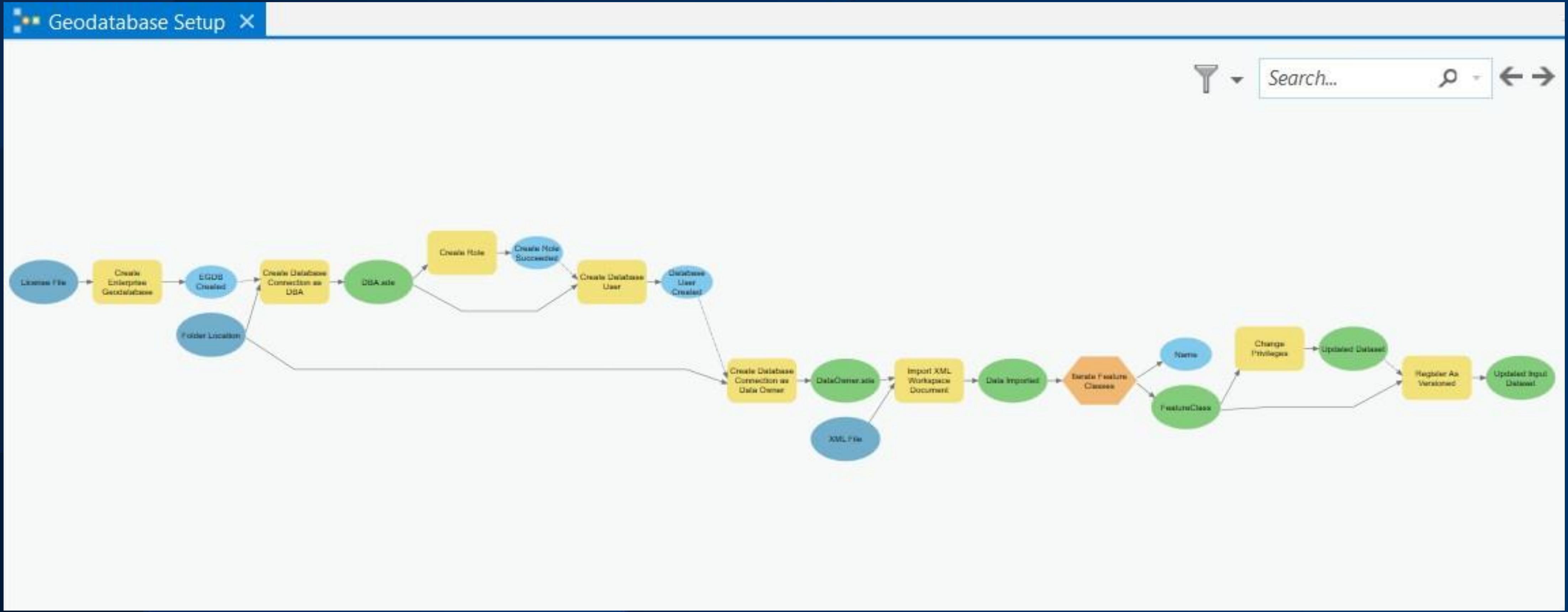
- Change Privileges GP tool
    - Grant or revoke view or edit permissions

Register As Versioned

- Register As Versioned GP tool
    - If using versioned editing workflows

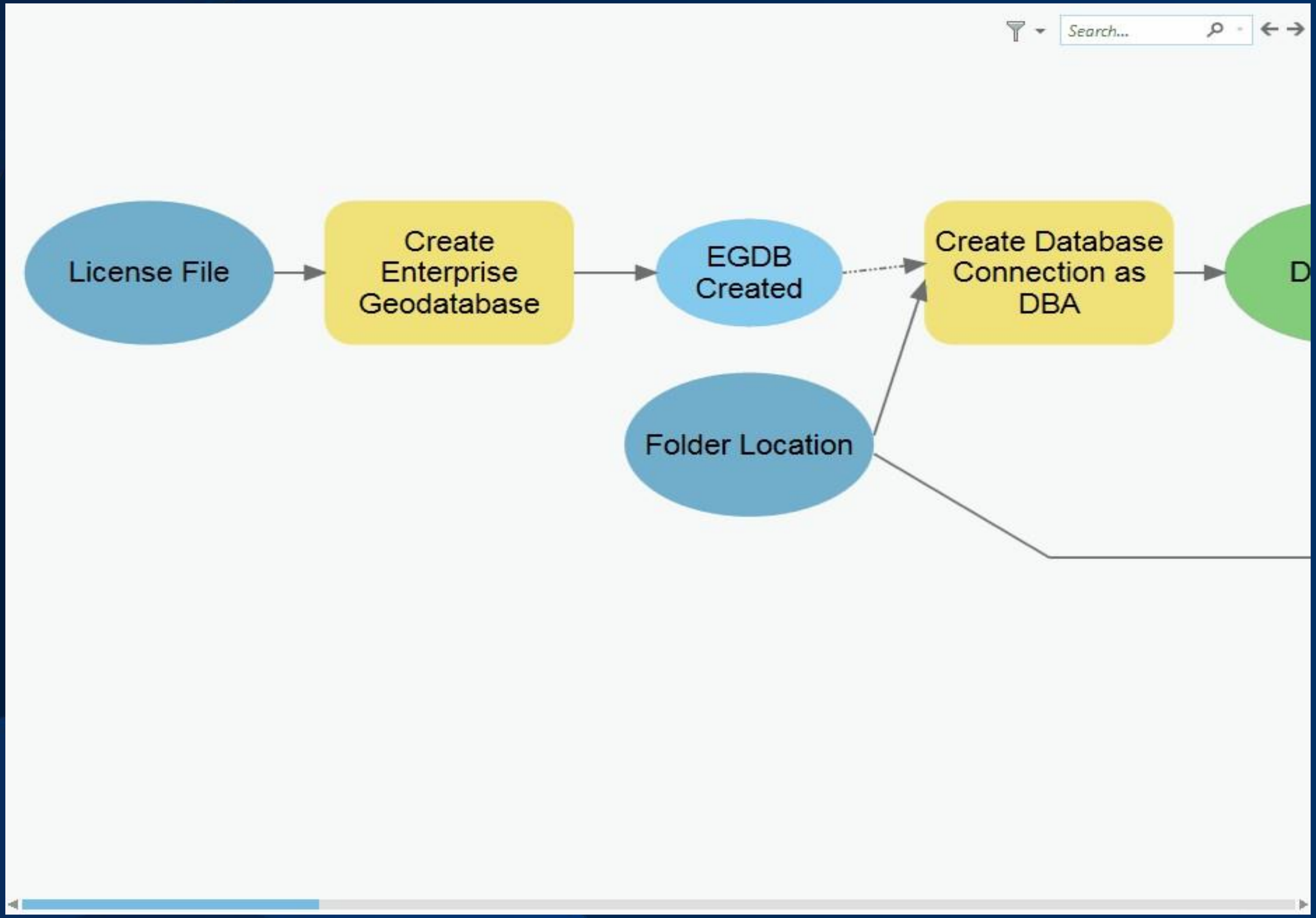
gisworx

Geodatabase Creation using ModelBuilder



gisworx

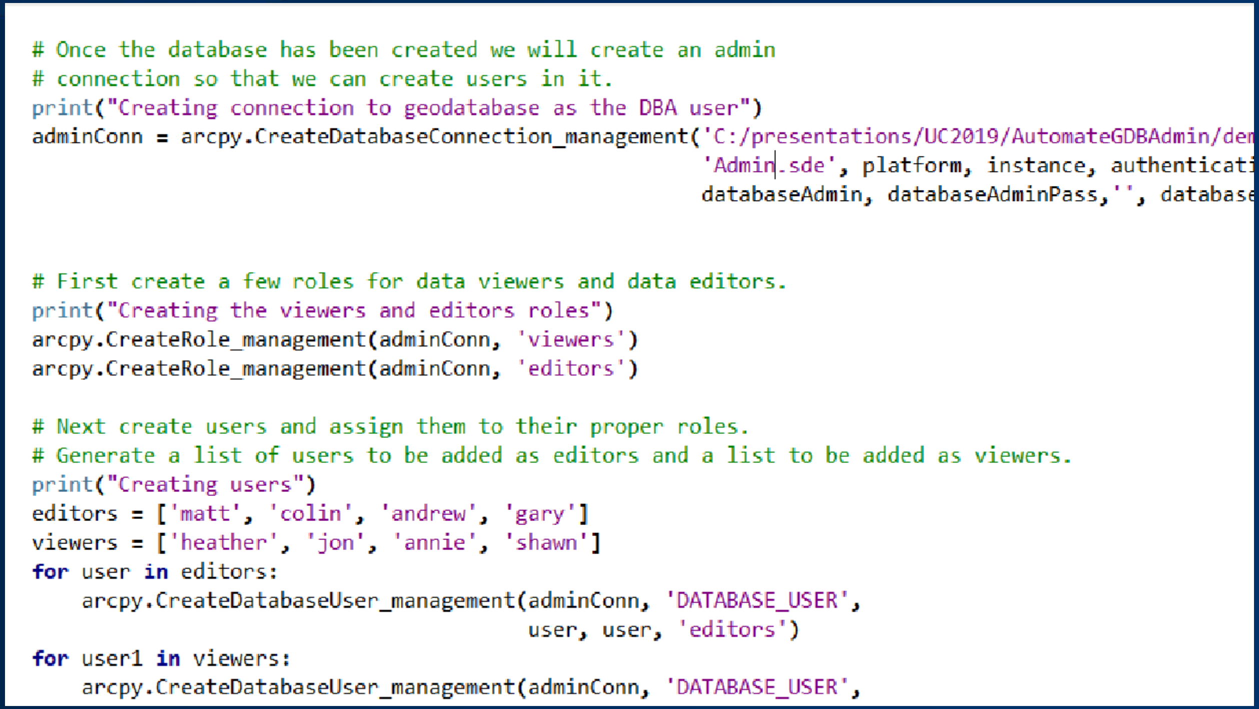
Geodatabase Creation using ModelBuilder


Search...
License File
Create Enterprise Geodatabase
EGDB Created
Create Database Connection as DBA
D
Folder Location

gisworx

Geodatabase Creation

Creating users and roles,
loading data,
setting permissions

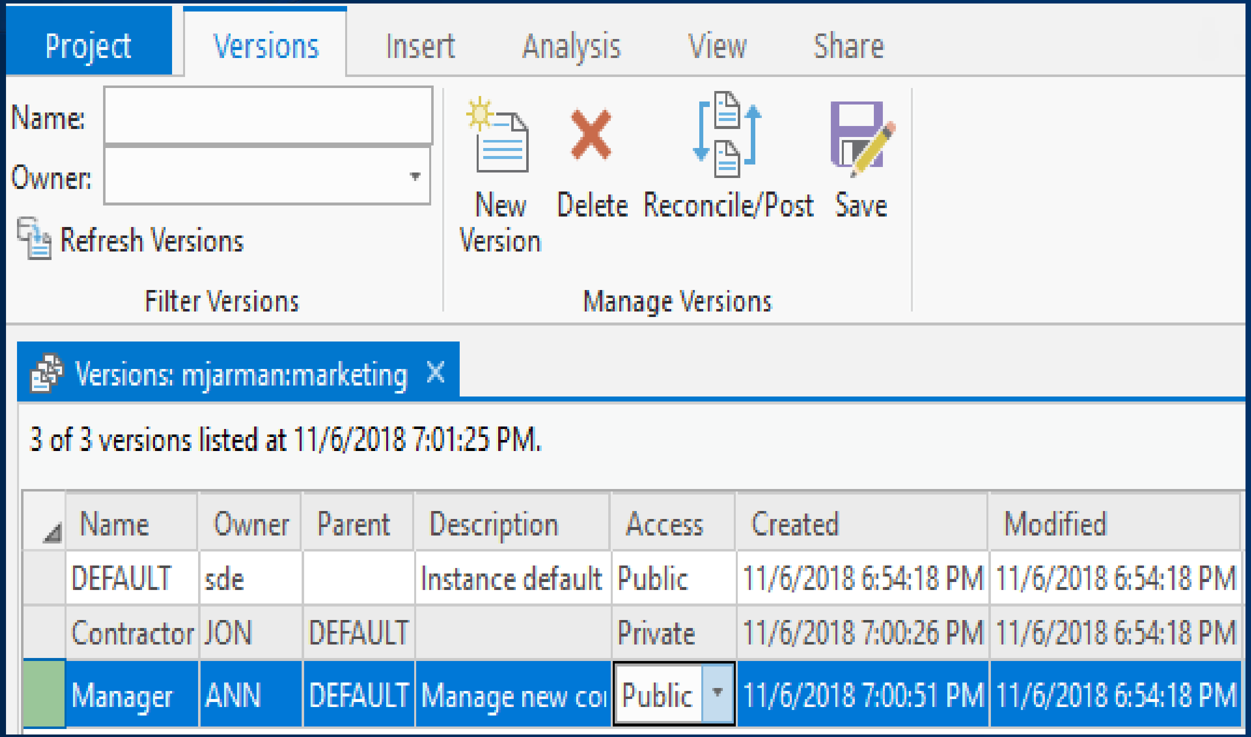


```
# Once the database has been created we will create an admin
# connection so that we can create users in it.
print("Creating connection to geodatabase as the DBA user")
adminConn = arcpy.CreateDatabaseConnection_management('C:/presentations/UC2019/AutomateGDBAdmin/dem
                                                      'Admin.sde', platform, instance, authenticati
                                                      databaseAdmin, databaseAdminPass,'', database


# First create a few roles for data viewers and data editors.
print("Creating the viewers and editors roles")
arcpy.CreateRole_management(adminConn, 'viewers')
arcpy.CreateRole_management(adminConn, 'editors')

# Next create users and assign them to their proper roles.
# Generate a list of users to be added as editors and a list to be added as viewers.
print("Creating users")
editors = ['matt', 'colin', 'andrew', 'gary']
viewers = ['heather', 'jon', 'annie', 'shawn']
for user in editors:
    arcpy.CreateDatabaseUser_management(adminConn, 'DATABASE_USER',
                                        user, user, 'editors')
for user1 in viewers:
    arcpy.CreateDatabaseUser_management(adminConn, 'DATABASE_USER',
```

gisworx

Version Management

Connection management and versioning workflows



3 of 3 versions listed at 11/6/2018 7:01:25 PM.

Name	Owner	Parent	Description	Access	Created	Modified
DEFAULT	sde		Instance default	Public	11/6/2018 6:54:18 PM	11/6/2018 6:54:18 PM
Contractor	JON	DEFAULT		Private	11/6/2018 7:00:26 PM	11/6/2018 6:54:18 PM
Manager	ANN	DEFAULT	Manage new coı	Public	11/6/2018 7:00:51 PM	11/6/2018 6:54:18 PM

gisworx

Managing connections with arcpy functions

Performed by the Geodatabase Administrator (sde user)

Block or allow new connections

- `arcpy.AcceptConnections()`

View connected users and their connection properties

- `arcpy.ListUsers()`

Disconnect users (use caution)

- `arcpy.DisconnectUser()`

Version administration tasks

Performed by several administrators as well as users with editing privileges

Data is registered as versioned

- Register as Versioned GP tool
    - Data Owner

Version created for editors

- Create Version GP tool
    - Database users with permissions on data

Editors connect to a specific version to make edits

- Use GP tools or manual edits in the map
    - Database users with edit permissions on data

Version administration tasks

Performed by Geodatabase Administrator and Data Owner

Reconcile and post

- Reconcile = pull changes from parent to child version
- Post = push reconciled changes from child to parent version
- Reconcile Versions GP tool
  - Automate the process
  - Must define how to deal with conflicts
  - Recommended to run as Geodatabase Administrator

Compress

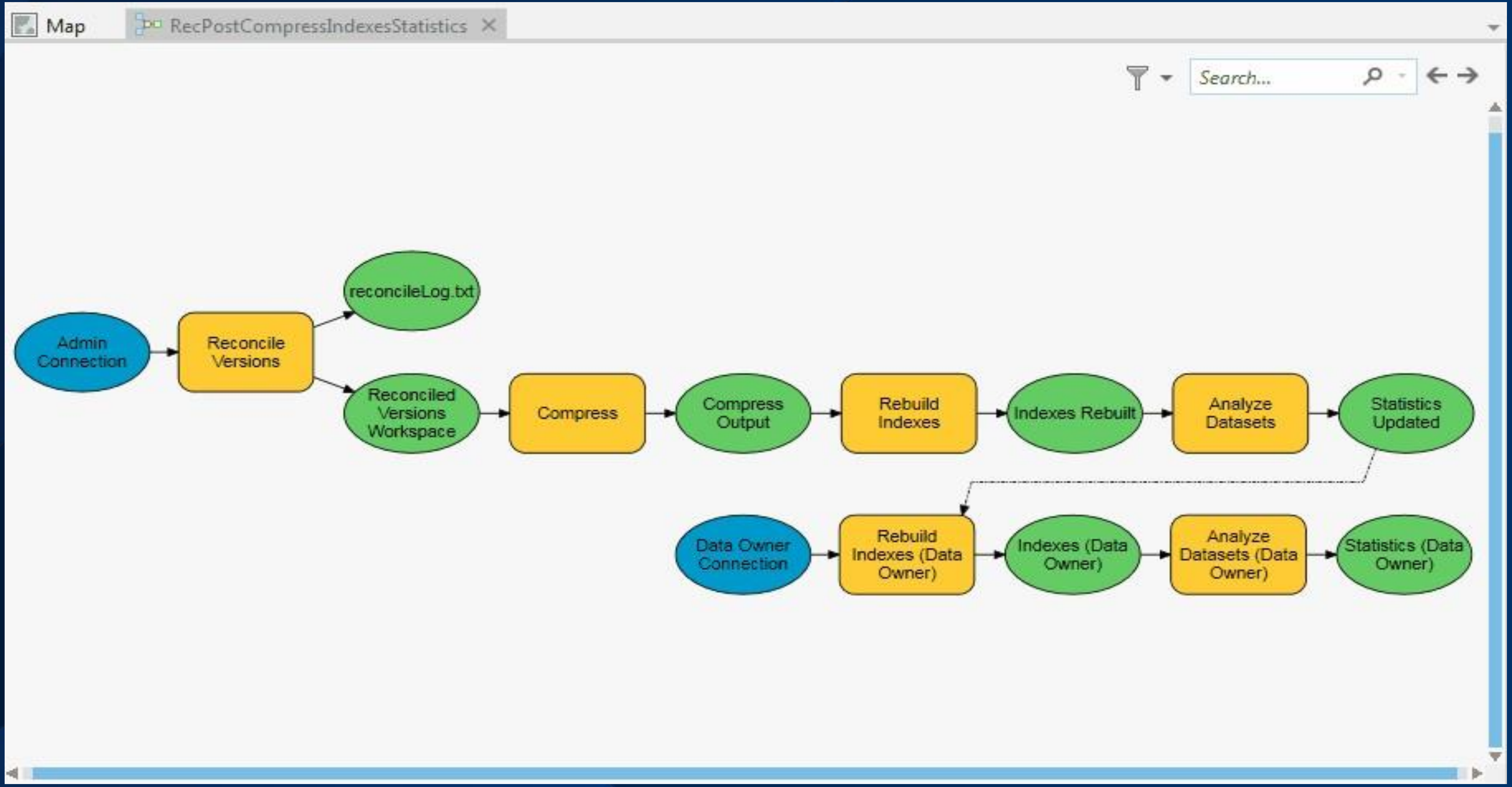
- Compress GP tool
  - Geodatabase Administrator

Update statistics and rebuild indexes (if needed)

- Analyze Datasets GP tool
- Rebuild Indexes GP tool
- Executed by both the Geodatabase Admin and Data Owner

gisworx

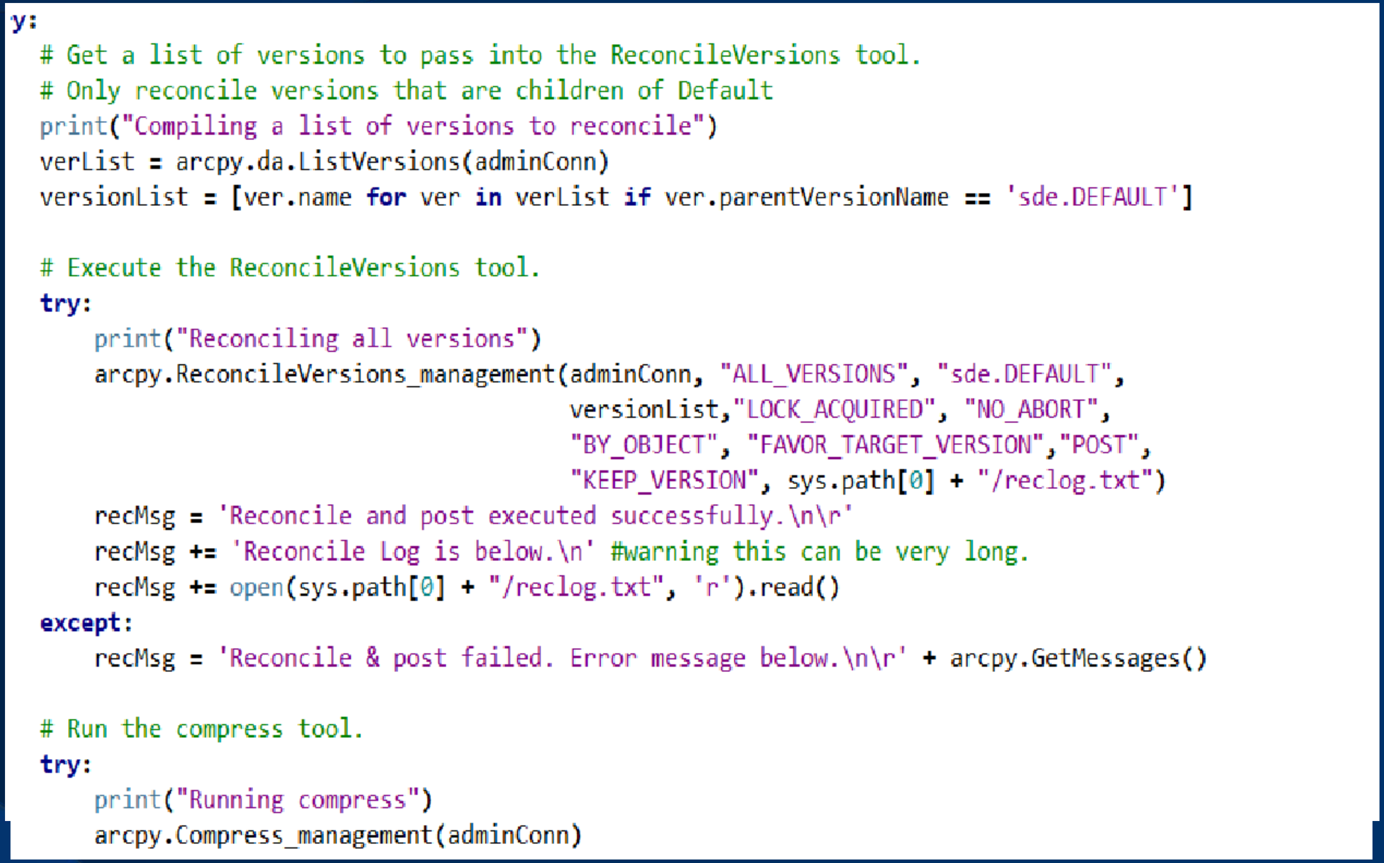
Geodatabase Maintenance using ModelBuilder



gisworx

Version Management

Manage connections & version management tasks



```
y:
  # Get a list of versions to pass into the ReconcileVersions tool.
  # Only reconcile versions that are children of Default
  print("Compiling a list of versions to reconcile")
  verList = arcpy.da.ListVersions(adminConn)
  versionList = [ver.name for ver in verList if ver.parentVersionName == 'sde.DEFAULT']

  # Execute the ReconcileVersions tool.
  try:
      print("Reconciling all versions")
      arcpy.ReconcileVersions_management(adminConn, "ALL_VERSIONS", "sde.DEFAULT",
                                         versionList,"LOCK_ACQUIRED", "NO_ABORT",
                                         "BY_OBJECT", "FAVOR_TARGET_VERSION","POST",
                                         "KEEP_VERSION", sys.path[0] + "/reclog.txt")
      recMsg = 'Reconcile and post executed successfully.\n\r'
      recMsg += 'Reconcile Log is below.\n' #warning this can be very long.
      recMsg += open(sys.path[0] + "/reclog.txt", 'r').read()
  except:
      recMsg = 'Reconcile & post failed. Error message below.\n\r' + arcpy.GetMessages()

  # Run the compress tool.
  try:
      print("Running compress")
      arcpy.Compress_management(adminConn)
```

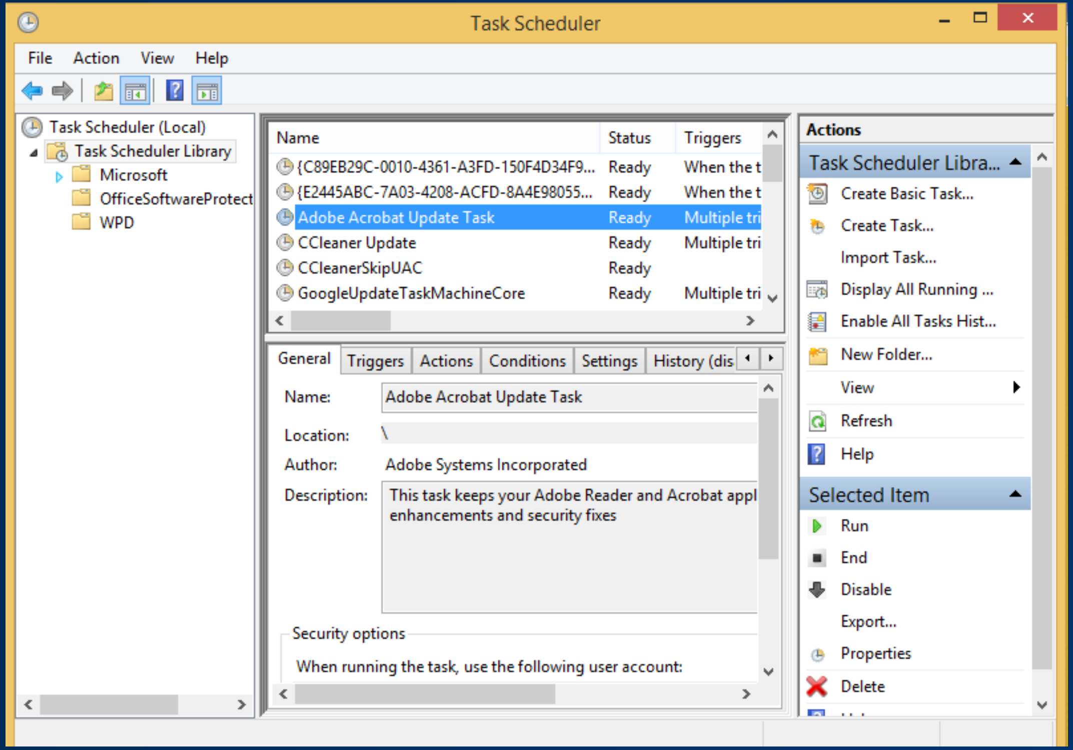
gisworx

Other considerations

- **Ability to run models in Python using arcpy.AddToolbox()**

```
# Import toolbox and run model
arcpy.AddToolbox("C:\\MyToolboxes\\MyToolboxName.tbx")
arcpy.MyModelName_MyToolboxName()
```

- **Use a task scheduler to run scripts overnight**
  - **Windows Task Scheduler**
  - **Linux cron job**



Name	Status	Triggers
{C89EB29C-0010-4361-A3FD-150F4D34F9...	Ready	When the t
{E2445ABC-7A03-4208-ACFD-8A4E98055...	Ready	When the t
Adobe Acrobat Update Task	Ready	Multiple tri
CCleaner Update	Ready	Multiple tri
CCleanerSkipUAC	Ready	
GoogleUpdateTaskMachineCore	Ready	Multiple tri

Name: Adobe Acrobat Update Task
Location: \
Author: Adobe Systems Incorporated
Description: This task keeps your Adobe Reader and Acrobat appl enhancements and security fixes

Security options
When running the task, use the following user account:

gisworx

Summary

You can definitely use Python to automate your admin tasks!

- Various administration functionality available to use
  - For all types of administrators
  - Use where most comfortable
    - Geoprocessing pane
    - ModelBuilder
    - Python

- Setup your geodatabase for multi-user editing

- Connection management

- Geodatabase maintenance tasks

- Use a task scheduler to automate scripts to run

Please Share Your Feedback

gisworx

Thank you!

ashraf.dar@gistec.com

+971-526655107

gisworx