

Administering your Enterprise Geodatabase through Python

Russell Brennan & Jill Penney



Assumptions

- Basic knowledge of python
 - Basic knowledge enterprise geodatabases and workflows
 - You want code
-
- Hold all questions till the end
 - Please turn off or silence cell phones



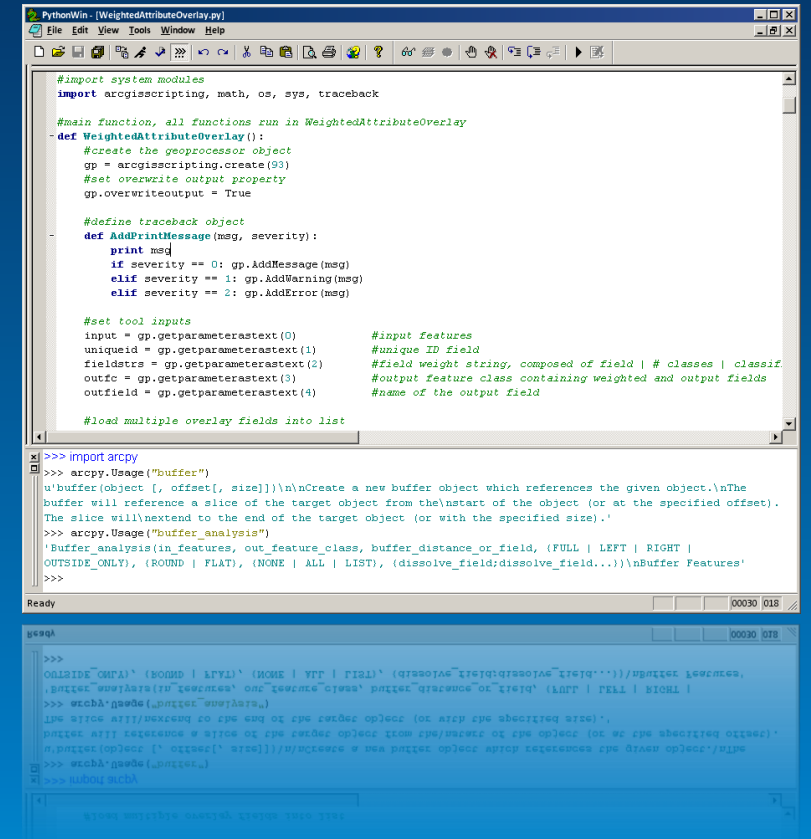
Roadmap

- Session is divided into three parts
- Part 1: What is the geodatabase
- Part 2: Geodatabase Creation
- Part 3: Version Management
- Demos throughout



Why Python

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



The image shows a screenshot of a Python script editor window titled "PythonWin - [WeightedAttributeOverlay.py]". The script defines a class `WeightedAttributeOverlay` with methods for creating a geoprocessor object, adding messages, and setting tool inputs. Below the script, a command prompt window shows the execution of `import arcpy` and the display of `arcpy.Usage("buffer")` and `arcpy.Usage("buffer_analysis")`.

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

#main function, all functions run in WeightedAttributeOverlay
def WeightedAttributeOverlay():
    #create the geoprocessor object
    gp = arcpy.GPToolbox()
    #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)
    uniqueid = gp.getparameterastext(1)
    fieldstrs = gp.getparameterastext(2)
    outfc = gp.getparameterastext(3)
    outfield = gp.getparameterastext(4)

    #load multiple overlay fields into list

    #input features
    #unique ID field
    #field weight string, composed of field | # classes | classif
    #output feature class containing weighted and output fields
    #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.\n\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'
>>>
```

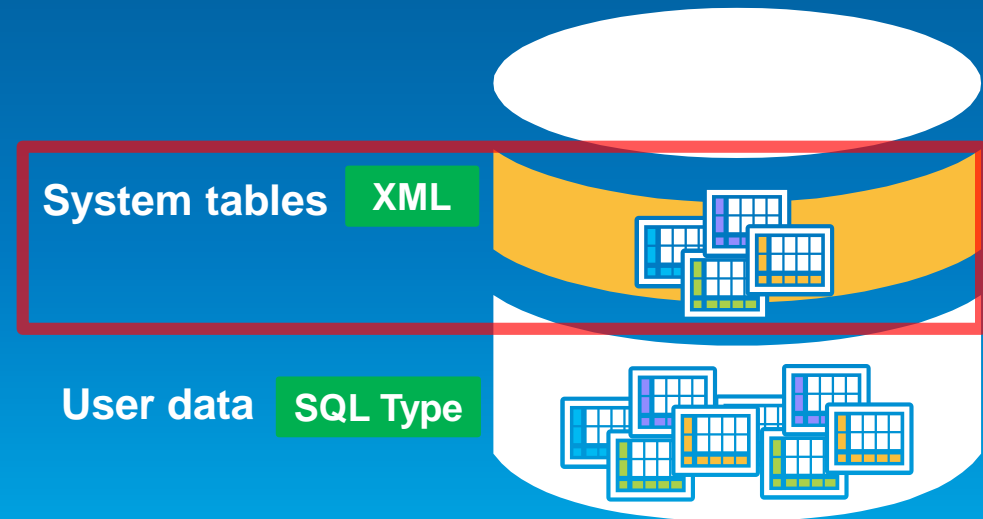
What is the Geodatabase

- **A physical store of geographic data**
 - Scalable storage model supported on different platforms
- **Core ArcGIS information model**
 - A comprehensive model for representing and managing GIS data
 - Implemented as a series of simple tables
- **A transactional model for managing GIS workflows**
- **Set of components for accessing data**



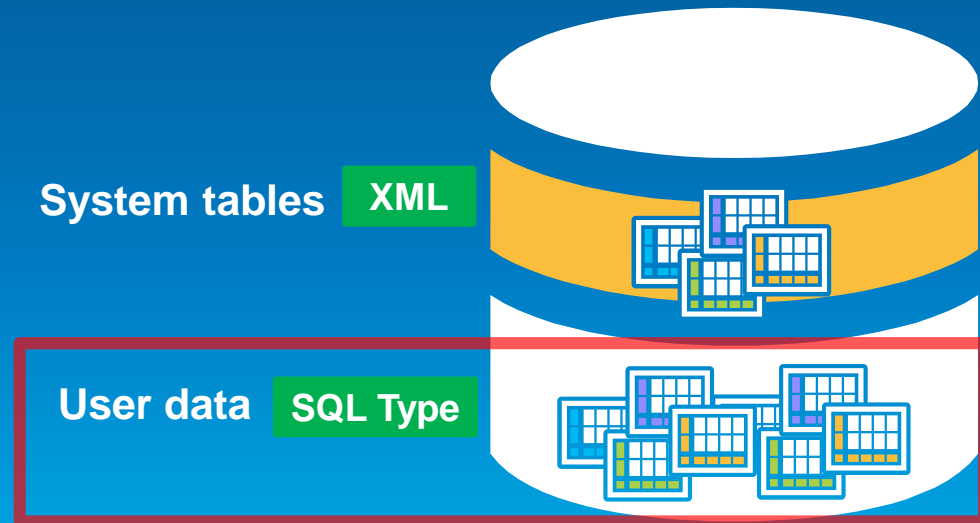
Geodatabase system tables

- System tables store definitions, rules, and behavior for datasets
- Tracks contents within a geodatabase
- Stores some database level metadata
 - Versions, domains, etc.
- Admin operations:
 - Version management
 - Connection management
 - Geodatabase upgrade



User-defined tables

- Stores the content of each dataset in the geodatabase
 - Datasets are stored in one or more tables
- Administrative Operations:
 - Granting/revoking privileges
 - Updating statistics/indexes
 - Registering as versioned
 - Adding global id's
 - Enabling editor tracking



Types of administrators

- Database administrator (DBA)
- Geodatabase administrator
- Data owner (aka dataset administrator)
- May or may not be the same person.



Types of administrators

Database

- Database administrator (DBA)
- Instance level admin



Types of administrators

Geodatabase

- Geodatabase administrator
- Owns the geodatabase repository



Types of administrators

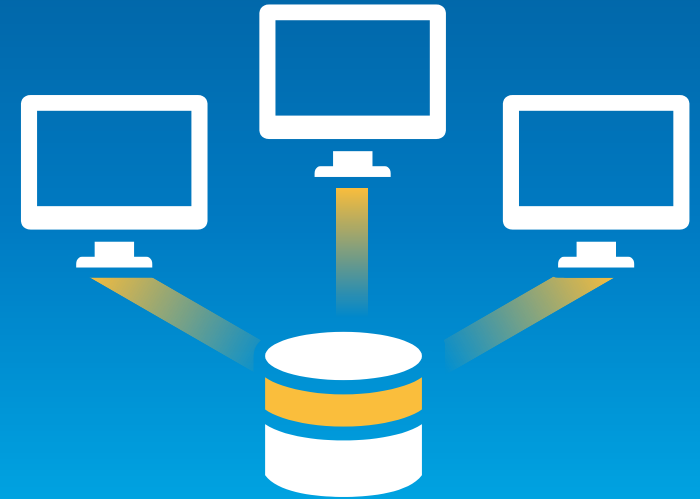
Data Owner

- **Dataset administrator**
 - Granting privileges to data
 - Modifying schema of data
 - Database statistics and index maintenance



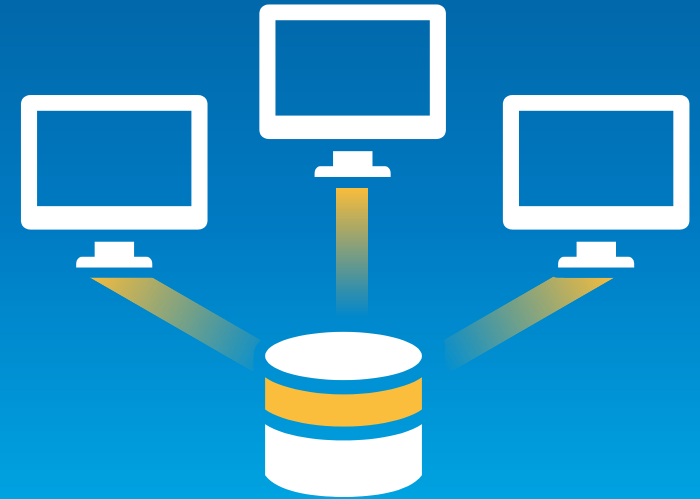
Connections

- A connection file is needed to access an enterprise geodatabase
- Control how you are connected to the database
- What user is connected
 - Creating data
 - Schema changes
 - Administering the geodatabase
- Instance/database you are connected to
- Version, historical archive or moment in time
- Changing properties requires new connection files.



Connection Tools

- **Create database connection**
 - Output '.sde' file
 - Can connect to both databases and geodatabases
- **Create ArcSDE connection file**
 - Only way to create 3-tier connections
 - Will be deprecated.



Demo 1

Creating Connections



Geodatabase Creation

with Python

Enabling geodatabase behavior

- You already have an existing database
- Enable Enterprise Geodatabase
 - From an existing database you can enable geodatabase functionality
 - Lays down geodatabase repository
 - GDB tables and stored procedures
- Must connect as appropriate user.



Creating Enterprise Geodatabases

- Create Enterprise Geodatabase tool
- When you have a need for:
 - Creating testing or development environments
 - Database does not already exist
- Run as DBA



Creating Users

- Create database user tool
- Creates a user in enterprise geodatabase or database
- Cannot create geodatabase administrative user
- DBA not geodatabase admin.



Creating Database Roles

- Makes it easier to assign/revoke privileges to a group
- Prior to creating users in the geodatabase
- When creating users you can assign them to a role
- DBA not geodatabase admin.



How to Create a Geodatabase



Creating and Loading Data

- Numerous tools for creating any type of data:
 - Create table, Create feature class, Create Raster Dataset, etc.
 - Create Geometric Network, Create Topology, Create Domain, etc.
- Also tools for loading data:
 - Feature class to feature class (single)
 - Feature class to geodatabase (multiple)
 - Import XML workspace
- Write a custom script



Configuration Keywords

- New tool at 10.3 for managing keywords
- Export and import
 - Export Geodatabase Configuration Keyword tool
 - Import Geodatabase Configuration Keyword tool
- Workflow
 - Export
 - Edit with text editor
 - Import



Managing privileges

- Allow other users of the geodatabase to view or edit data that you own
- Change privileges tool
- Allows multiple input datasets to be passed in
- Grant view only or view and edit
 - View = select
 - Edit = insert, update, delete
- Must be connected as data owner



Putting it all together

Geodatabase Creation



Version administration

Parts of the version administration workflow

- Reconciling/posting/compressing
- Updating statistics and indexes on system tables
- Updating statistics and indexes on user data tables
- Managing user connections



Disconnecting user connections

- A user who is connected but has gone home
- Create a cold backup of the database
- Running large queries that are using up resources
- Reconcile/post/compress process (optional)



Managing user connections

- **Block/allow connections**
 - `arcpy.AcceptConnections`
 - Provide boolean
- **Finding connected users**
 - `arcpy.ListUsers`
 - Returns a tuple of properties for each connected user
 - ID, name, machine name, connection time, connection type
- **Disconnecting users**
 - `arcpy.DisconnectUser`
 - Use ids provided from listusers function or use 'ALL' keyword



Demo 4

Connection Management



Reconciling and Posting Versions

- Reconcile = pulling changes from a parent to child version
- Post = pushing reconciled changes to parent version from child
- Reconcile versions tool
- Recommended to run as geodatabase administrator
 - Can 'see' and reconcile all versions in the geodatabase



Indexes and Statistics

- Update after major 'data change' events
 - Reconcile
 - Compress
 - Appending data
 - Typically not necessary after loading new data
- Can be done by both:
 - Geodatabase Admin (system tables)
 - Data owner (data tables)
- We suggest to run regularly.



Demo 5

Version Maintenance



Summary



Thanks for attending

Please fill out session surveys

www.esri.com/RateMyDevSummitSession

Administering Your Enterprise
Geodatabase through Python



Slides and Code

esriurl.com/PythonAdminDS2015





Understanding our world.