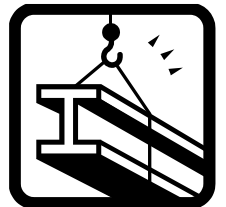
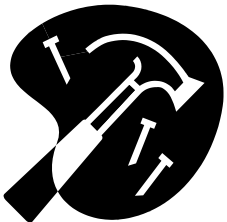

Geoprocessing made easy with ESRI's Modelbuilder

&

Creating a script tool with Python



Geoprocessing?

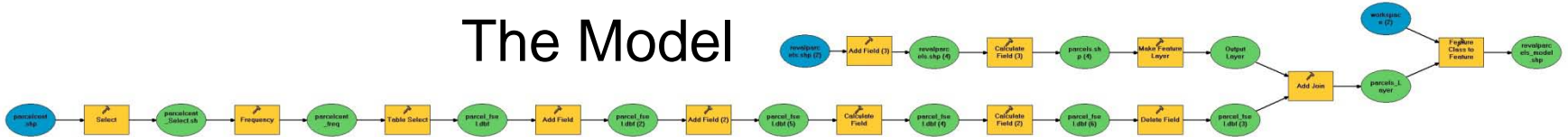
- ESRI Definition:

- A GIS operation used to manipulate GIS data.

- A typical geoprocessing operation takes an input dataset, performs an operation on that dataset, and returns the result of the operation as an output dataset.
 - Common geoprocessing operations include geographic feature overlay, feature selection and analysis, topology processing, raster processing, and data conversion.
 - Geoprocessing allows for definition, management, and analysis of information used to form decisions.
-

Geoprocessing approaches:

The Model



The Script

```
PythonWin
File Edit View Tools Window Help
reval_stacked.py
1 # -----
2 # reval_stacked.py
3 # Purpose: Add a field to revalparcels and populate stacked parcels with a Y
4 # Created on: Mon Mar 23 2009 02:41:19 PM
5 # (geoprocessing functions generated by ArcGIS/ModelBuilder)
6 # (Script by Joel Calhoun)
7 # -----
8
9 # Import system modules
10 import sys, string, os, arcgisscripting
11
12 # Create the Geoprocessor object
13 gp = arcgisscripting.create()
14
15 # Set the necessary product code
16 gp.SetProduct("ArcInfo")
17
18 # Load required toolboxes...
19 gp.AddToolbox("C:/Program Files/ArcGIS/ArcToolbox/Toolboxes/Conversion Tools.tbx")
20 gp.AddToolbox("C:/Program Files/ArcGIS/ArcToolbox/Toolboxes/Data Management Tools.tbx")
21 gp.AddToolbox("C:/Program Files/ArcGIS/ArcToolbox/Toolboxes/Analysis Tools.tbx")
22
Interactive Window
PythonWin 2.4.1 (#65, Mar 30 2005, 09:13:57) [MSC v.1310 32 bit (Intel)] on win32
Portions Copyright 1994-2004 Mark Hammond (mhammond@skippinet.com.au) - see 'Help/About PythonWin' for further copyright information.
>>> |
Ready NUM 00003 005
```

Modelbuilder?



What is it?

ESRI's application to build geoprocessing models.




What is a Modelbuilder model?

The graphical display for a geoprocessing workflow.

Why use it?

Save time on repetitive geoprocessing tasks.

Where is it?

Modelbuilder is accessed from ArcToolbox  in either ArcMap  or ArcCatalog. 

Pilot Project

■ Objectives

Modelbuilder

- ❑ Clip parcels to pilot area extent.
- ❑ Clip additional feature classes to pilot area extent.

Python

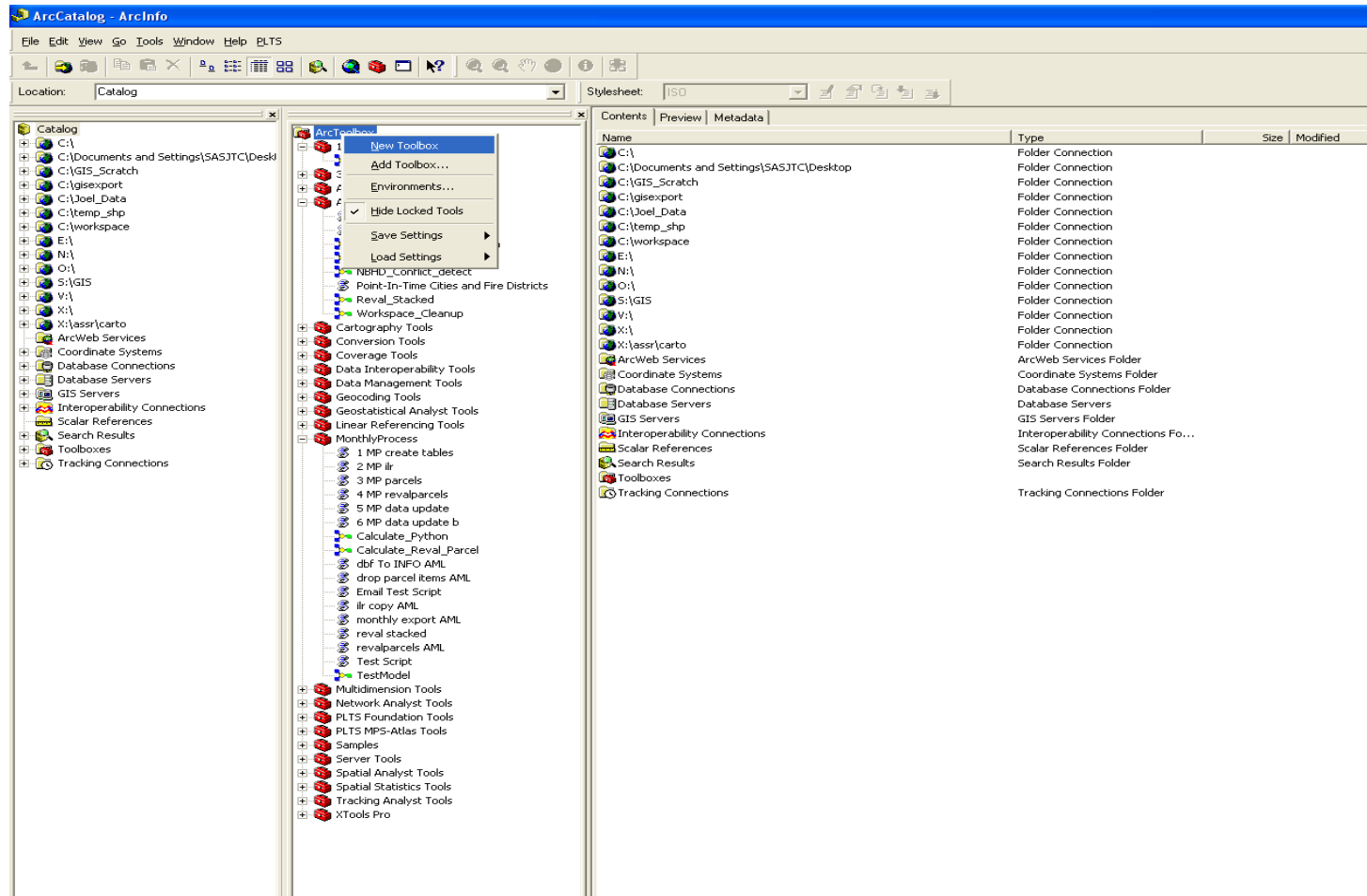
- ❑ Clip parcels to pilot area extent with Python
- ❑ Clip additional feature classes to pilot area extent.

ArcToolbox

- ❑ Create a script tool from a Python script.
-

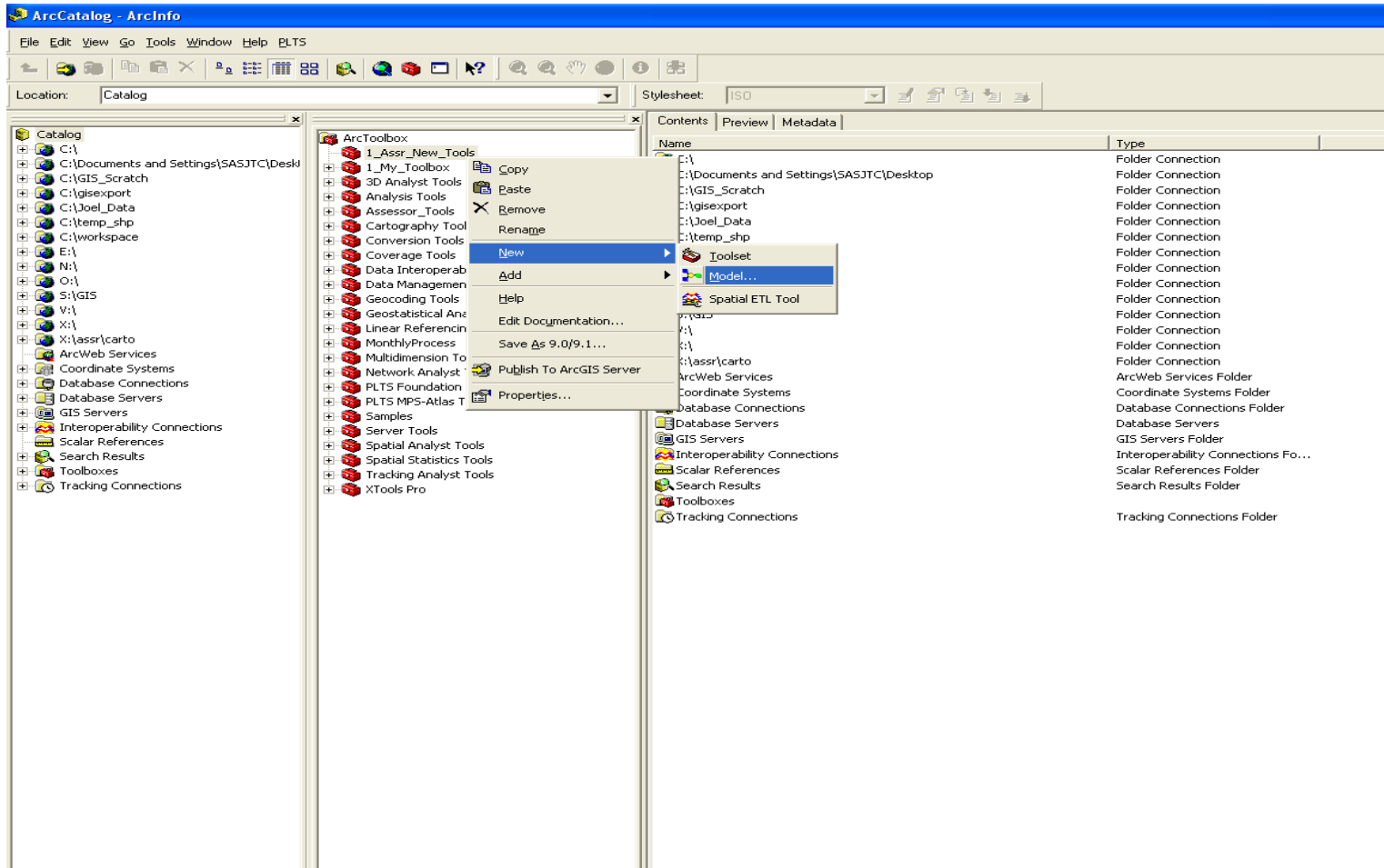
Getting Started: Modelbuilder

Open ArcCatalog and create a new toolbox to store your model



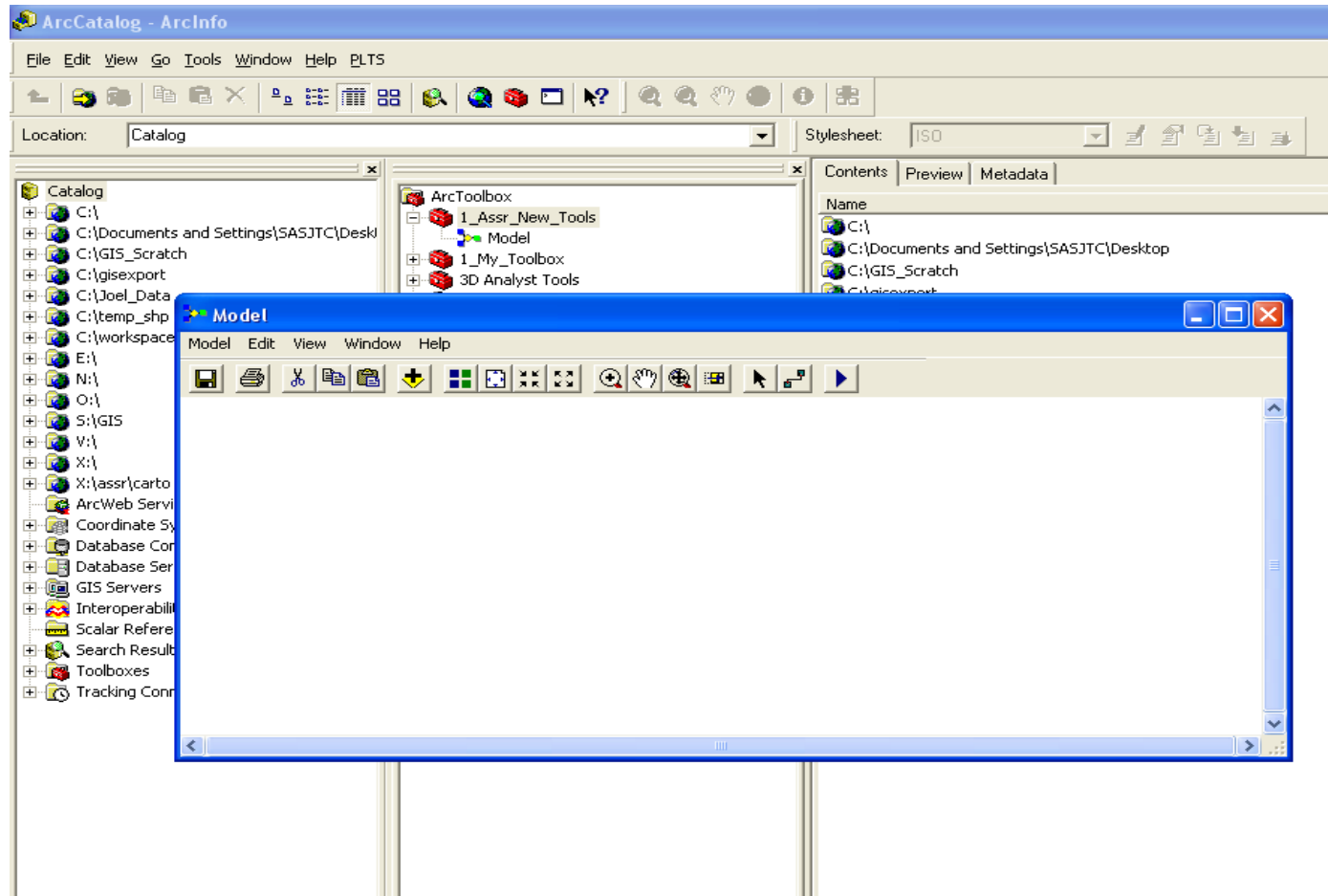
ArcToolbox

Add a new model to your toolbox



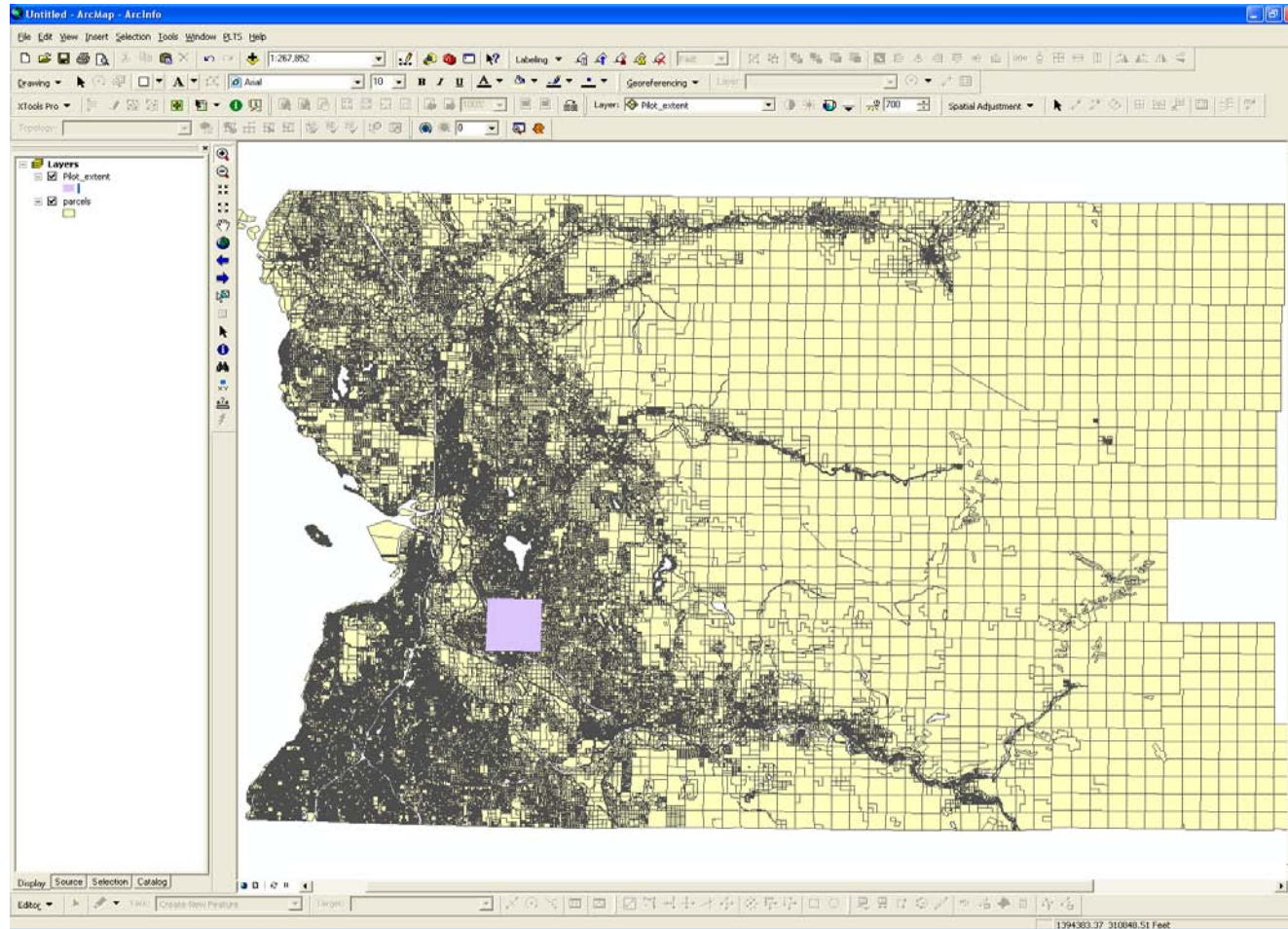
Modelbuilder Approach

The Modelbuilder window



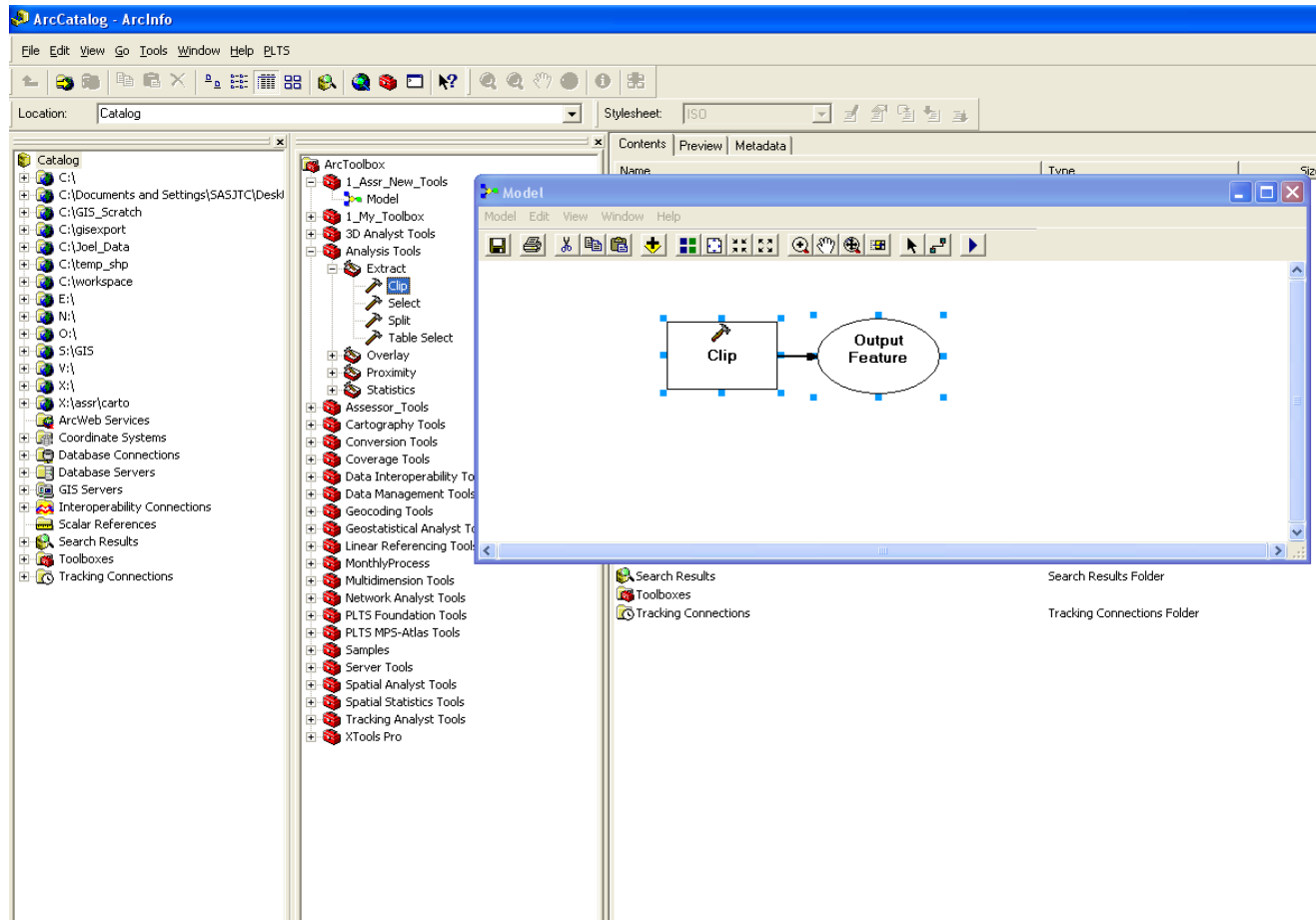
Features

Parcels to be clipped by pilot study extent



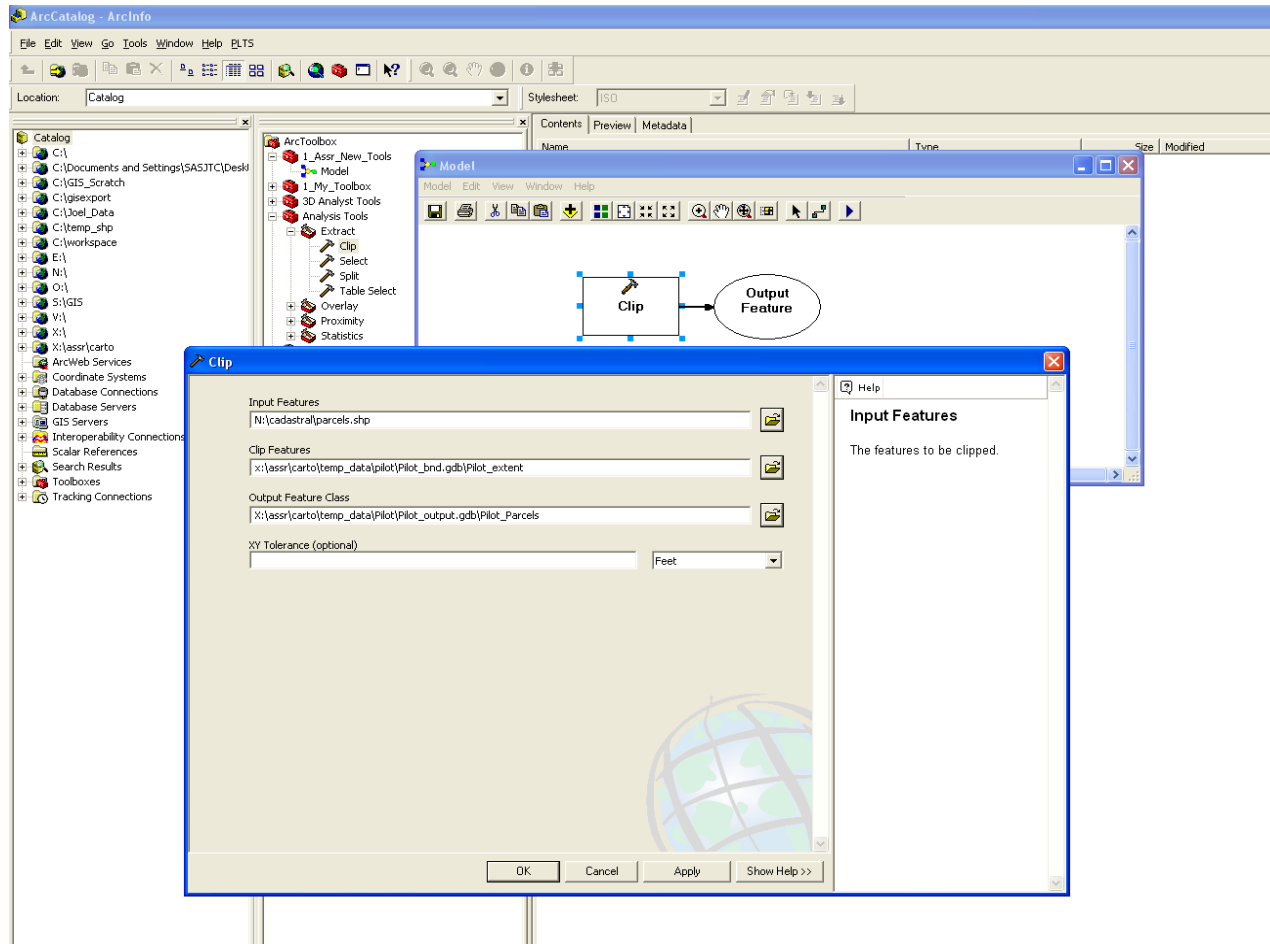
Clip Tool

Drag a tool from toolbox to the Modelbuilder workspace



Clip Parameters

Double click on the tool to bring up the parameter dialog



Clip Model

The completed model

The screenshot displays the ArcCatalog interface with a custom ArcToolbox model named 'Clip' open. The model is a flow diagram with two input nodes, 'Pilot_exte nt' and 'parcels.sh p', both represented as blue ovals. Arrows from these nodes point to a central yellow rectangular process node labeled 'Clip'. An arrow from the 'Clip' node points to an output node, 'Pilot_Parc els', represented as a green oval. The background shows the ArcCatalog catalog tree on the left and a table of folder connections on the right.

Name	Type	Size	Modified
C:\	Folder Connection		
C:\Documents and Settings\SASJTC\Desktop	Folder Connection		
C:\GIS_Scratch	Folder Connection		
C:\gisexport	Folder Connection		
C:\Joel_Data	Folder Connection		
C:\temp_shp	Folder Connection		
C:\workspace	Folder Connection		
Folder Connection	Folder Connection		
Folder Connection	Folder Connection		
Folder Connection	Folder Connection		
Folder Connection	Folder Connection		
Folder Connection	Folder Connection		
Folder Connection	Folder Connection		
Folder Connection	Folder Connection		
ArcWeb Services Folder	ArcWeb Services Folder		
Coordinate Systems Folder	Coordinate Systems Folder		
Database Connections Folder	Database Connections Folder		
Database Servers	Database Servers		
GIS Servers Folder	GIS Servers Folder		
Interoperability Connections Fo...	Interoperability Connections Fo...		
Scalar References Folder	Scalar References Folder		
Search Results Folder	Search Results Folder		
Tracking Connections Folder	Tracking Connections Folder		

Clip Model Successful

3D shading indicates that a component of the model ran successfully

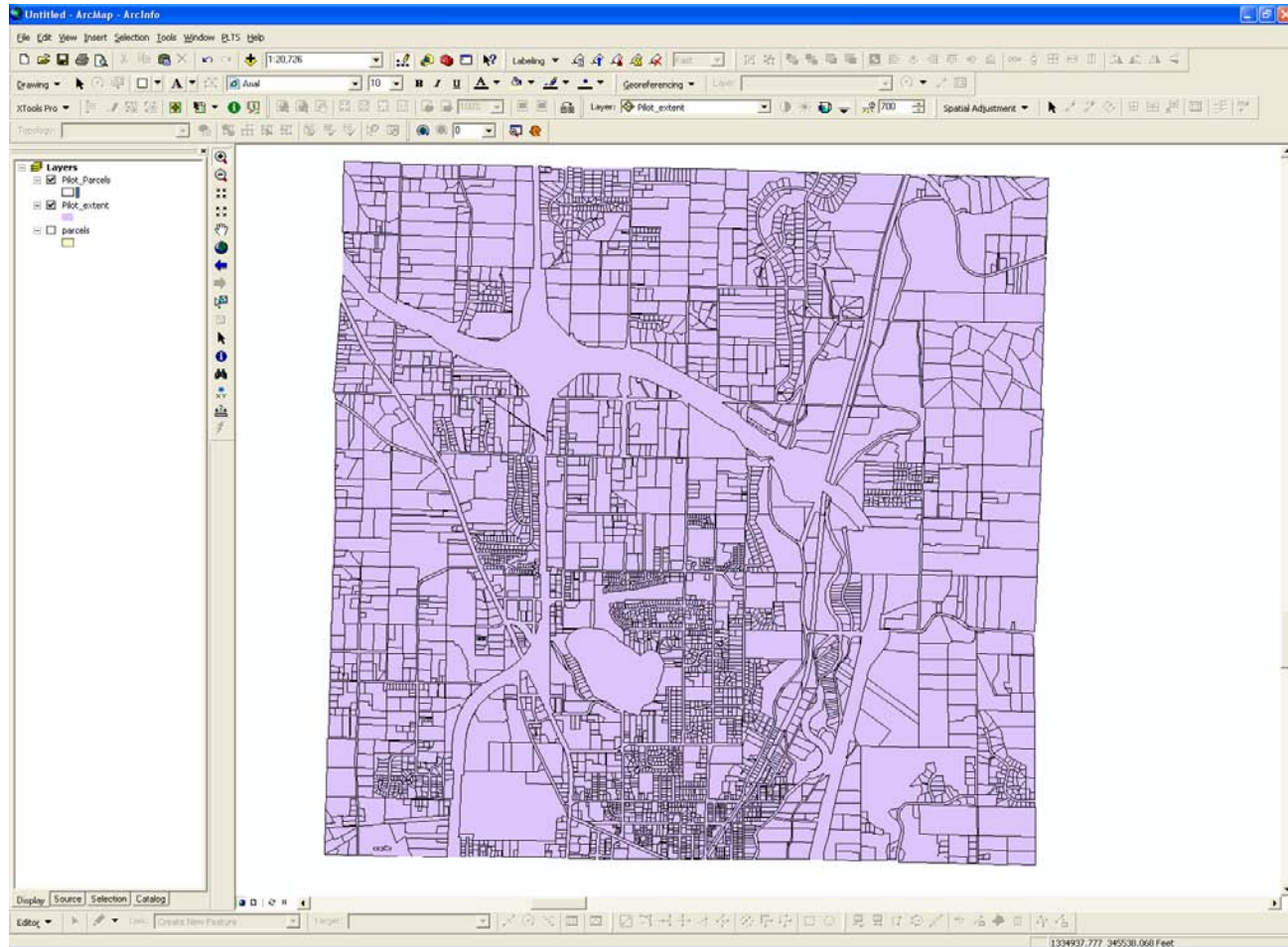
The screenshot displays the ArcCatalog interface with a workflow model named 'Clip_Demo' open. The model consists of three components: 'Pilot_exte nt' (blue oval), 'parcels.sh p' (blue oval), and 'Clip' (yellow rectangle). The 'Clip' component is highlighted with a 3D effect, indicating successful execution. The output of the model is 'Pilot_Parc els' (green oval).

A 'Clip_Demo' dialog box is open, showing the execution progress. The progress bar is full, and the status is 'Completed'. The log text reads:

```
Cracking Features...
Assembling Features...
Executed (Clip) successfully.
End Time: Fri Jun 05 15:22:59 2009 (Elapsed Time:
17.00 seconds)
```

Clip Result

The new clipped parcel dataset



Clip Model

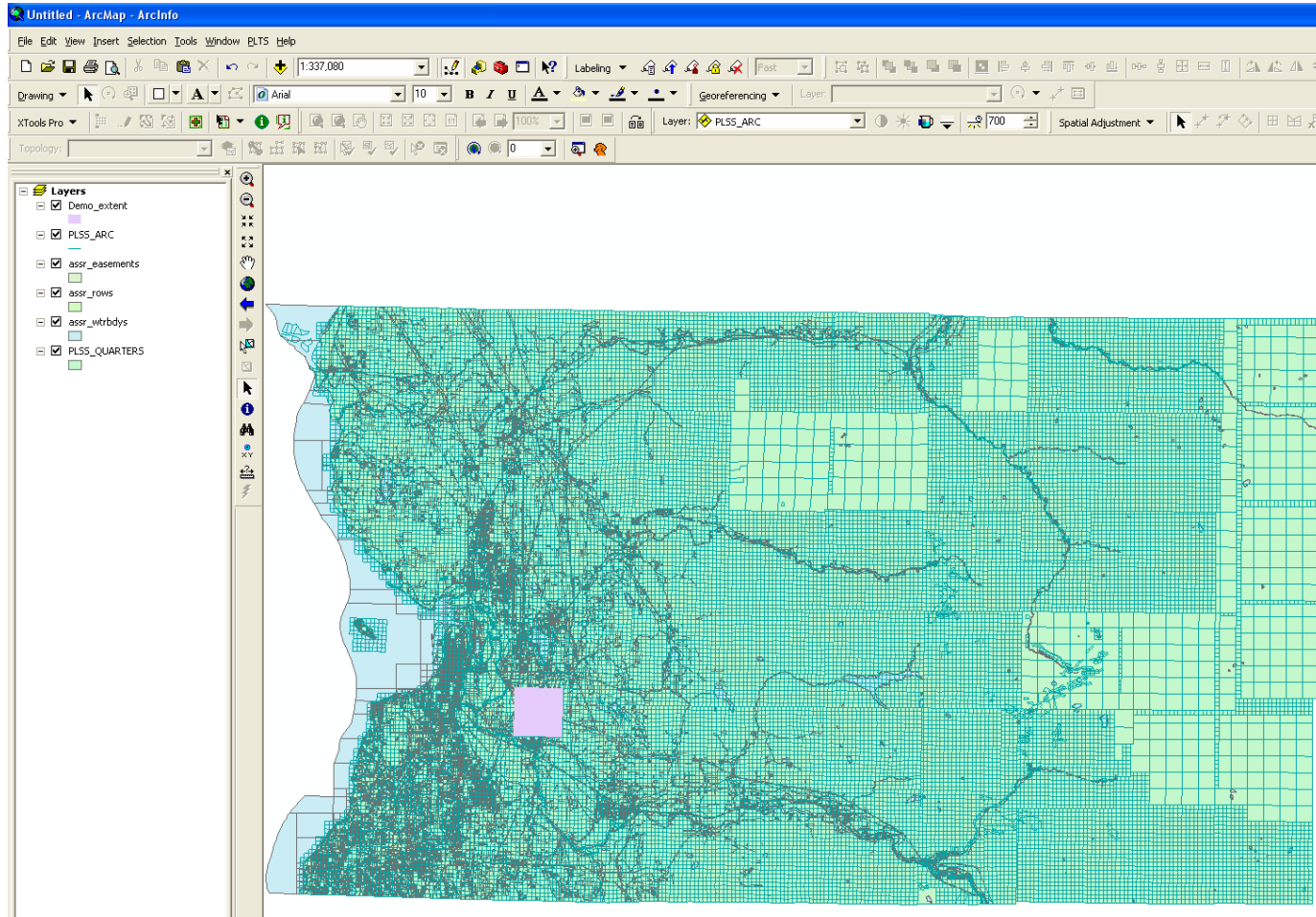
Single clip

The screenshot displays the ArcCatalog software interface. The main window shows a 'Model' workflow diagram. The diagram consists of three nodes: two input nodes on the left, 'Pilot_exte nt' (blue oval) and 'parcels.sh p' (blue oval), both pointing to a central yellow rectangular tool node labeled 'Clip'. An arrow points from the 'Clip' tool to an output node on the right, 'Pilot_Parc els' (green oval). The background shows the ArcCatalog interface with a file tree on the left and a 'Contents' table on the right. The 'Contents' table lists various folder connections and their properties.

Name	Type	Size	Modified
C:\	Folder Connection		
C:\Documents and Settings\SASJTC\Desktop	Folder Connection		
C:\GIS_Scratch	Folder Connection		
C:\gisexport	Folder Connection		
C:\Joel_Data	Folder Connection		
C:\temp_shp	Folder Connection		
C:\workspace	Folder Connection		
Folder Connection	Folder Connection		
Folder Connection	Folder Connection		
Folder Connection	Folder Connection		
Folder Connection	Folder Connection		
Folder Connection	Folder Connection		
Folder Connection	Folder Connection		
Folder Connection	Folder Connection		
ArcWeb Services Folder	ArcWeb Services Folder		
Coordinate Systems Folder	Coordinate Systems Folder		
Database Connections Folder	Database Connections Folder		
Database Servers	Database Servers		
GIS Servers Folder	GIS Servers Folder		
Interoperability Connections Fo...	Interoperability Connections Fo...		
Scalar References Folder	Scalar References Folder		
Search Results Folder	Search Results Folder		
Tracking Connections Folder	Tracking Connections Folder		

Features

The feature classes to be clipped



Multi Clip

Open the input data properties, select the list option

The image displays a workflow diagram within a software window titled 'Clip_Demo'. The workflow consists of three main components: two input data sources, a processing tool, and an output data source. The inputs are 'Pilot_extent' and 'parcels.shp', both represented by blue ovals. Arrows from these two inputs point to a yellow rectangular tool icon labeled 'Clip'. An arrow from the 'Clip' tool points to the final output, 'Pilot_Parcel s', represented by a green oval. The 'parcels.shp' input is highlighted with a blue dashed border. In the foreground, the 'parcels.shp Properties' dialog box is open, showing the 'General' tab. Under the heading 'This variable contains:', the radio button for 'A list of values' is selected. Below this, there is a 'Feedback' section with a 'Feedback Variable:' label and an empty text box. At the bottom of the dialog are 'OK', 'Cancel', and 'Apply' buttons.

Clip Iteration

Add multiple feature classes to be clipped

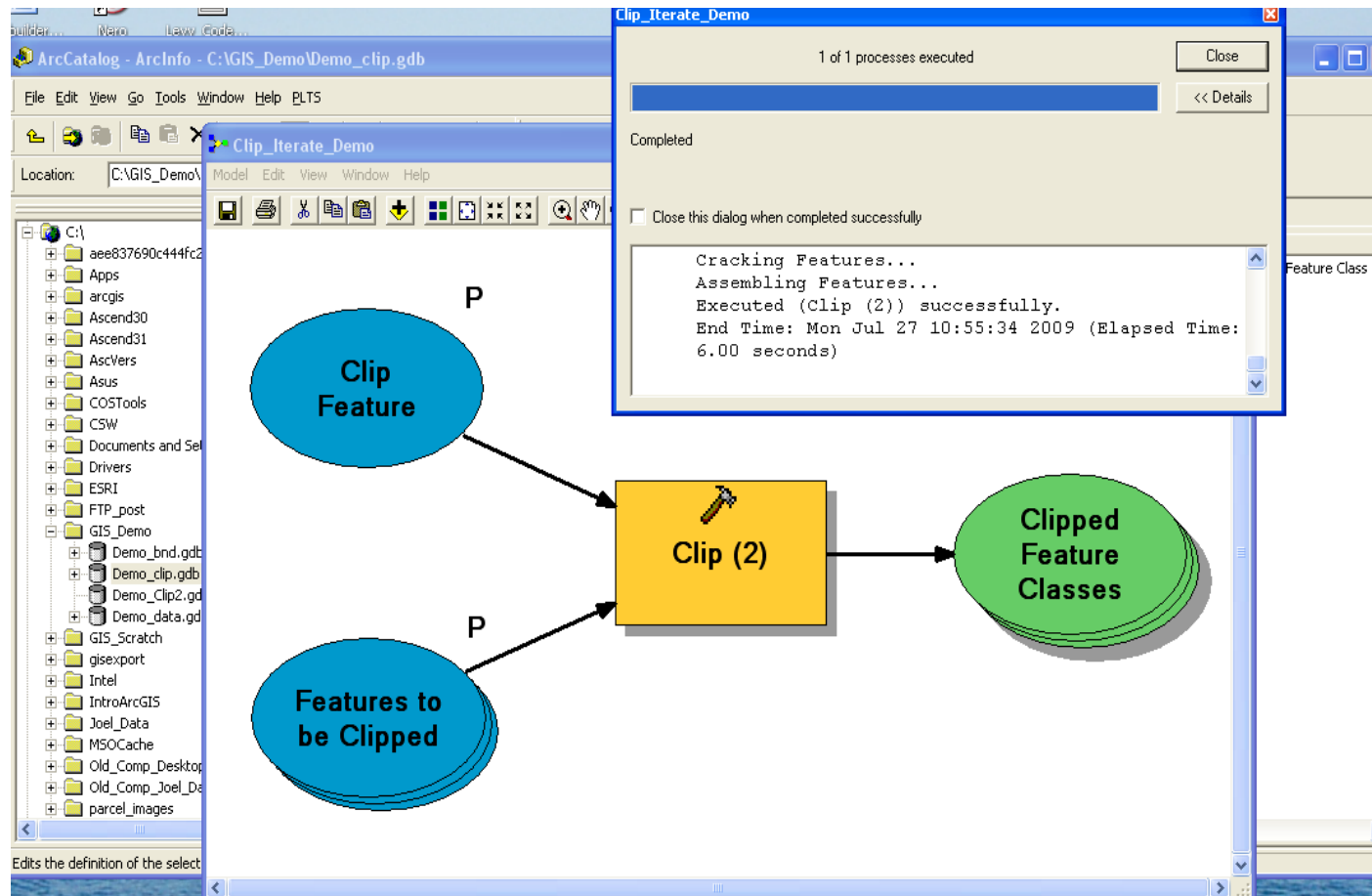
The image displays a screenshot of the ArcGIS software interface, specifically the 'Features to be Clipped' dialog box. The dialog box is titled 'Features to be Clipped' and contains a list of five feature classes to be clipped:

Features to be Clipped	
1	C:\GIS_Demo\Demo_data.gdb\assr_easements
2	C:\GIS_Demo\Demo_data.gdb\assr_rows
3	C:\GIS_Demo\Demo_data.gdb\assr_wtrbdys
4	C:\GIS_Demo\Demo_data.gdb\PLSS_ARC
5	C:\GIS_Demo\Demo_data.gdb\PLSS_QUARTERS

The dialog box also includes a 'Help' button, a 'No description available' message, and buttons for 'OK', 'Cancel', 'Apply', and 'Show Help >>'. To the right of the dialog box, a process flow diagram illustrates the workflow. It shows two input boxes labeled 'Clip Feature' and 'Features to be Clipped', both with a 'P' above them, pointing to a central yellow box labeled 'Clip (2)'. An arrow points from the 'Clip (2)' box to an output box on the right.

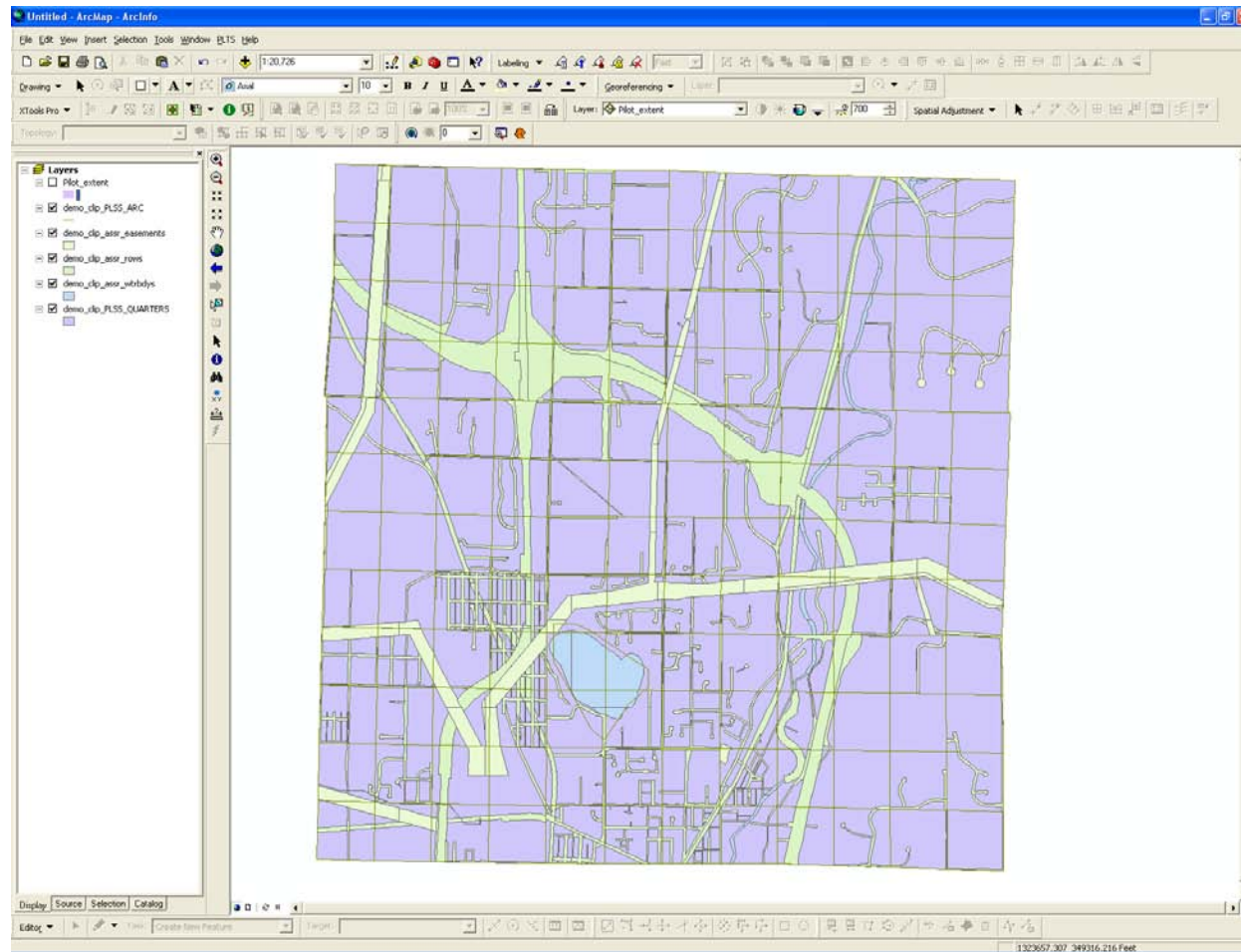
Multi Clip Complete

Clip iteration successful



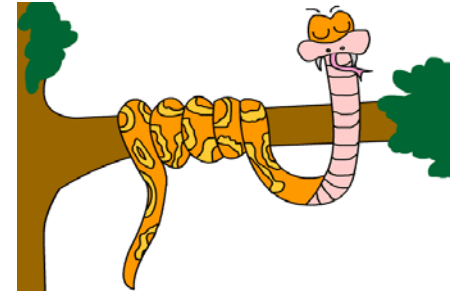
Results

Verify the results



Modelbuilder Demo

Python?



What is it?

An open-source object-oriented, scripting language.

Why use it?

- ❑ It is free and comes bundled with ArcGIS.
- ❑ Most sample scripts and help documentation are written in Python.
- ❑ Modelbuilder models can be easily exported as Python scripts.
- ❑ It has an integrated development environment with debugging tools.

Where can I get further information?

- ❑ www.python.org (official website)
- ❑ ArcGIS Desktop Help

Getting Started: Python

Open the script in PythonWin or IDLE

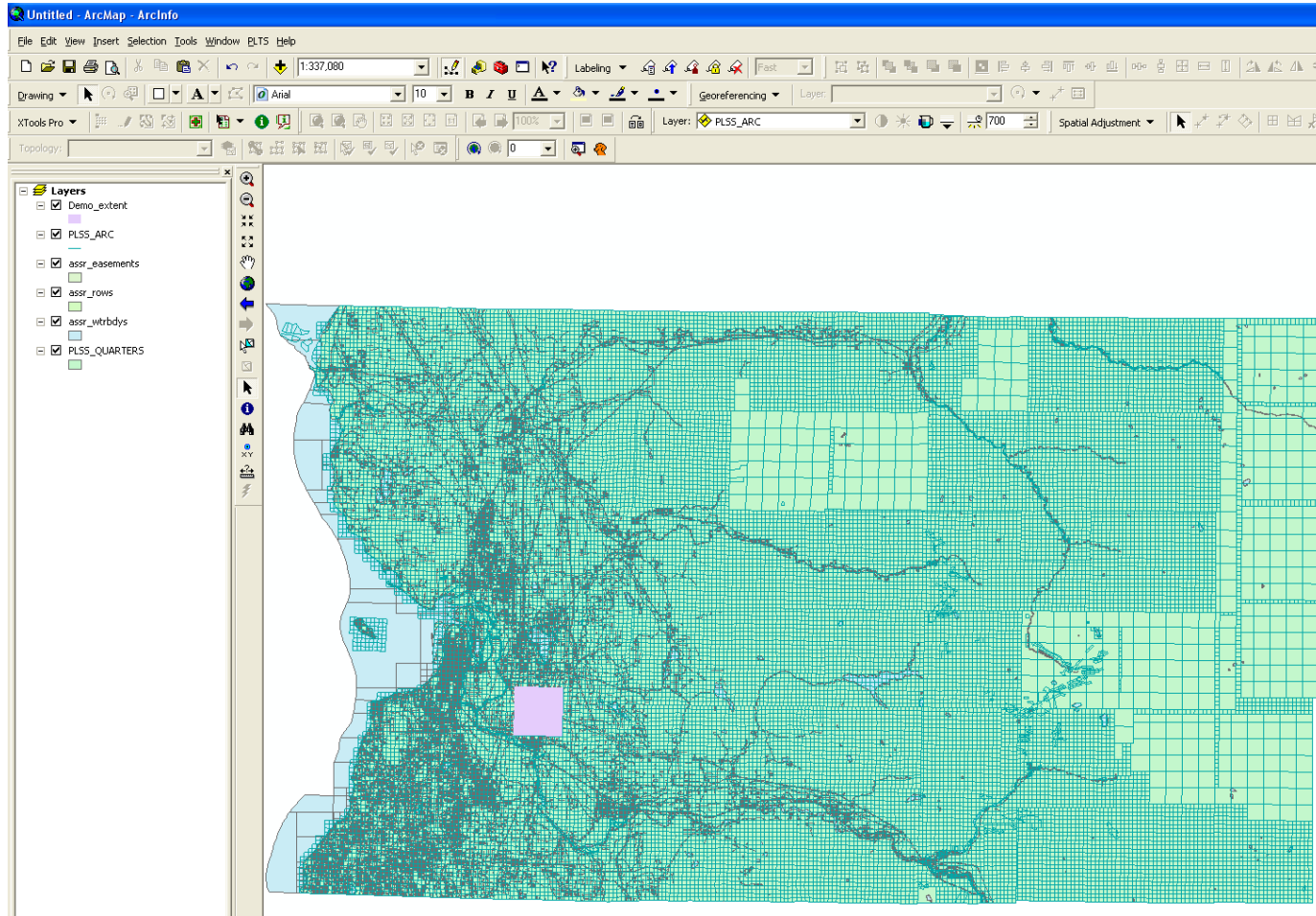
```
PythonWin
File Edit View Tools Window Help
Clip_Demo.py
1 # -----
2 # Clip_Demo.py
3 # Created on: Mon Jun 08 2009 02:17:48 PM
4 # (generated by ArcGIS/ModelBuilder)
5 # -----
6
7 # Import system modules
8 import sys, string, os, arcpy
9
10 # Create the Geoprocessor object
11 gp = arcpy.CreateGeoprocessor()
12
13 # Load required toolboxes...
14 gp.AddToolbox("C:/Program Files/ArcGIS/ArcToolbox/Toolboxes/Analysis Tools.tbx")
15
16
17 # Local variables...
18 Pilot_Parcel = "X:\\assr\\carto\\temp_data\\Pilot\\Pilot_output.gdb\\Pilot_Parcel"
19 parcels_shp = "N:\\cadastral\\parcels.shp"
20 Pilot_extent = "x:\\assr\\carto\\temp_data\\pilot\\Pilot_bnd.gdb\\Pilot_extent"
21
22 # Process: Clip...
23 gp.Clip_analysis(parcels_shp, Pilot_extent, Pilot_Parcel, "")
24
25
Interactive Window
PythonWin 2.4.1 (#65, Mar 30 2005, 09:13:57) [MSC v.1310 32 bit (Intel)] on win32.
Portions Copyright 1994-2004 Mark Hammond (mhammond@skippinet.com.au) - see 'Help/About PythonWin' for further copyright information.
>>>
```

Geoprocessor module

Geoprocessor object

Features

The feature classes to be clipped



ArcGIS Help

Clip multiple feature classes

The screenshot shows the ArcGIS Desktop Help window. The title bar reads 'ArcGIS Desktop Help'. The navigation bar includes 'Back', 'Forward', 'Home', 'Font', 'Print', and 'Online Help'. The search bar contains the text 'scripts, exposing methods'. The search results list includes 'scripts', 'exposing methods', 'exposing properties', 'generating text on hatches with', 'getting messages', 'importing modules into', 'introducing to ArcToolbox', 'limitations on modifying schema of data w', 'process of writing of', 'reasons for use of', 'running outside ArcGIS applications', 'saving and naming', 'saving as geoprocessing tools', 'setting messages', 'setting parameter information', 'setting parameters', 'stopping', 'transferring environment settings to', 'updating schema of data with', 'using as source of script tools', 'using layers in', 'using multivalue input in', 'using string variables in', 'viewing environment settings in', 'vs. models', 'Scripts folder', 'contents of', 'scroll bars', 'viewing columns with', 'viewing rows with', 'SD (surface distance)', 'explanation of', 'SDC data', 'using', and 'SDC dataset'. The main content area is titled 'Geoprocessing' and has a sub-section 'Tools and methods'. Below this is a 'related topics' link. The text explains that the geoprocessing object exposes methods and properties for scripting. It provides an example of using the `ListFeatureClasses` method in a Python script. The code block is as follows:

```
gp.workspace = "d:/MyData"
out_workspace = "d:/MyData/Results/"
clip_features = "d:/MyData/TestArea/Boundary.shp"

# Get a list of all feature classes in the workspace
fcs = gp.ListFeatureClasses()

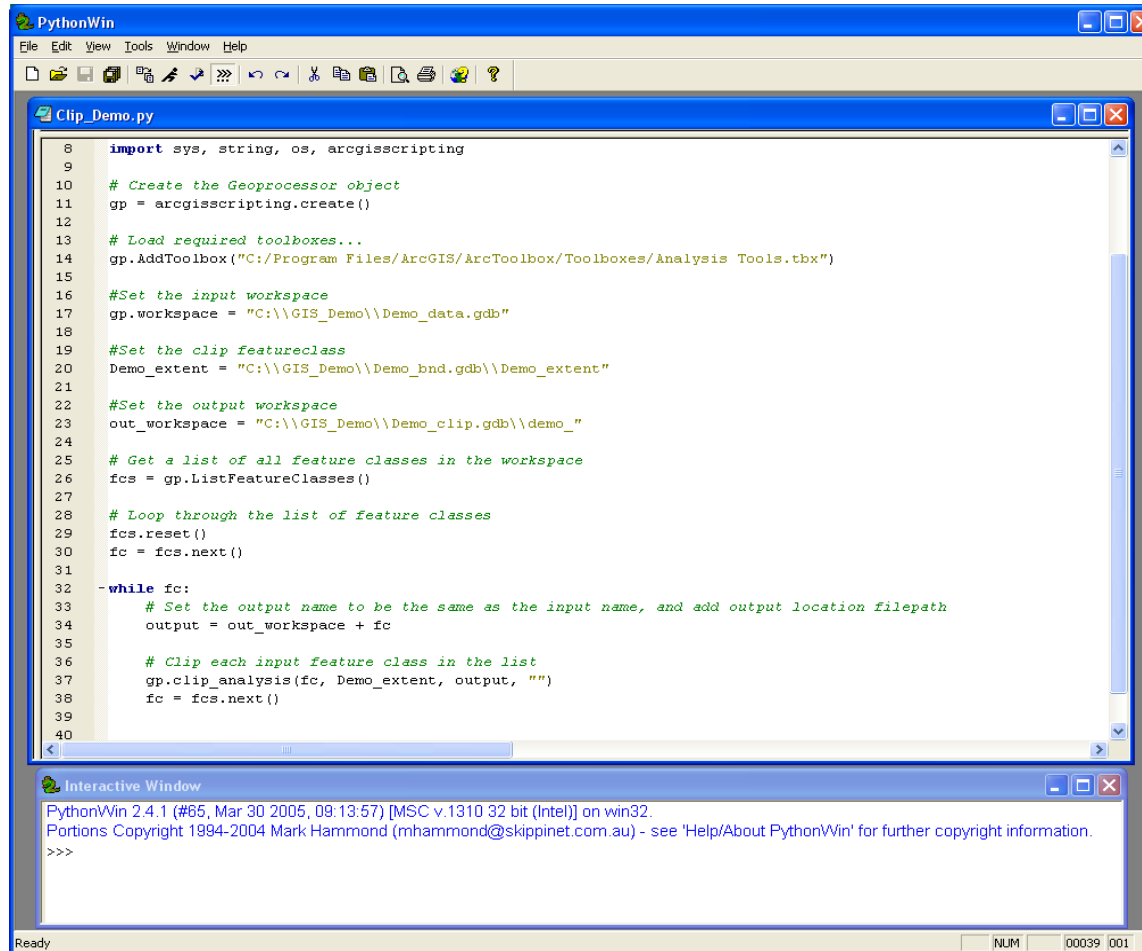
# Loop through the list of feature classes
fcs.reset()
fc = fcs.next()
while fc:
    # Set the output name to be the same as the input name, and locate in the 'out_workspace' workspace
    output = out_workspace + fc

    # Clip each input feature class in the list
    gp.clip_analysis(fc, clip_features, output, 0.1)
    fc = fcs.next()
```

Below the code block, the text states: 'Toolbox tools can be accessed as methods directly from the geoprocessor as well. The geoprocessor can access any tool that has been registered with the system, so geoprocessing operations, such as Union or Buffer, can be easily executed, as well as model and script tools. In the example above, the Clip tool from the [Analysis toolbox](#) is used to clip a number of feature classes in a batch operation. [Learn more about running a tool](#)'

Enhanced Script

Loop through feature classes



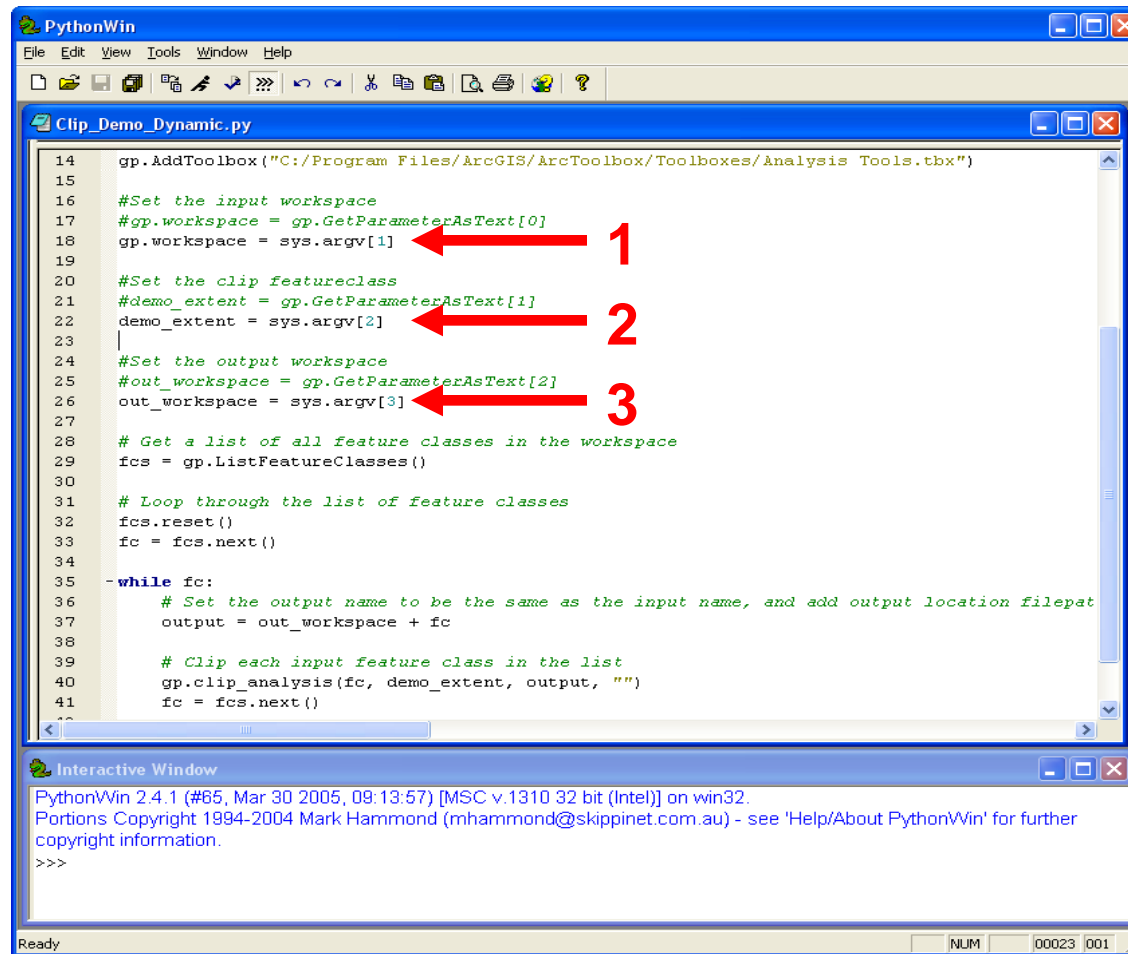
The image shows a screenshot of the PythonWin application window. The main window displays a Python script named 'Clip_Demo.py'. The script is designed to loop through feature classes in a workspace and perform a clip analysis. The code includes comments in green and Python code in black. The script sets up the ArcGIS Geoprocessor, adds the Analysis Tools toolbox, and configures the workspace and feature classes. It then uses a while loop to iterate through the feature classes, setting the output name and performing the clip analysis for each class.

```
8 import sys, string, os, arcgisscripting
9
10 # Create the Geoprocessor object
11 gp = arcgisscripting.create()
12
13 # Load required toolboxes...
14 gp.AddToolbox("C:/Program Files/ArcGIS/ArcToolbox/Toolboxes/Analysis Tools.tbx")
15
16 #Set the input workspace
17 gp.workspace = "C:\\GIS_Demo\\Demo_data.gdb"
18
19 #Set the clip featureclass
20 Demo_extent = "C:\\GIS_Demo\\Demo_bnd.gdb\\Demo_extent"
21
22 #Set the output workspace
23 out_workspace = "C:\\GIS_Demo\\Demo_clip.gdb\\demo_"
24
25 # Get a list of all feature classes in the workspace
26 fcs = gp.ListFeatureClasses()
27
28 # Loop through the list of feature classes
29 fcs.reset()
30 fc = fcs.next()
31
32 -while fc:
33     # Set the output name to be the same as the input name, and add output location filepath
34     output = out_workspace + fc
35
36     # Clip each input feature class in the list
37     gp.clip_analysis(fc, Demo_extent, output, "")
38     fc = fcs.next()
39
40
```

The bottom window is the Interactive Window, which shows the PythonWin version (2.4.1), the date and time (Mar 30 2005, 09:13:57), the system information (MSC v.1310 32 bit (Intel) on win32), and the copyright information (Portions Copyright 1994-2004 Mark Hammond (mhammond@skippinet.com.au) - see 'Help/About PythonWin' for further copyright information). The prompt is >>>.

Dynamic Scripts

Use arguments to make your scripts dynamic



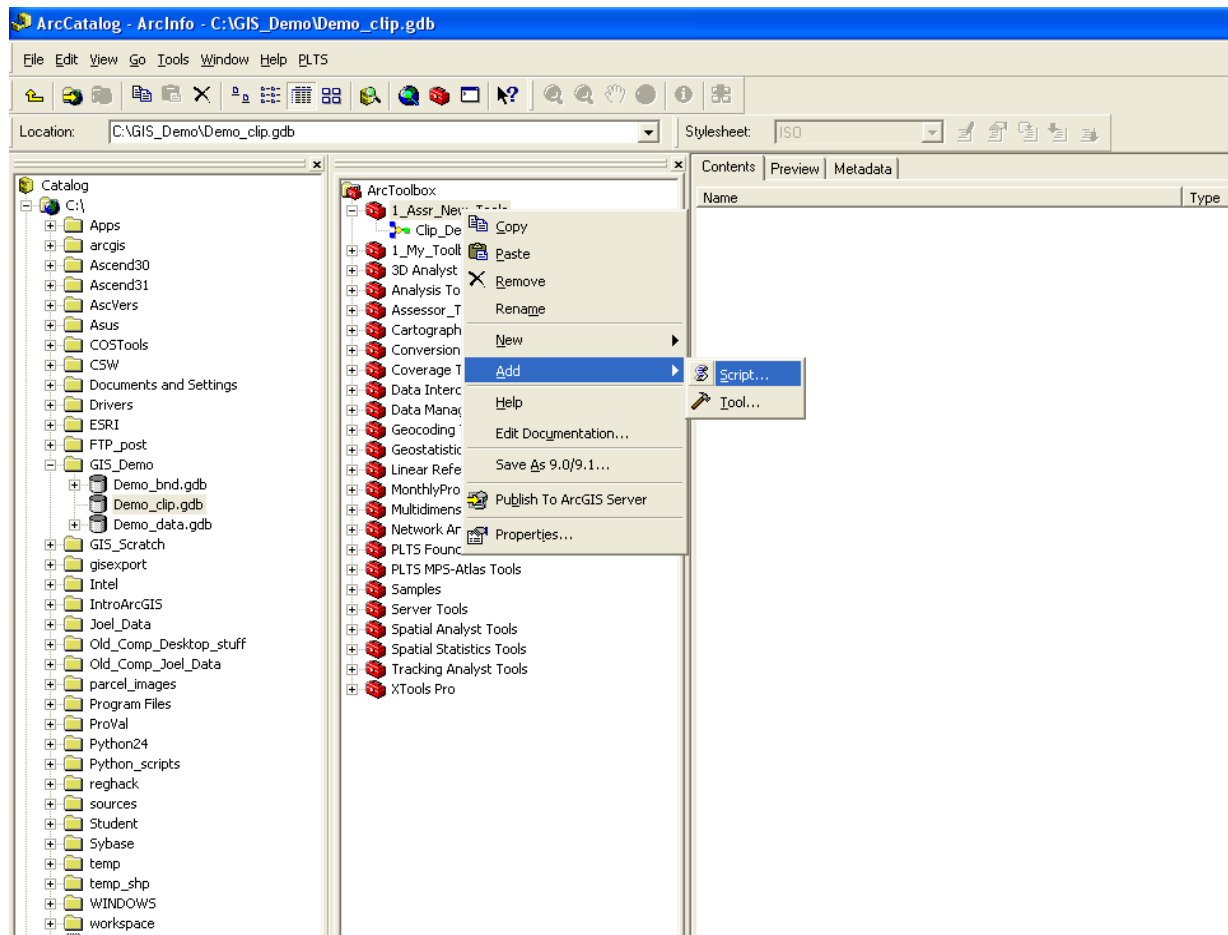
The screenshot shows a PythonWin window titled "PythonWin" with a menu bar (File, Edit, View, Tools, Window, Help) and a toolbar. The main window displays a script named "Clip_Demo_Dynamic.py" with the following code:

```
14 gp.AddToolbox("C:/Program Files/ArcGIS/ArcToolbox/Toolboxes/Analysis Tools.tbx")
15
16 #Set the input workspace
17 #gp.workspace = gp.GetParameterAsText[0]
18 gp.workspace = sys.argv[1] ← 1
19
20 #Set the clip featureclass
21 #demo_extent = gp.GetParameterAsText[1]
22 demo_extent = sys.argv[2] ← 2
23
24 #Set the output workspace
25 #out_workspace = gp.GetParameterAsText[2]
26 out_workspace = sys.argv[3] ← 3
27
28 # Get a list of all feature classes in the workspace
29 fcs = gp.ListFeatureClasses()
30
31 # Loop through the list of feature classes
32 fcs.reset()
33 fc = fcs.next()
34
35 -while fc:
36     # Set the output name to be the same as the input name, and add output location filepath
37     output = out_workspace + fc
38
39     # Clip each input feature class in the list
40     gp.clip_analysis(fc, demo_extent, output, "")
41     fc = fcs.next()
42
```

The script is annotated with three red arrows and numbers (1, 2, 3) pointing to the lines where arguments are passed to `sys.argv`. Below the script editor is an "Interactive Window" showing the PythonWin version (2.4.1) and copyright information, with a prompt `>>>` at the bottom.

Creating a script tool from a script

Add the script to your new toolbox



Add Script

Set up the new tool

Name:

Dynamic_Clip

Label:

Dynamic Feature Class Clip

Description:

Clips all features classes in a workspace to an input clip feature class.

Stylesheet:

Store relative path names (instead of absolute paths)

< Back Next > Cancel

Associate your script

Script File:

C:\Python_scripts\Clip_Demo_Dynamic.py

Show command window when executing script

< Back Next > Cancel

Set up your arguments

Display Name	Data Type
Input Feature Class ...	Workspace or Feature Dataset
Clip Extent Feature C...	Feature Class
Output Feature Class...	Workspace or Feature Dataset
@	Feature Layer

Click any parameter above to see its properties below.

Parameter Properties

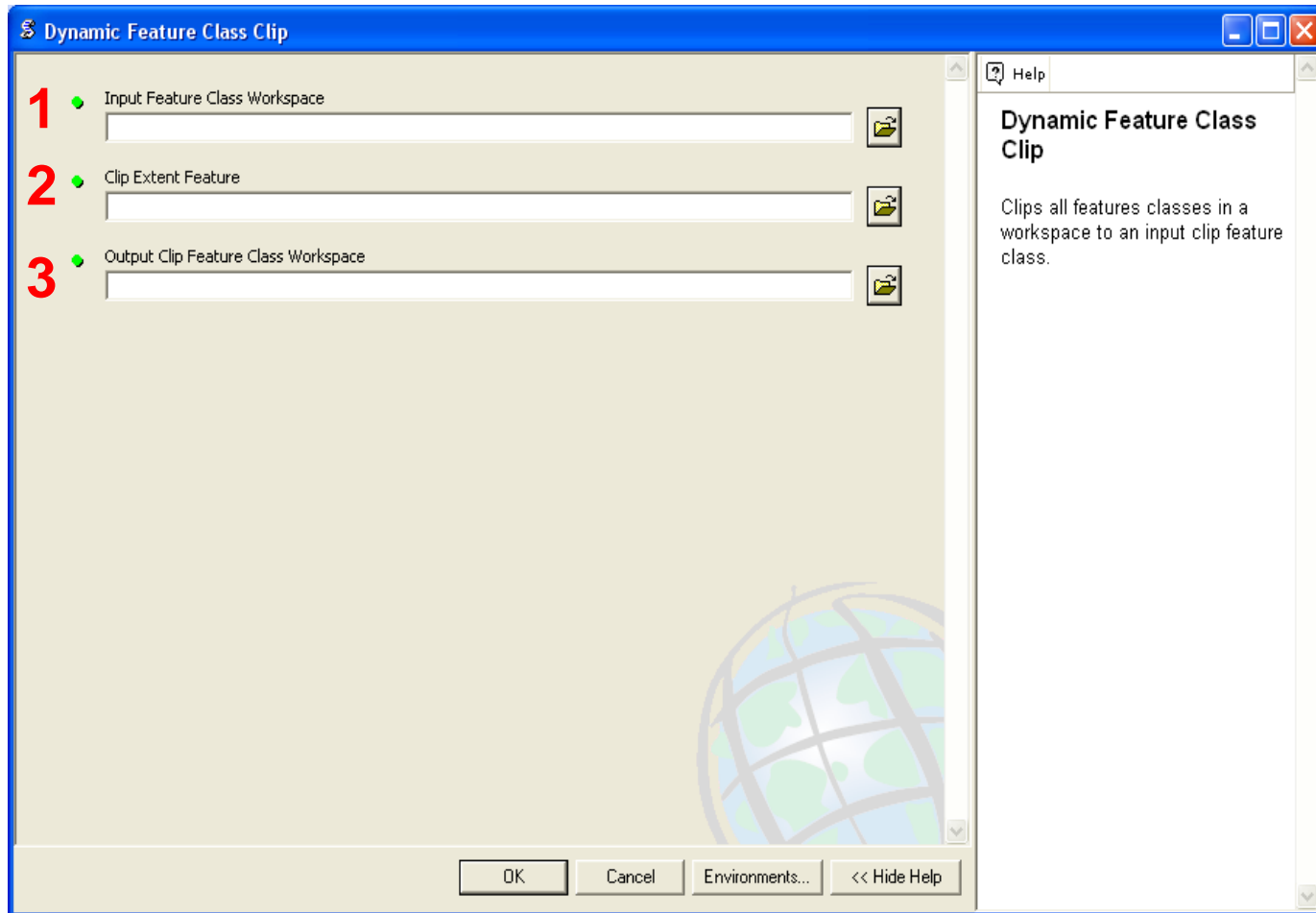
Property	Value
Type	Required
Direction	Input
Multivalue	No
Default	
Environment	
Domain	
Dependency	

To add a new parameter, type the name into an empty row in the name column, click in the Data Type column to choose a data type, then edit the Parameter Properties.

< Back Finish Cancel

Custom Script Tool

The clip tool parameter dialog



Script Tool Complete

New clipped feature classes in the red box

The screenshot displays the ArcCatalog interface. The main window shows a catalog tree on the left with the 'Dynamic Feature Class Clip' tool selected under the 'Clip_Demo' folder. A dialog box titled 'Dynamic Feature Class Clip' is open, showing the status 'Completed' and a log of the execution process. The log text is as follows:

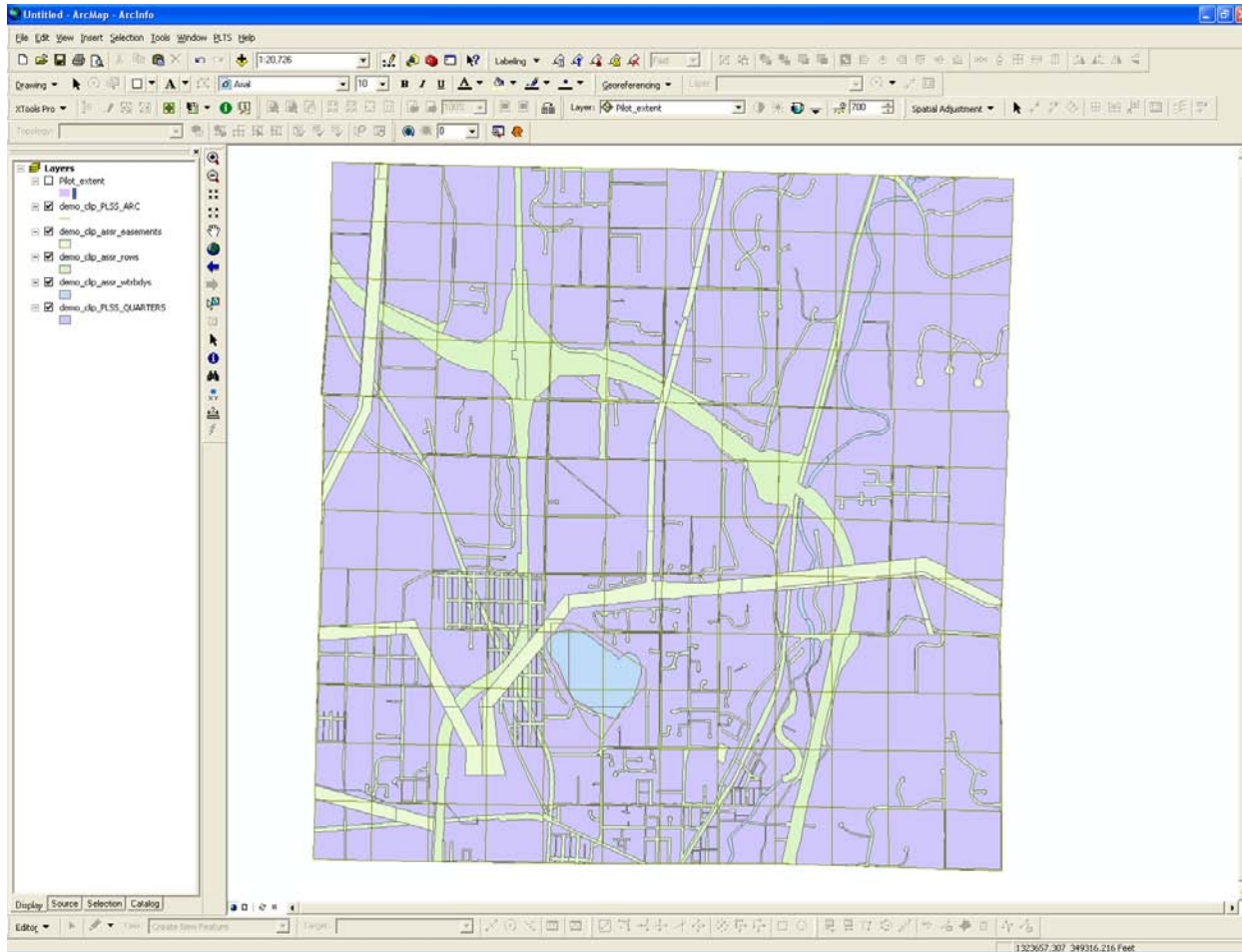
```
Executing (Dynamic_Clip_1): Dynamic_Clip C:\GIS_Demo\Demo_data.gdb C:\GIS_Demo\Demo_bnd.gdb\Demo_extent C:\GIS_Demo\Demo_clip.gdb
Start Time: Mon Jun 15 09:59:32 2009
Running script Dynamic_Clip...
Completed script Dynamic_Clip...
Executed (Dynamic_Clip_1) successfully.
End Time: Mon Jun 15 09:59:43 2009 (Elapsed Time: 11.00 seconds)
```

In the background, a table in the 'Contents' pane lists the newly created feature classes, which are highlighted with a red border:

Name	Type	Size	Creation Date
demo_clip_asrr_easements	File Geodatabase Feature Class	48.83 KB	6/15/2009 9:59:40 AM
demo_clip_asrr_rows	File Geodatabase Feature Class	383.50 KB	6/15/2009 9:59:42 AM
demo_clip_asrr_wtrbdys	File Geodatabase Feature Class	21.43 KB	6/15/2009 9:59:39 AM
demo_clip_PLSS_ARC	File Geodatabase Feature Class	58.72 KB	6/15/2009 9:59:38 AM
demo_clip_PLSS_QUARTERS	File Geodatabase Feature Class	17.73 KB	6/15/2009 9:59:37 AM

Results

Verify the results



Python Demo
