

# Automating Workflows with Models and Scripts



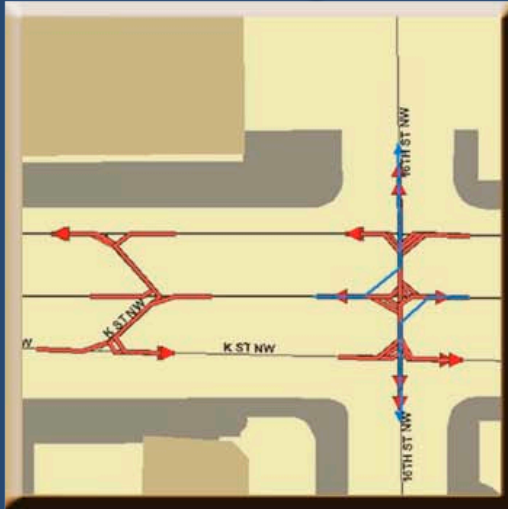
# Agenda

- Overview of geoprocessing framework
- Overview of ModelBuilder
- Automating workflows with ModelBuilder
- Overview of geoprocessing scripting
- Integrating with 3rd party applications
- Sharing your models and tools
- Resources

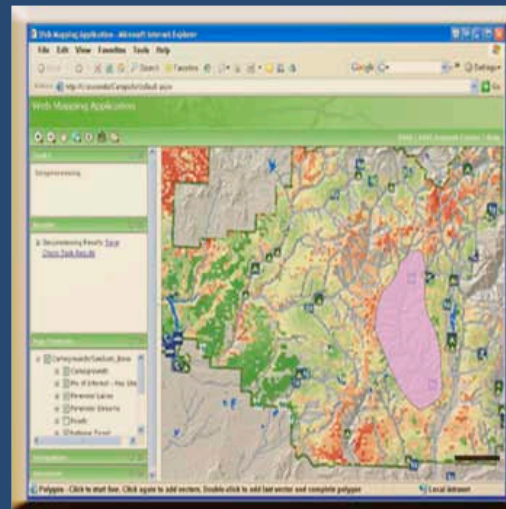
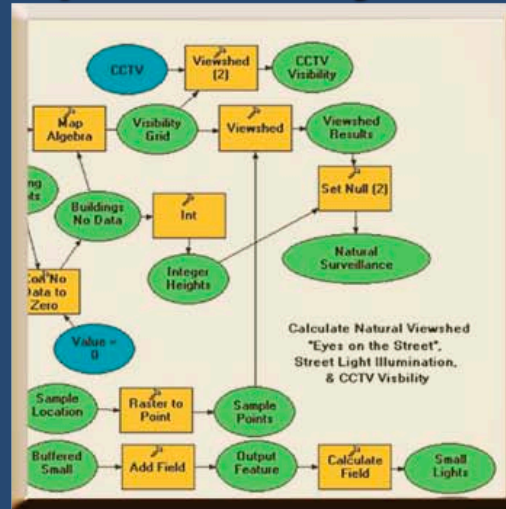
# Geoprocessing is...

...One of the four critical components of GIS

## Data Management



## Spatial Analysis

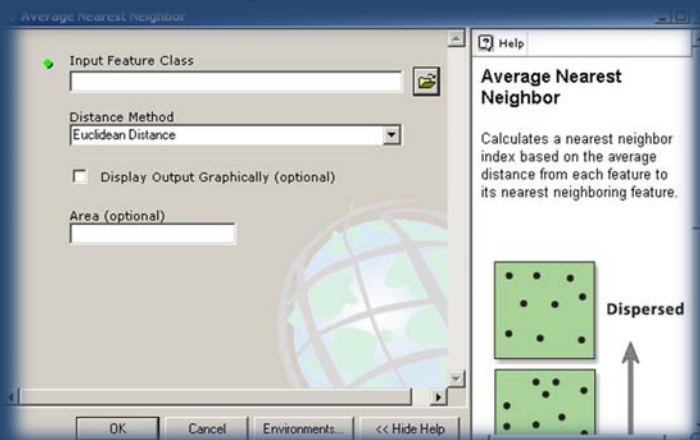


## Visualization

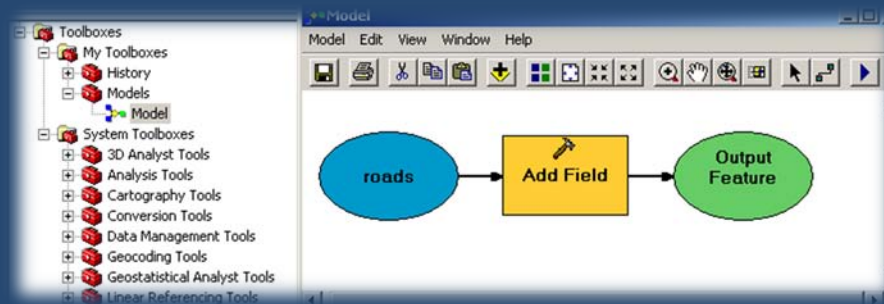
## Dissemination

# Geoprocessing framework

As a dialog...



As a model...



As a script...

As a command...



```
hillshade_tool_input.py - C:\data\GP_scripts\hillshade_tool_input.py
File Edit Format Run Options Windows Help

# Create the Geoprocessor object
gp = arcgisscripting.create()
gp.OverWriteOutput = 1

# Check out any necessary licenses
gp.CheckOutExtension("Spatial")

# Load required toolboxes...
gp.AddToolbox("C:/Program Files/ArcGIS/ArcToolbox/Toolboxes/Spatial Analy...

# Local variables...
Output_raster = "C:\data\GP_scripts\hillshade"

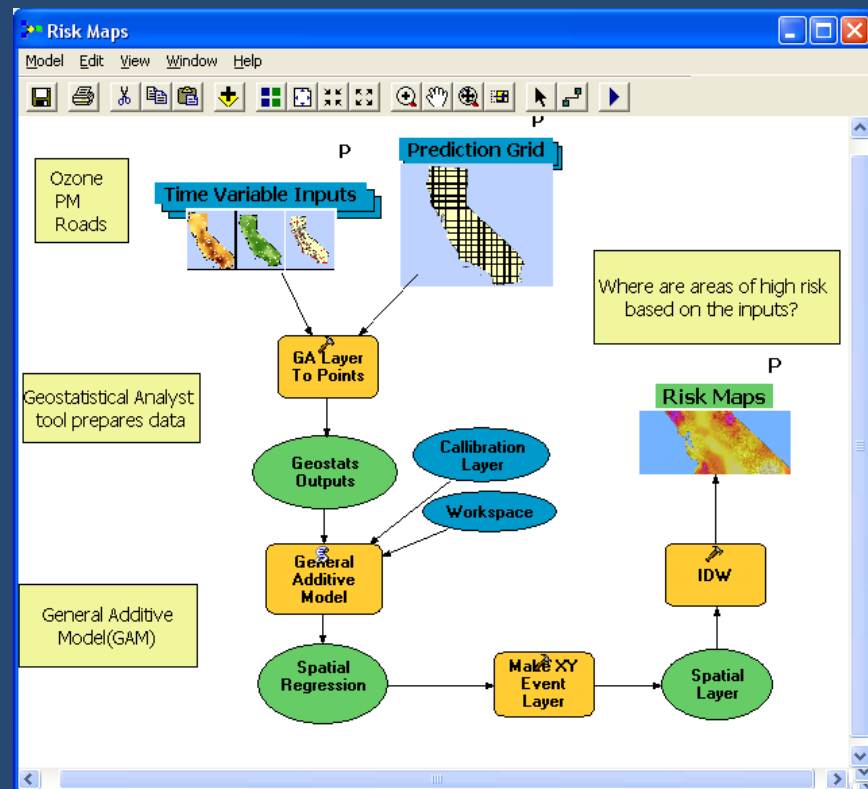
elevfloat = sys.argv[1]

print "\n Input raster: " + elevfloat
print "\n Output raster: " + Output_raster

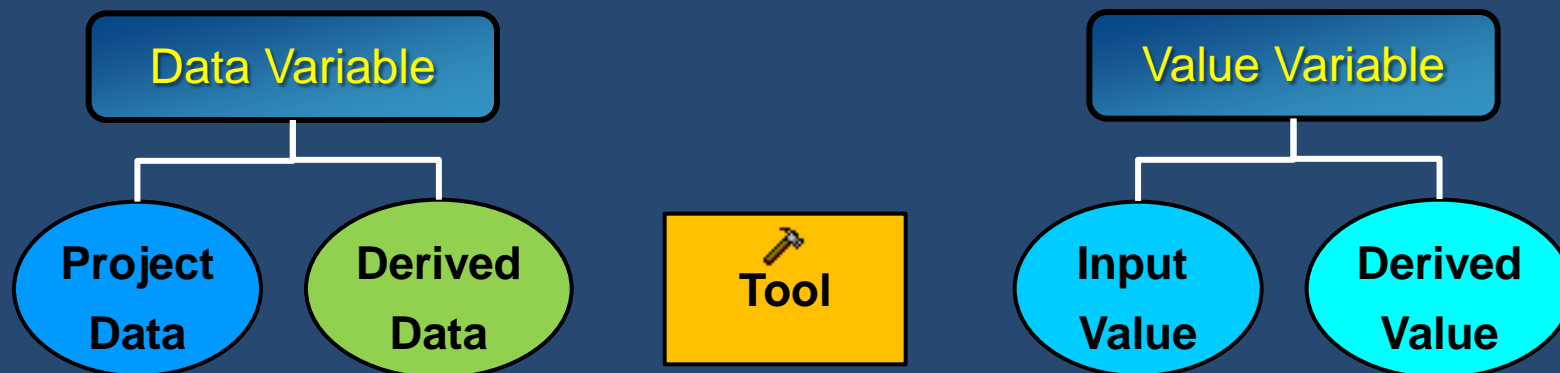
# Process: Hillshade...
hillshade --in=Input_raster --out=Output_raster --elevation=elevfloat --...
```

# Overview of ModelBuilder

- Graphical programming interface
- Self-documenting and sharable
- Save long processing sequences and rerun them easily
- Sophisticated conditional processing
  - If/then
  - Iterations




# ModelBuilder Review



- **Tools**

- Drag and drop into window
- Use Add button 

- **Data**

- Drag and drop into window
- Drag and drop onto tools
- Fill in dialog
- Use Add button 

- **ArcGIS Desktop Help > Geoprocessing > Automating Your Workflows > Using ModelBuilder > Working with Variables**

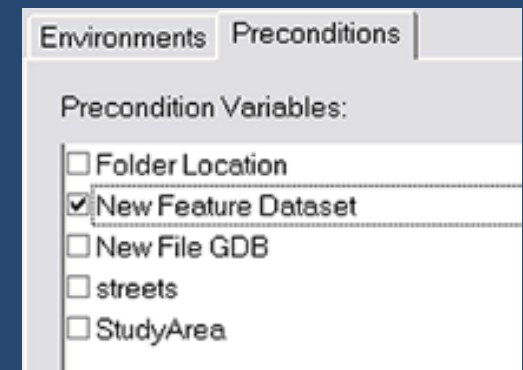
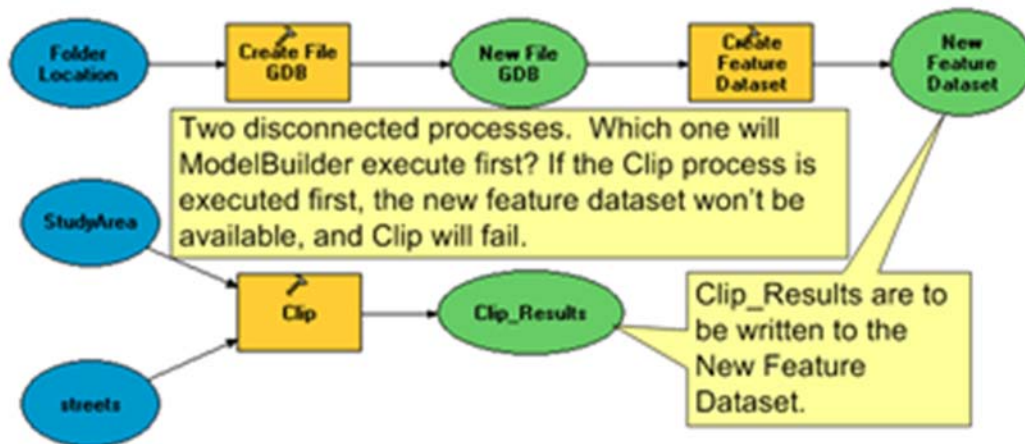
# Demonstration – ModelBuilder Review

- Add a tool to ModelBuilder
- Connect tool's output to another tool
- Populate input and output variables
- Make input and output variables parameters
- Run the model as a tool

# Conditional Processing Workflows

## Preconditions

- Workflows may require processes to be run in certain order
- Any variable (including input and output variables) can be used as *precondition*



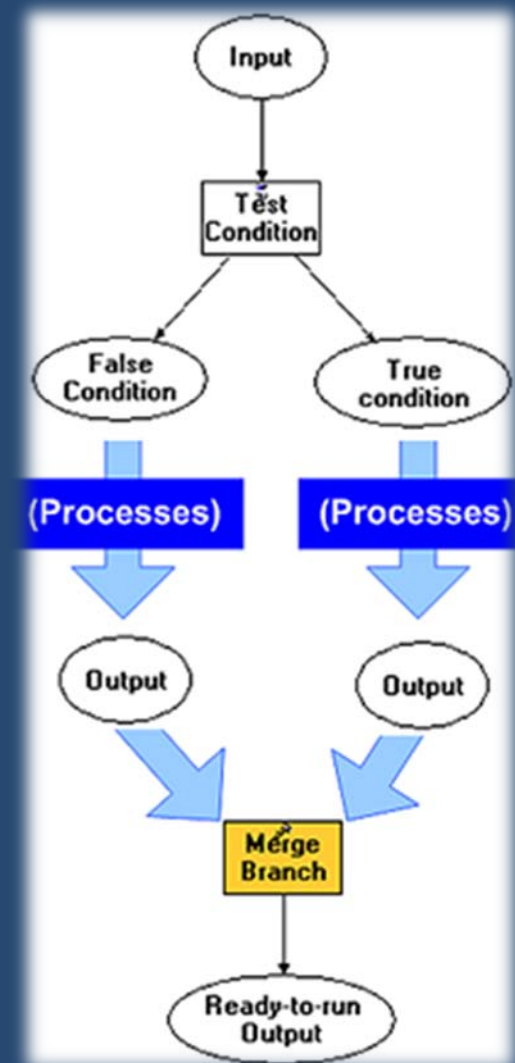


# Conditional Processing Workflows

## Branching

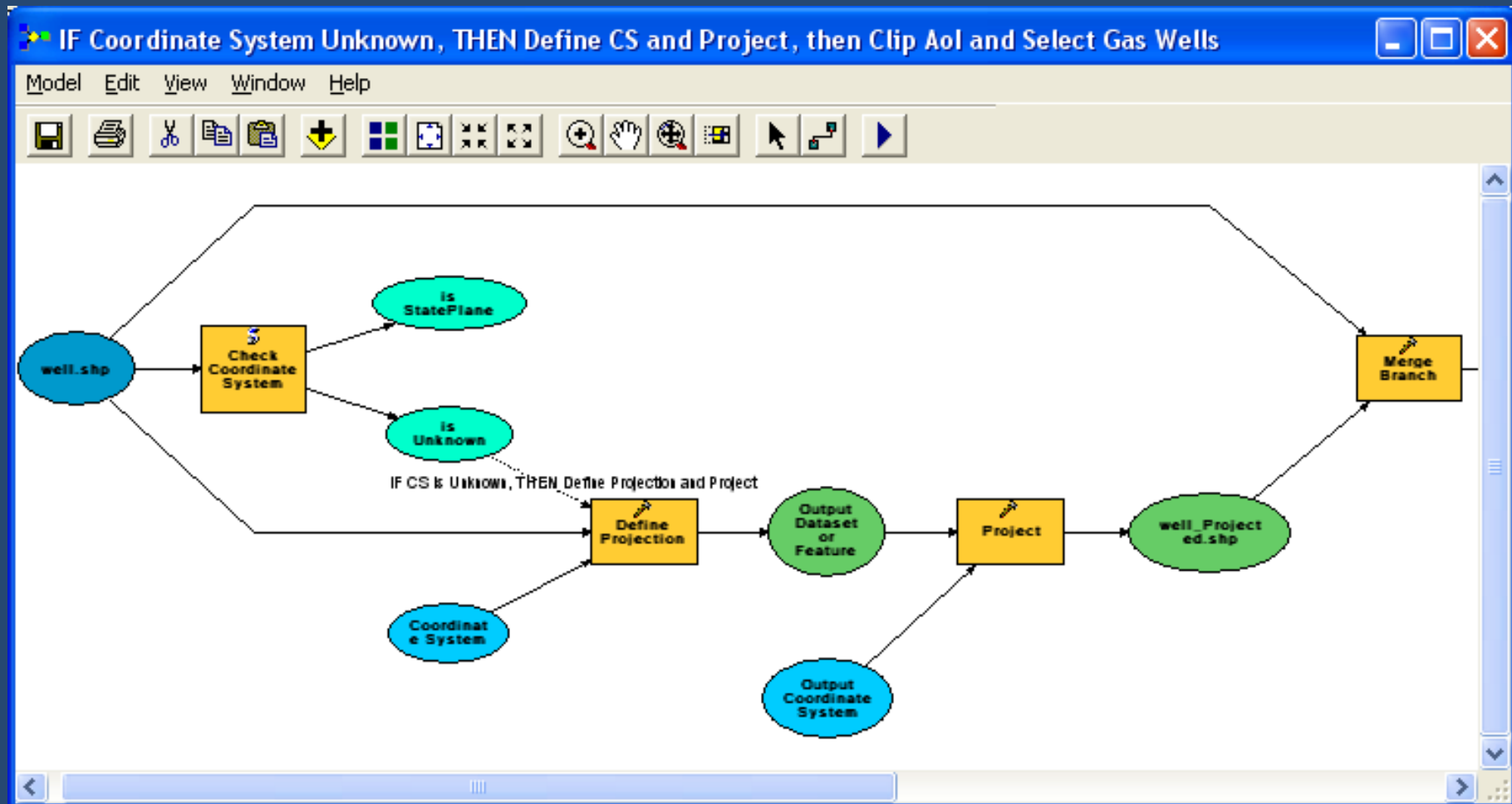
- Many workflows require branching
  - If input is point, run this process
  - If input is polygon, run that process
- If-then-else logic requires a script
  - many example scripts exist
- Use geoprocessing tool for merging branches

**ArcGIS Desktop Help > Geoprocessing > Automating Workflows with Model > Branching: Implementing if-then-else logic**



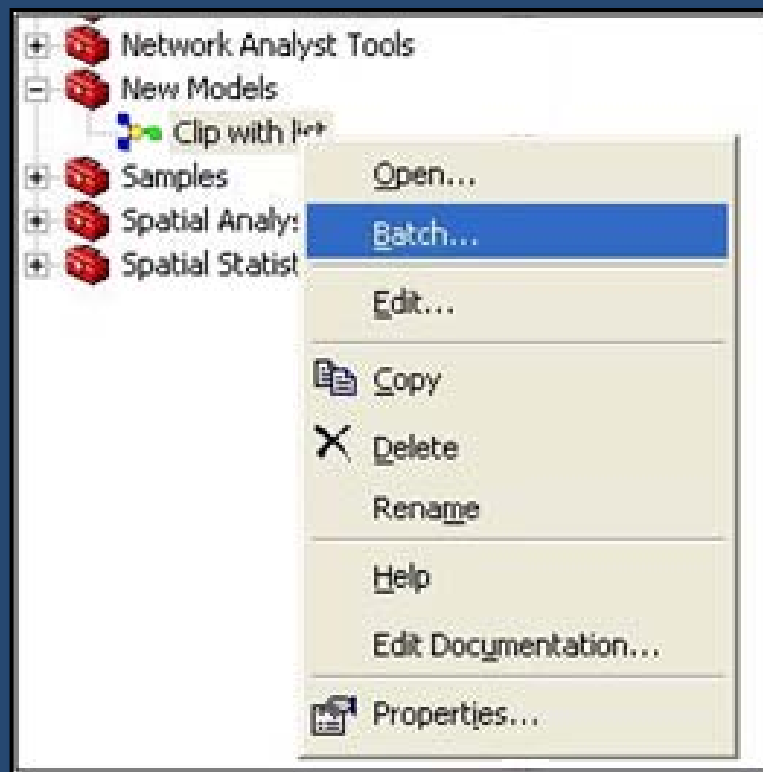
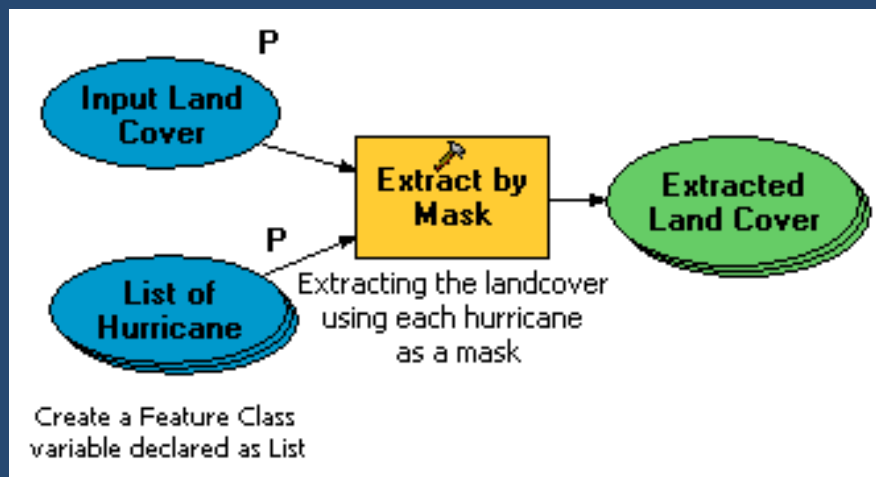
# Conditional Processing Workflows

## Branching Example



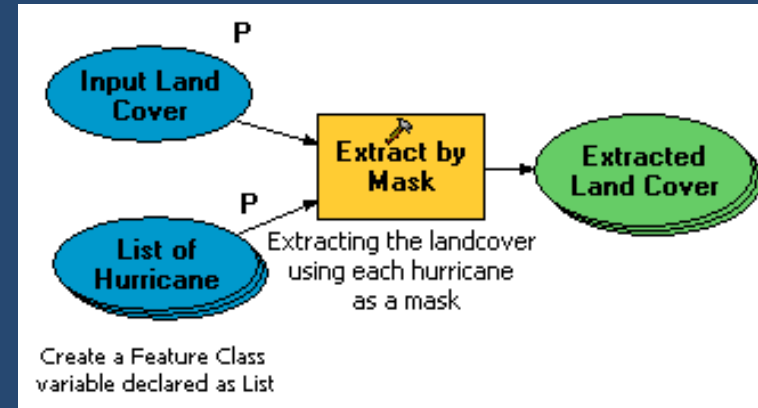
# Executing Model for Multiple Inputs

- By default, a model runs once for a set of inputs
- Several options to execute model multiple times
  - Batch processing
  - List processing
  - Series processing
  - Conditional/Boolean processing



# Variable Substitution

- Iterations may create many outputs
  - Clip land area affected by hurricanes
  - Each output must have unique name
- Commonly used variables
  - %workspace%
  - %extent%
- Built-in keywords for keeping track of iterations
  - %n% is the current iteration number
  - %i% is the current list index
  - %v% is the current series value

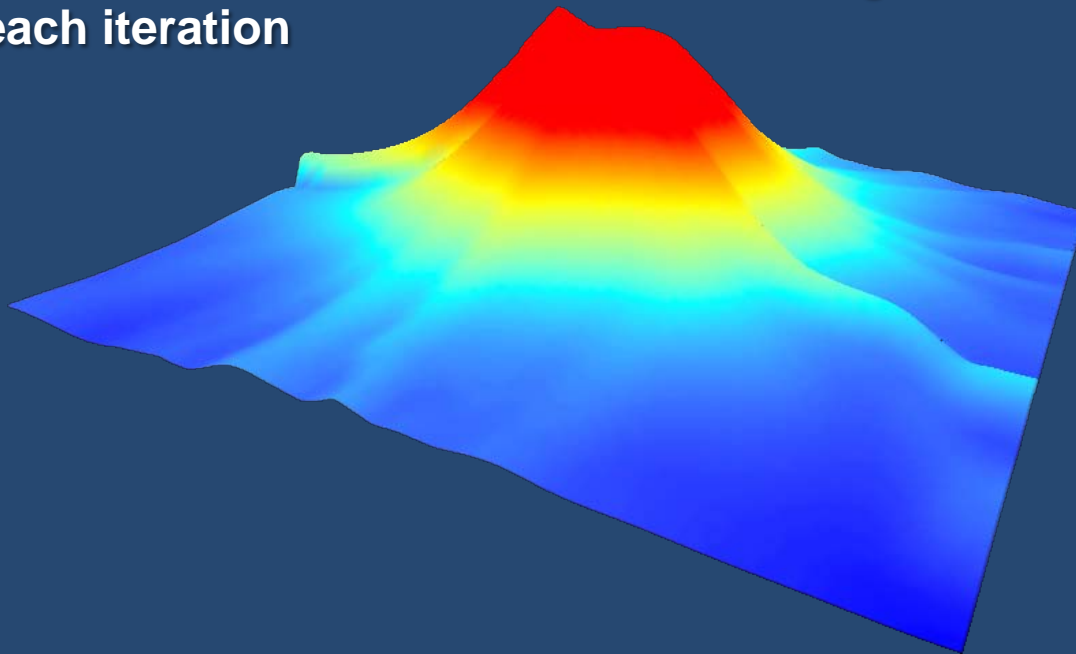


**ArcGIS Desktop Help > Geoprocessing > Automating Your Work with Models > Using ModelBuilder > Model Iteration**

# Demonstration - simulate creation of a volcano

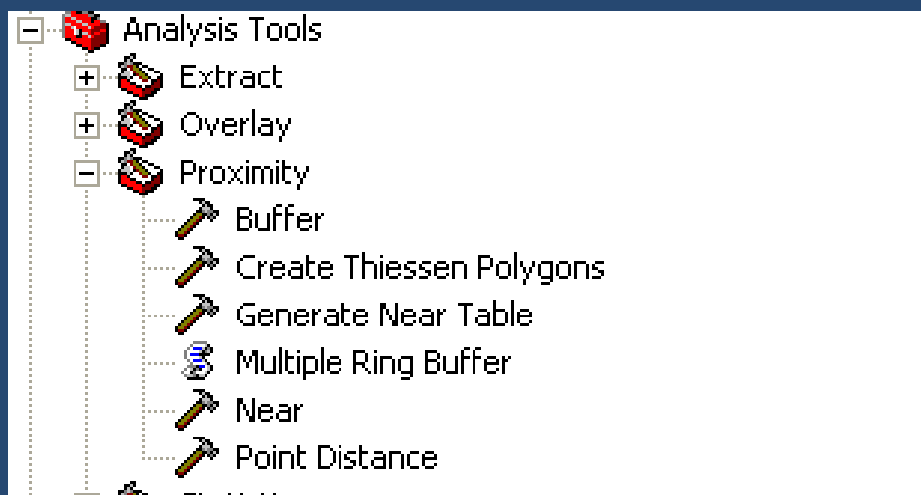
- **Techniques**

- Use model iteration
- Use feedback in the model to build on existing layers
- Use in-line variable substitution to preserve outputs for each iteration
- Use randomization tools of ArcGIS to generate random inputs for each iteration



# Geoprocessing Scripts

- **Why create geoprocessing scripts?**
  - Schedule your model to run at specific dates and times
  - Integrate with any 3rd party software that supports scripting
  - Let your existing script take advantage of geoprocessing framework
  - Simply because you prefer scripting to ModelBuilder
- **Many geoprocessing scripts are part of System Toolboxes**

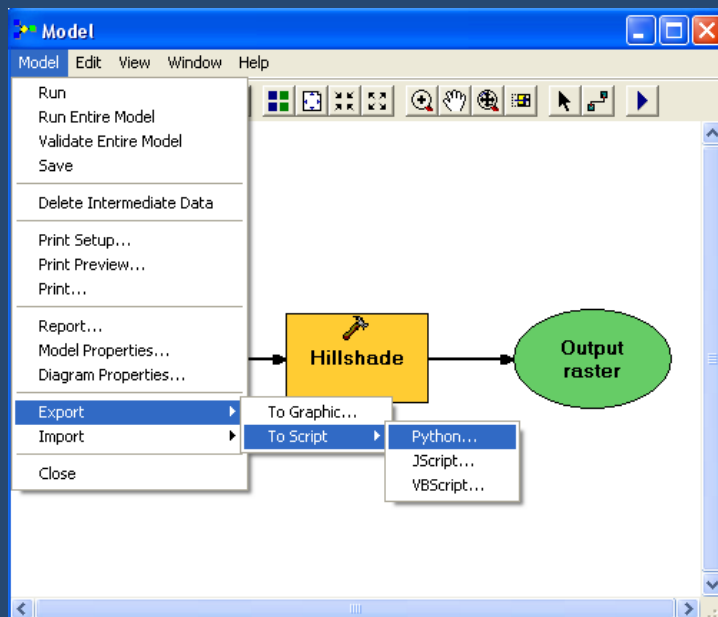


# Examples of Geoprocessing Scripts

- **Create hillshade with various azimuth and altitude values from a single DEM**
- **Determine best location for new monitoring station**
  - add candidate locations and recalculate prediction maps with new input
- **Create geometries from CSV files**
- **Epidemiological modeling through integration between ArcGIS and R statistical package**

# Get Started Writing Geoprocessing Scripts

- Export Geoprocessing Model to Script
- Model may require some changes to run successfully as script
  - E.g., ArcMap layers not available to stand-alone script



```
74 hillshade_tool.py - C:\Documents and Settings\katj3468.VOSTOKD
File Edit Format Run Options Windows Help
# -----
# hillshade_tool.py
# Created on: Wed Dec 31 2008 11:49:05 AM
# (generated by ArcGIS/ModelBuilder)
# -----
# Import system modules
import sys, string, os, arcgisscripting

# Create the Geoprocessor object
gp = arcgisscripting.create()

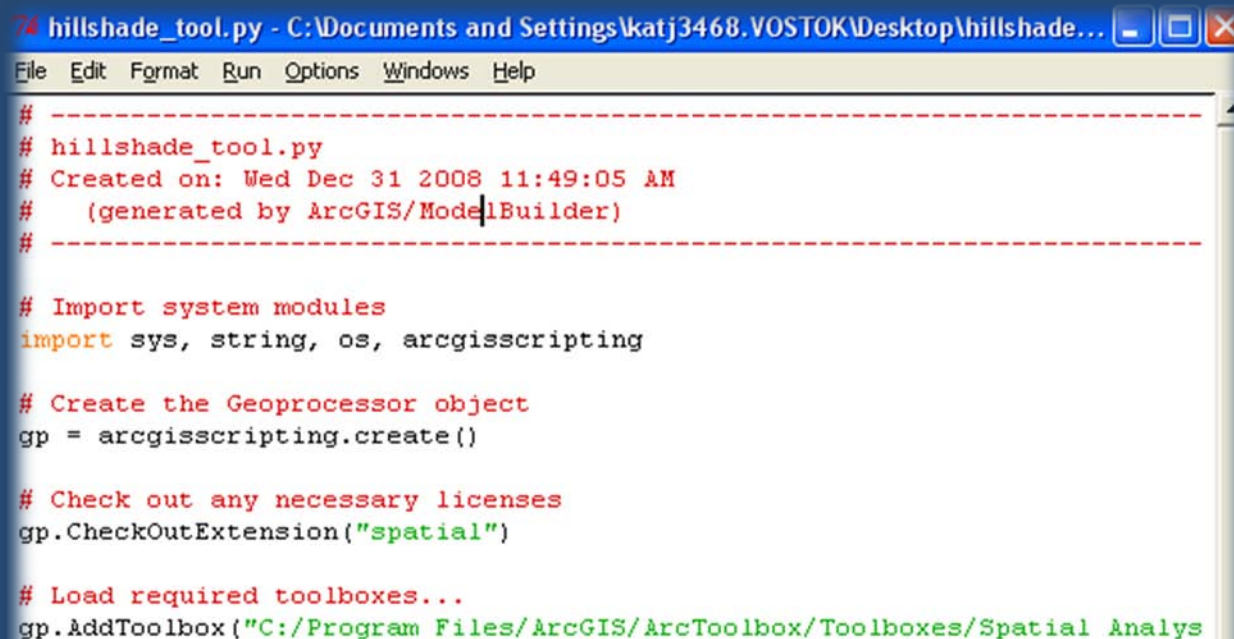
# Check out any necessary licenses
gp.CheckOutExtension("spatial")

# Load required toolboxes...
```



# Demonstration – Export Model to Script

- Create geoprocessing model from Hillshade tool
- Export model to script
- Run stand-alone script
  - Correct script errors to ensure successful execution as stand-alone script



```
74 hillshade_tool.py - C:\Documents and Settings\katj3468.VOSTOK\Desktop\hillshade...
File Edit Format Run Options Windows Help
# -----
# hillshade_tool.py
# Created on: Wed Dec 31 2008 11:49:05 AM
# (generated by ArcGIS/ModelBuilder)
# -----

# Import system modules
import sys, string, os, arcgisscripting

# Create the Geoprocessor object
gp = arcgisscripting.create()

# Check out any necessary licenses
gp.CheckOutExtension("spatial")

# Load required toolboxes...
gp.AddToolbox("C:/Program Files/ArcGIS/ArcToolbox/Toolboxes/Spatial Analys
```

# Scripting Languages

- **Many to choose from for geoprocessing**
  - Python, VBScript, JScript
  - Others
- **ESRI Recommends Python**
  - “It’s really good”
  - Modular
  - Object-oriented
  - Easy to maintain
  - Scalable
  - Cross platform (windows & UNIX/Linux)
  - Established and active user community
  - Most geoprocessing examples are available in Python



# A Little About Python

- Where to write code
  - Text editors (e.g. Notepad)
  - Python Command line
  - IDEs (e.g. PythonWin or IDLE)
- Variables in Python
  - No declaration or type assignment
  - Can hold strings, numbers, lists, files, etc.
- Import system or custom modules

```
>>> import math
>>> print math.sqrt(9)
3.0
>>> |
```



# Tips for Initializing the Geoprocessor

- **Version 9.2 and above does not require PythonWin**
  - win32 can be installed separately if needed for other modules
- **ArcGIS 9.3 can initialize both 9.2-style and 9.3-style geoprocessor**
  - to initialize 9.3-style geoprocessor, specify “9.3” as argument

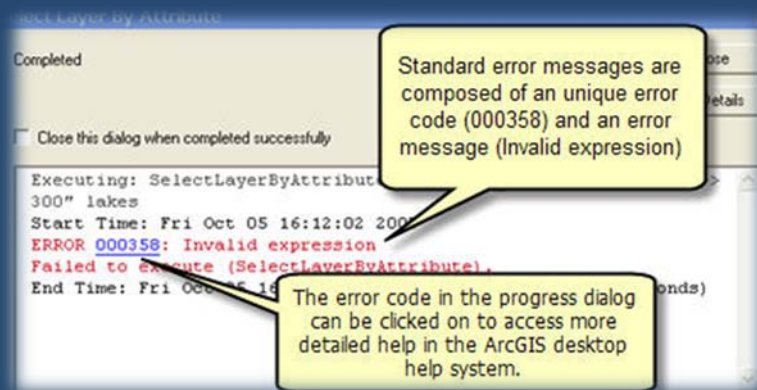
```
# creates the geoprocessing object
import arcgisscripting

Gp = arcgisscripting.create()    # 9.2-style geoprocessor
Gp = arcgisscripting.create(9.3) # 9.3-style geoprocessor

# creates the geoprocessing object
#PythonWin required, “old” method
import win32com.client
Gp = win32com.client.Dispatch("esriGeoprocessing.GpDispatch.1")
```

# Getting Messages

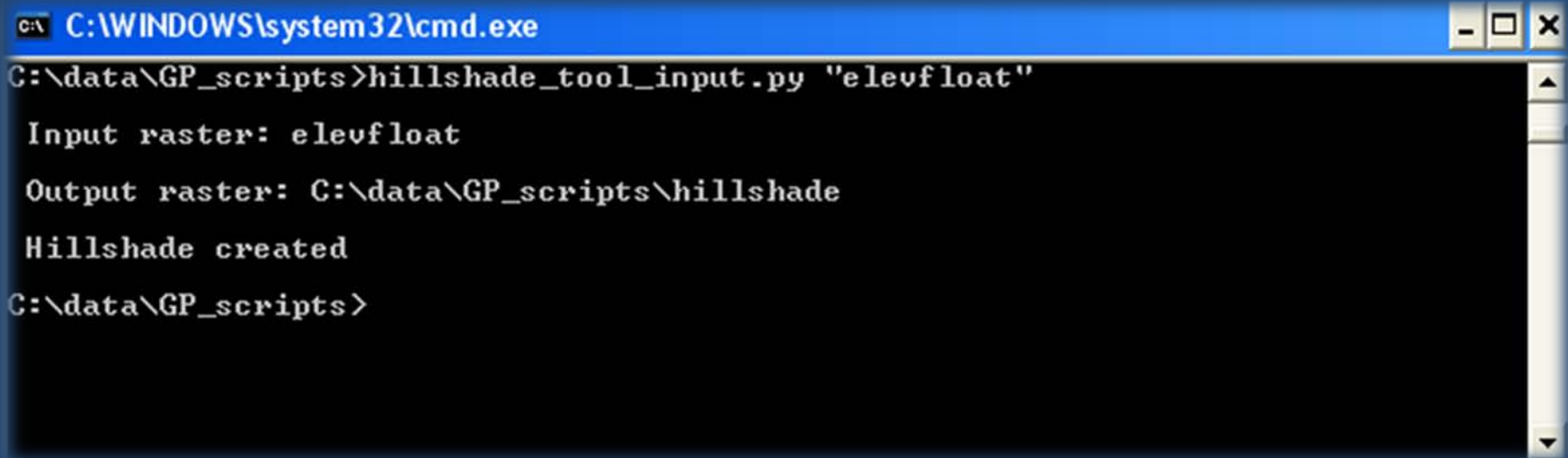
- **GetMessages** returns messages for the specified severity
  - Informative (0), warning (1), error (2)
- **GetMessage** returns message using its index number
  - Index is zero-based
- **GetIDMessage** will return standard system error message by ID



**ArcGIS Desktop Help > Geoprocessing > Automating Workflows with Scripts > Accessing Tools with a Geoprocessing Script > Error Handling With Python**

# Stand-alone Scripts

- Can be run independent of ArcGIS applications
- Can be scheduled to run daily/weekly, anytime

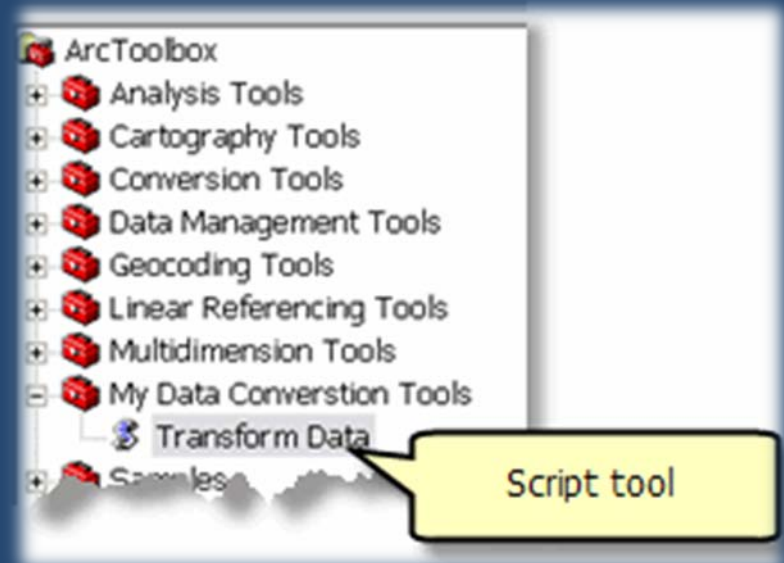


```
C:\WINDOWS\system32\cmd.exe
C:\data\GP_scripts>hillshade_tool_input.py "elevfloat"
Input raster: elevfloat
Output raster: C:\data\GP_scripts\hillshade
Hillshade created
C:\data\GP_scripts>
```



# Make Your Script Part of Geoprocessing Framework

- **Add custom script to ArcToolbox**
  - Use in ModelBuilder
  - Use in Command Line
  - Use in other scripts
  - Inherits all geoprocessing properties
  - Communicates with application
  - Layers added to map, etc.
  - Messages
- **More easily shared than script file**
  - Not everyone knows how to run a stand-alone script
  - Puts a familiar face on your work





# Toolbox Aliases

- Always suffix tool with alias

The image displays two screenshots of the ArcGIS software interface. The left screenshot shows the 'Coverage Tools' toolbox expanded to 'Analysis' > 'Extract', where the 'Clip' tool is highlighted with a red box and labeled with the alias `gp.Clip_arc()`. The right screenshot shows the 'Analysis Tools' toolbox expanded to 'Extract', where the 'Clip' tool is highlighted with a red box and labeled with the alias `gp.Clip_analysis()`.

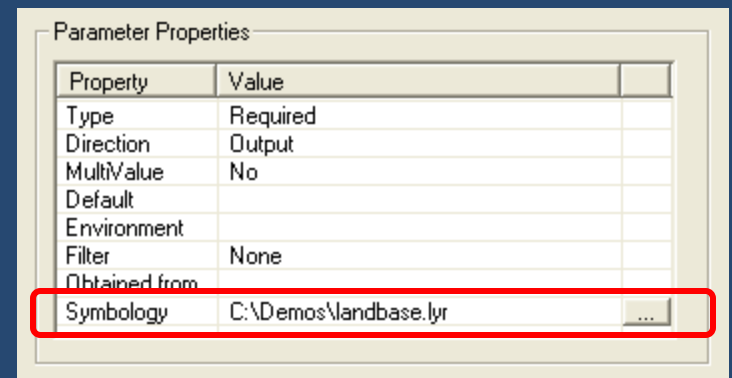
Below these screenshots is a list of toolboxes and their corresponding aliases, connected by red dashed arrows:

ArcToolbox	
+ 3D Analyst Tools	-----> 3d
+ Analysis Tools	-----> analysis
+ Cartography Tools	-----> cartography
+ Conversion Tools	-----> conversion
+ Coverage Tools	-----> arc
+ Data Interoperability Tools	--> interop
+ Data Management Tools	--> management
+ Geocoding Tools	-----> geocoding
+ Geostatistical Analyst Tools	--> ga
+ Linear Referencing Tools	-----> lr
+ Multidimension Tools	-----> md
+ Network Analyst Tools	-----> na
+ Samples	-----> samples
+ Server Tools	-----> lr
+ Spatial Analyst Tools	-----> sa
+ Spatial Statistics Tools	-----> stat
+ Tracking Analyst Tools	-----> ta

- When creating custom toolbox, give it alias

# Script Tool Parameters

- **Script Tools can have parameters**
  - Makes scripts reusable
- **Parameters can have dependencies**
  - Available fields depend on specific feature class selected
- **Many options to control behavior of script tool**
  - Required versus optional parameters
  - Validation messages (“C:\Temp\Example.gdb already exists”)
- **Parameter filters**
  - Coded Value and range domains
  - Feature type domain
  - Workspace domain
- **Symbology for output parameters**



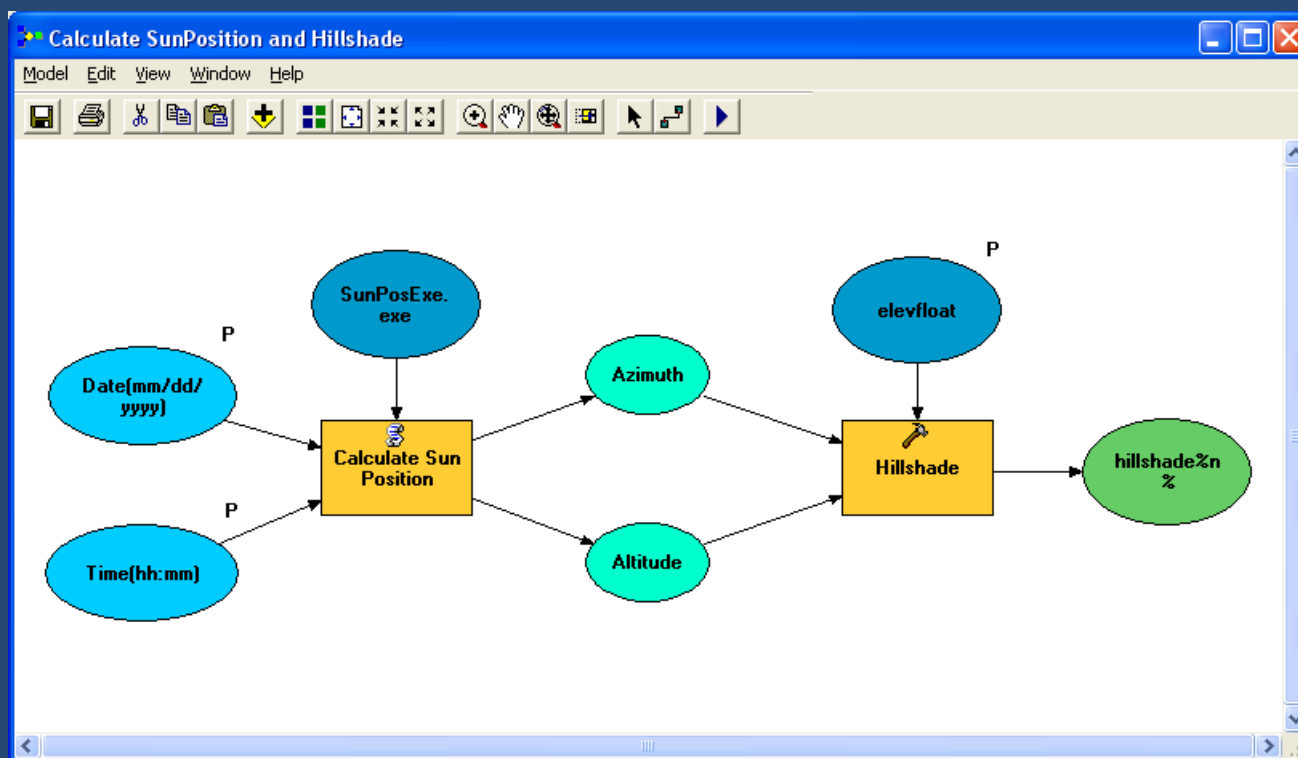
# Script Tool Validation

- **Default Script Tool Validation**
  - Have all the required parameter values been supplied?
  - Are the values of the appropriate data types?
  - Does the input or output exist?
- **Tool Validator Class**
  - Custom tool behavior
  - Can use Python for validation code
  - Extensive Help article available

**ArcGIS Desktop Help > Geoprocessing > Automating your work with scripts > Creating script tools > Customizing script tool behavior**

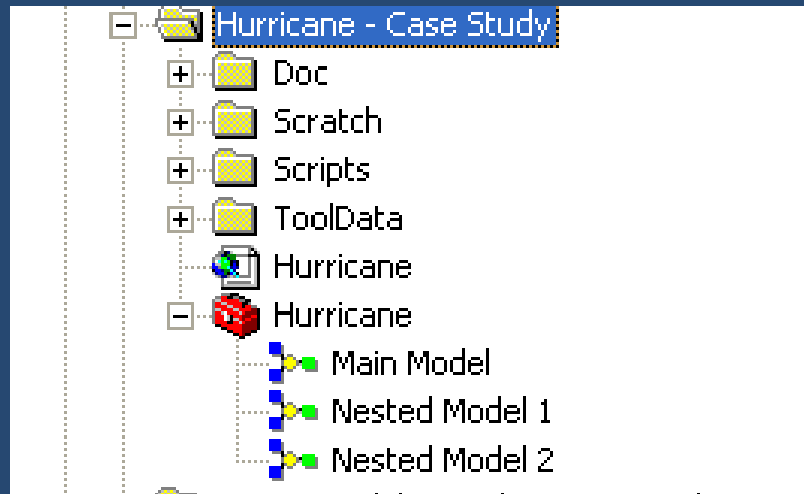
# Demonstration – Putting it all together

- Create a script tool using Sun Position executable
- Add the tool to ModelBuilder to create hillshade at specified time of day
- Run model as batch to generate hillshade at different times of day



# Sharing your Models and Scripts

- Models are stored inside toolbox
- Scripts that are part of toolbox also have stand-alone .py file(s)
- Consider licensing implications when sharing tools
- File Structure suggestion
  - Many shared tools on Community Gallery use this structure

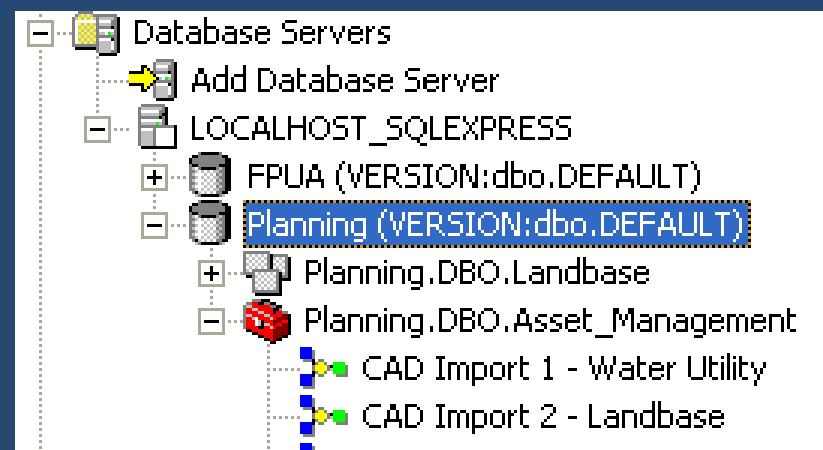


**ArcGIS Desktop Help > Geoprocessing > Sharing Tools and Toolboxes**

# Sharing your Models and Scripts

## *Storing Models and Scripts*

- Toolboxes can be stored inside geodatabases
- It is possible to expose geoprocessing tools as ArcGIS Server services



**ArcGIS Hesktop Help > Geoprocessing > Geoprocessing with ArcGIS Server**

# Training

## Instructor-Led Training

- **Advanced Analysis with ArcGIS**
  - What is spatial analysis? Geoprocessing, ModelBuilder, etc
- **Introduction to Geoprocessing Scripts Using Python**
  - Geoprocessing framework, enumerations, cursor objects
- **Writing Advanced Geoprocessing Scripts Using Python**
  - Working with geometry, lists, dictionaries, subsets of data

## Online Training

- **Geoprocessing with ArcGIS Desktop**
- **Geoprocessing using ModelBuilder**
- **Getting Started With Scripting At ArcGIS 9**

# Where To Get Help

- Geoprocessing Resource Center  
<http://resources.esri.com/geoprocessing>
- ArcGIS Desktop Help  
<http://webhelp.esri.com/arcgisdesktop/9.3>
- Conference Proceedings  
<http://www.esri.com/events/cahinvrug/index.html>