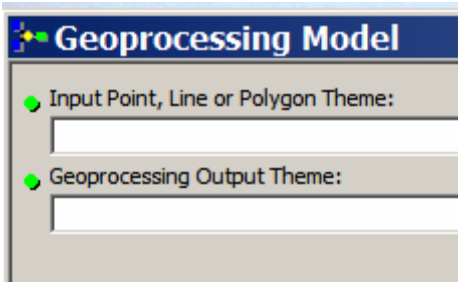
Geoprocessing Blunders

Dave Verbyla
Professor of GIS
Dept. of Forest Sciences,
University of Alaska Fairbanks
Email: D.Verbyla@uaf.edu

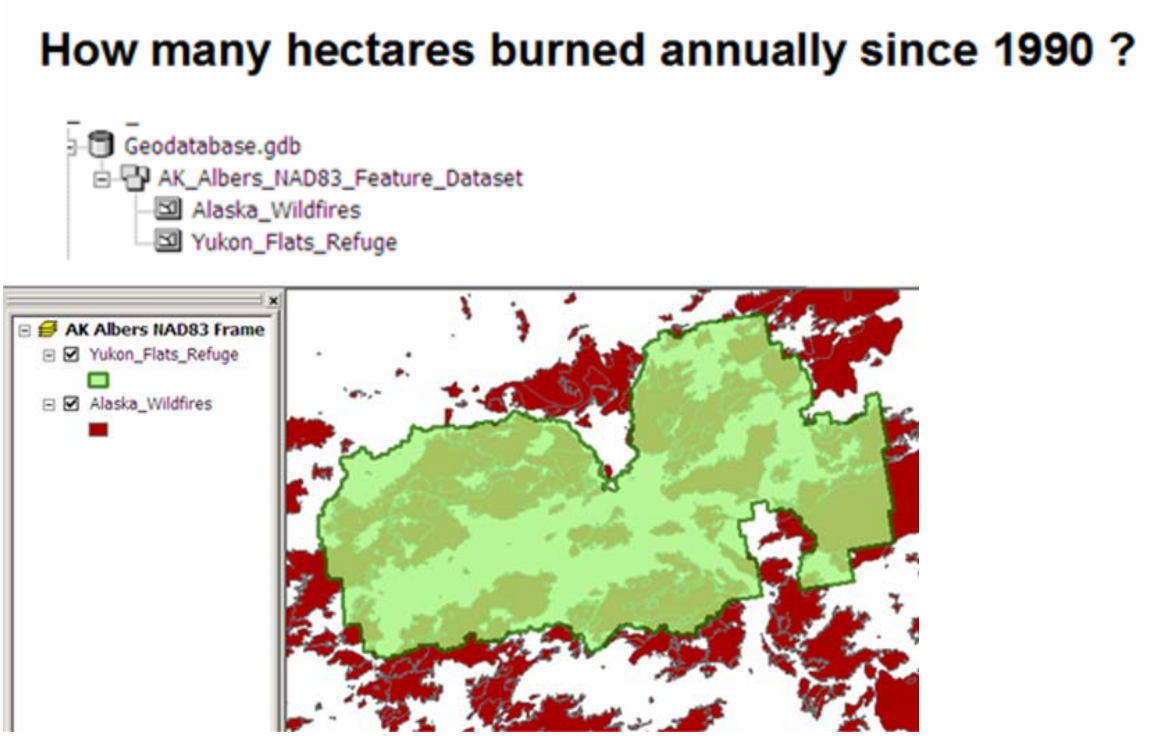
It is pretty easy to create GIS models that will allow anyone to perform a variety of GIS analysis tasks.



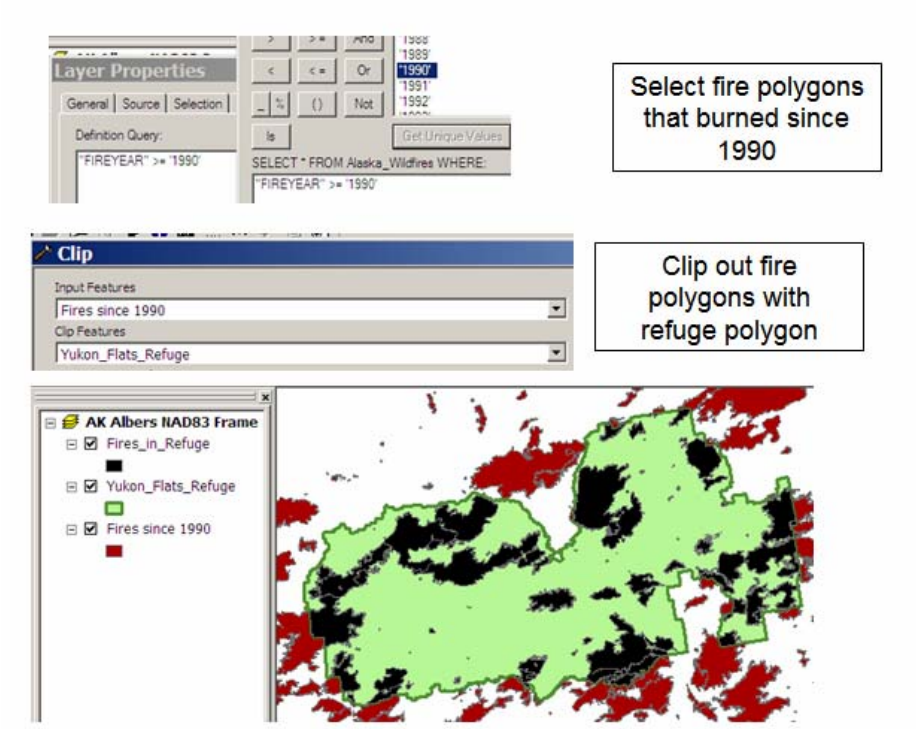
But beware of geoprocessing blunders! Potential geoprocessing blunders abound for novice users, especially for folks blindly using geoprocessing tools and models. GIS blunders can range from errors or bugs in GIS software to user errors due to incomplete understanding of GIS fundamentals. You can discover geoprocessor blunders by test driving a Geoprocessing tool with simple example data to test your Geoprocessing in a step by step manner. In this paper, I will use ArcGIS9.3 to illustrate some common GIS blunders using simple test case data.

GIS Software Bugs

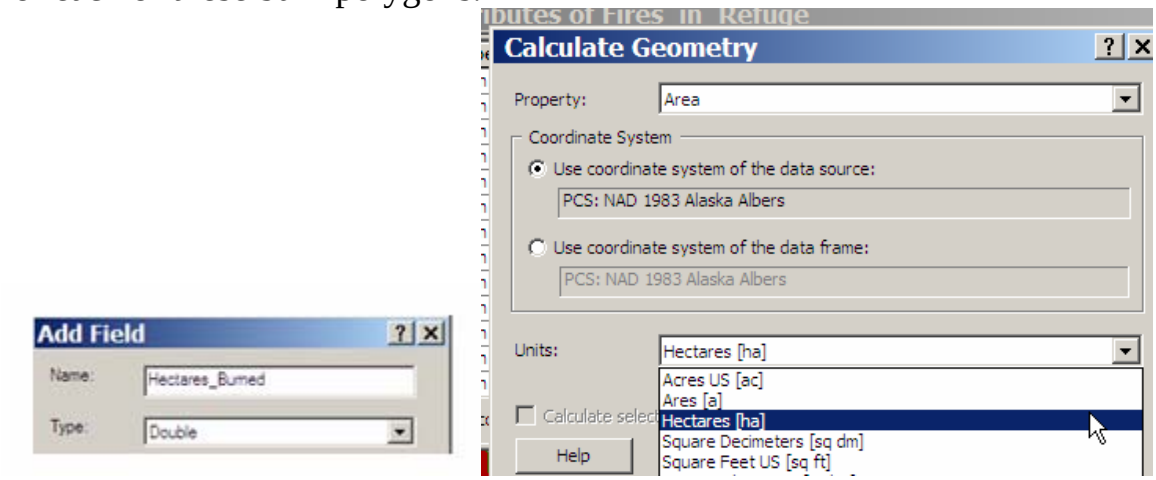
Blunders due to GIS software errors are not common, but do occur. Here is an example where we want to estimate the area burned in hectares within the Yukon Flats National Wildlife Refuge on an annual basis. We start with a GIS geodatabase of Alaska Wildfires and the boundary of the Yukon Flats National Wildlife Refuge with X,Y coordinates in meters.



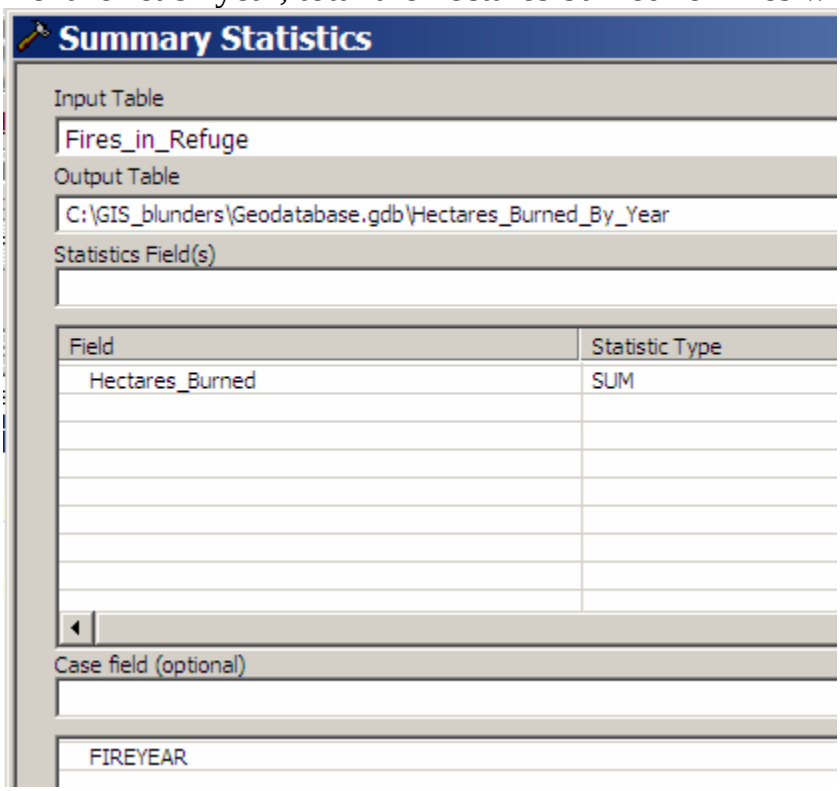
The first step in the analysis is to use a definition query to select all fires that burned since 1990 and clip out these fire polygons using the refuge boundary.



Next add a field in the polygon attribute table to contain hectares burned for each burn polygon within the refuge boundary. Then calculate the hectares for each of these burn polygons.



Next for each year, total the hectares burned for fires within the refuge:



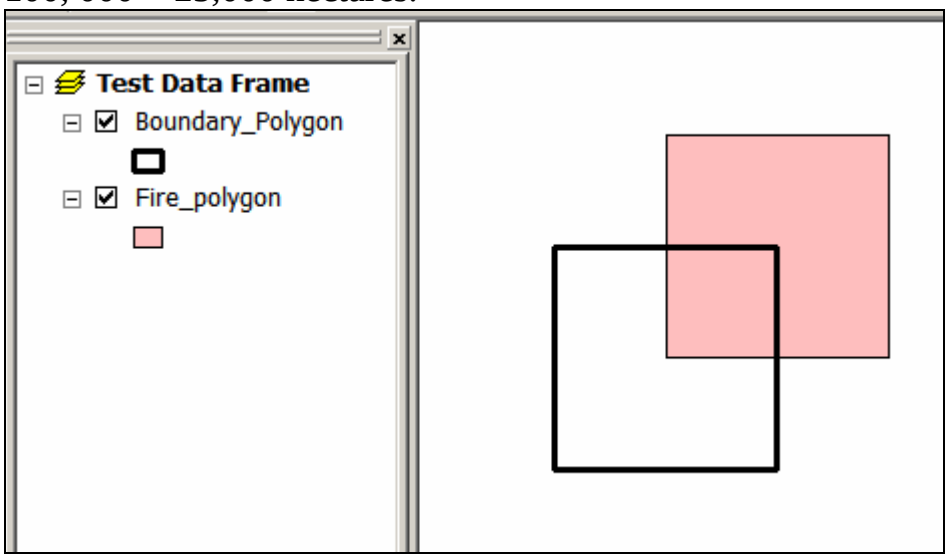
The resulting table is the result of a blunder in our analysis and the total hectares burned are tiny values!

Attributes of Hectares_Burned_By_Year

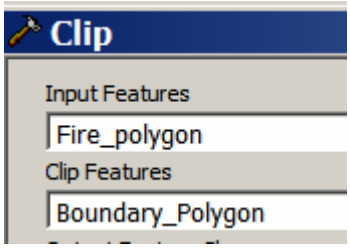
FIREYEAR	FREQUENCY	SUM_Hectares_Burned
1990	7	0.33876
1991	10	0.271019
1992	7	0.02594
1993	8	0.090088
1994	9	0.020065
1995	1	0.000287
1996	9	0.137908
1997	1	0.020219
1999	5	0.124926
2000	3	0.01114
2002	3	0.015477
2003	2	0.160103
2004	24	1.047969
2005	13	0.521705
2006	3	0.012186
2007	11	0.024309

Why and where did the blunder occur? The easiest way to discover the source of the blunder is to use a sample dataset where the exact areas are known.

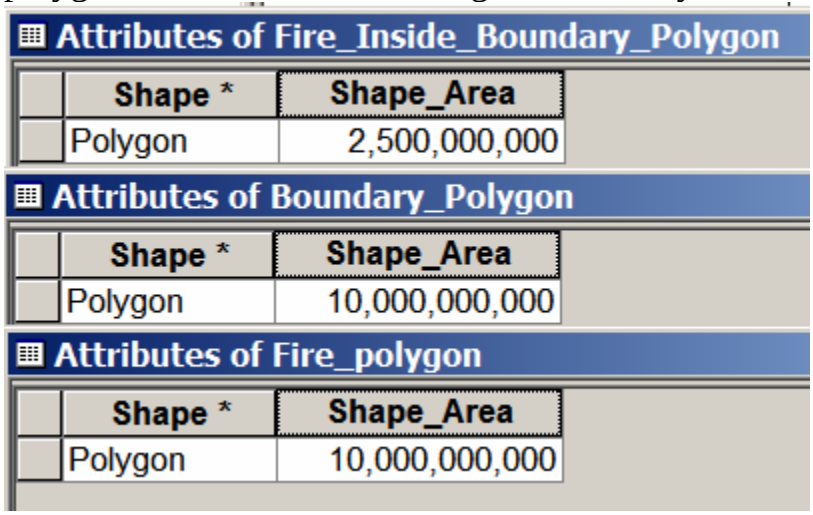
Here are polygons where each polygon is exactly100, 000 hectares in area and the portion of the fire polygon inside the boundary would be ¼ of 100, 000 = 25,000 hectares.

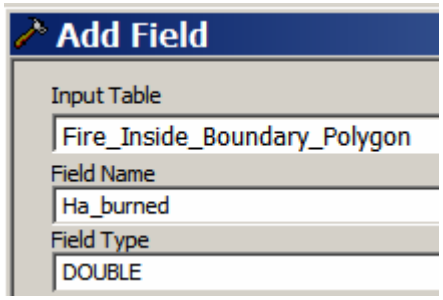


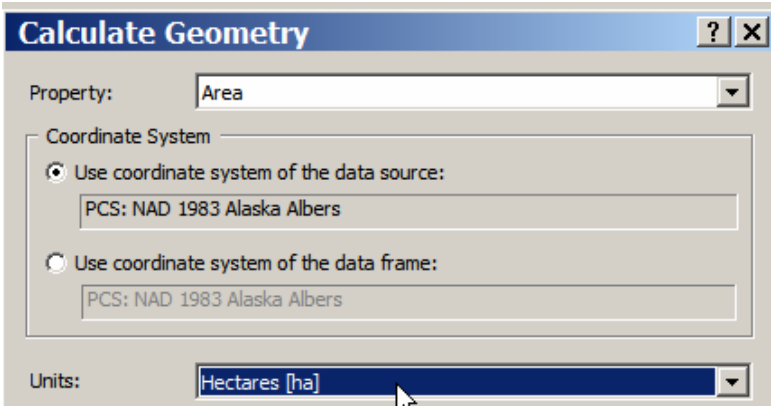
Using these simple polygons, we repeat the Geoprocessing steps to see where the blunder occurs…



So far, so good, *Shape_Area* is in m^2 and the fire area inside the boundary polygon is ¼ the area of the original boundary and fire polygons:







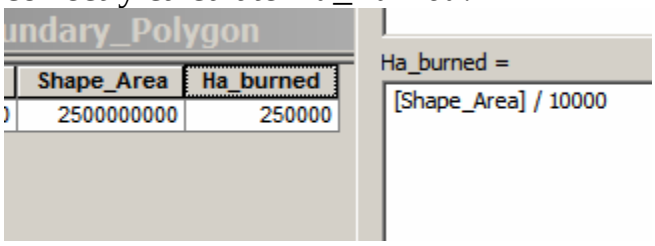
The polygon has an area of 2,500,000,000 m^2 and since there is 10,000 m^2 in one hectare, the area in hectares for this polygon is 250,000 hectares.

Attributes of Fire_Inside_Boundary_Polygon

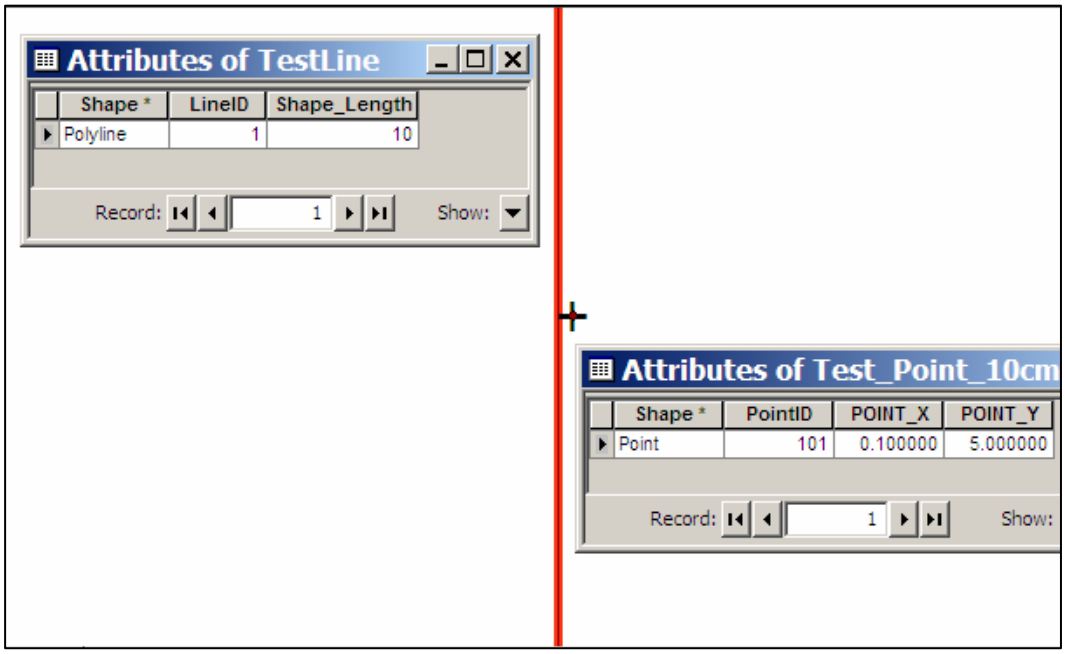
Shape *	Shape_Area	Ha_burned
Polygon	2500000000	0.513764

Thus there is a bug in the Calculate Geometry step in our analysis. The Hectares returned from Calculate Geometry is always incorrect because of this bug. (Fortunately, this bug was fixed in ArcGIS9.3 with service pack 1).

Usually there is a work-around that can solve the problem, once you discover the bug. In this example, you could use the *Shape_Area* field to correctly calculate *Ha_Burned*:

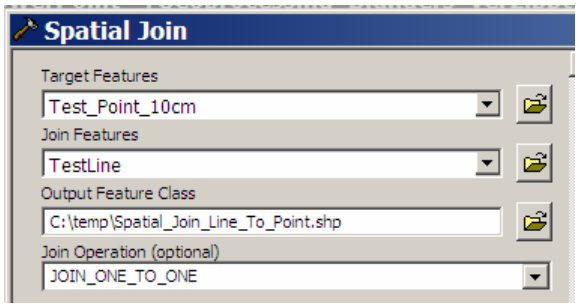


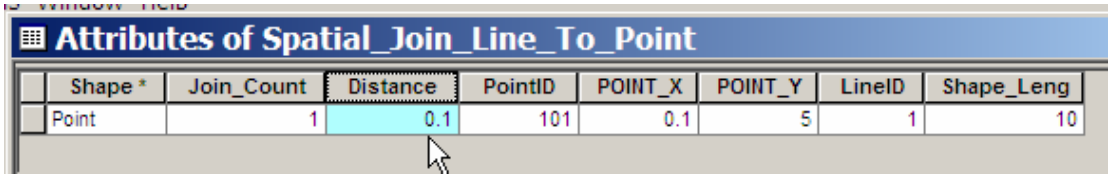
Sometimes there is a bug in one particular aspect of a Geoprocessing tool. For example, the Spatial Join tool can be used to transfer attributes and distance from the closest feature to each target feature. We test drive this tool with a simple test data set of a line and a point with known locations.



The line begins at X=0.0 meters, Y=0.0 meters and ends at X=0.0 meters, Y=10.0 meters. The point is exactly 0.10 meters away from the line.

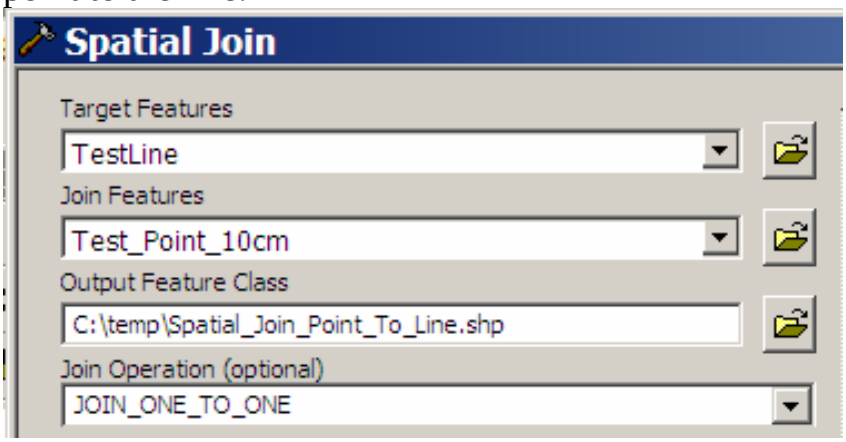
Using the Spatial Join tool, we want to find the closest feature and transfer that feature information to the target feature. So we try Spatial Joining the line information to the point.

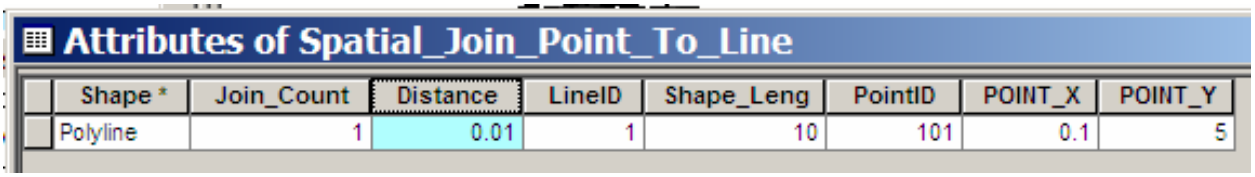




The tool works perfectly and returns a distance of 0.1 meters and the correct line information.

Next, we try the same tool, using the same data, spatially joining the closest point to the line.

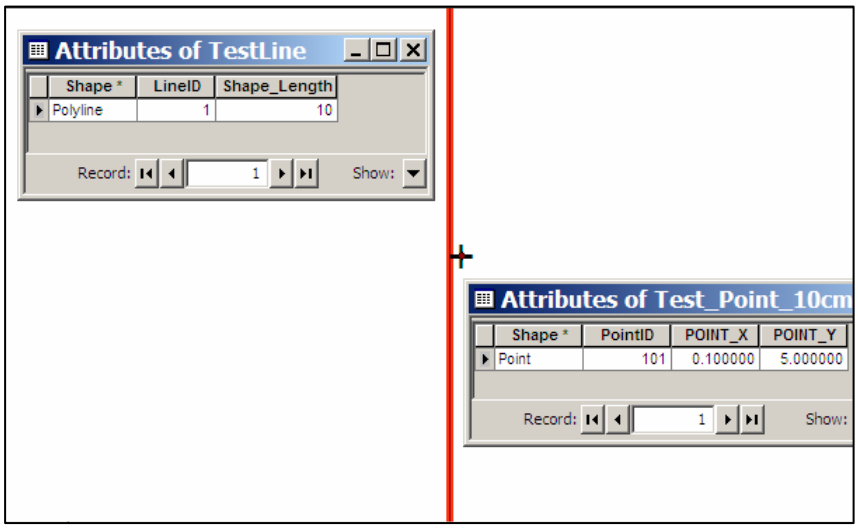




There is a bug in this tool, yielding an incorrect distance value. Thus the Spatial Join tool would work correctly if you wanted to estimate how far mailbox points are from streets and transfer the street information to each mailbox point. However, it would yield incorrect distances if you applied the tool to estimate the distance an eroding river bank shoreline is away from cabin locations.

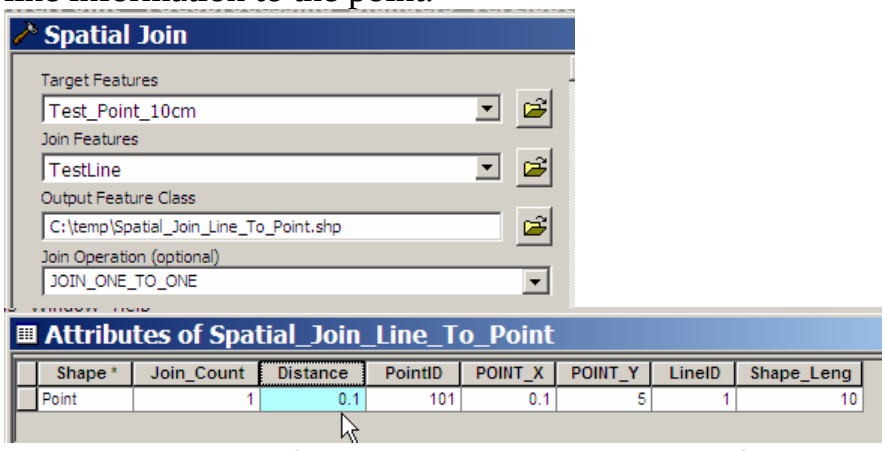
Bugs Depending On Geoprocessing Settings

Sometimes a Geoprocessing tool works perfectly with certain settings, but has a bug with other settings. To discover bugs, it pays to test drive the Geoprocessing tool with simple features that have a known answer. For example, we create a line and point as follows:



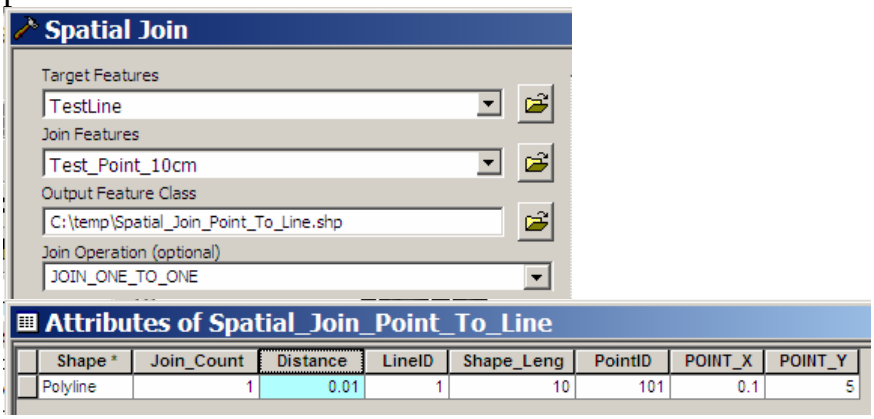
The line begins at X=0.0 meters, Y=0.0 meters and ends at X=0.0 meters, Y=10.0 meters. The point is exactly 0.10 meters away from the line.

Using the Spatial Join tool, we want to find the closest feature and transfer that feature information to the target feature. So we try Spatial Joining the line information to the point.



The tool works perfectly and returns a distance of 0.1 meters and the correct line information.

Now try the same tool, using the same data, only spatially joining the closest point to the line.



Attributes of Spatial_Join_Point_To_Line

Shape *	Join_Count	Distance	LineID	Shape_Leng	PointID	POINT_X	POINT_Y
Polyline	1	0.01	1	10	101	0.1	5

There is a serious bug here…a distance of 0.01 (1 cm) is returned instead of the correct distance of 0.10 (10cm). So the Spatial Join tool works joining lines to points, but has a bug joining points to lines.

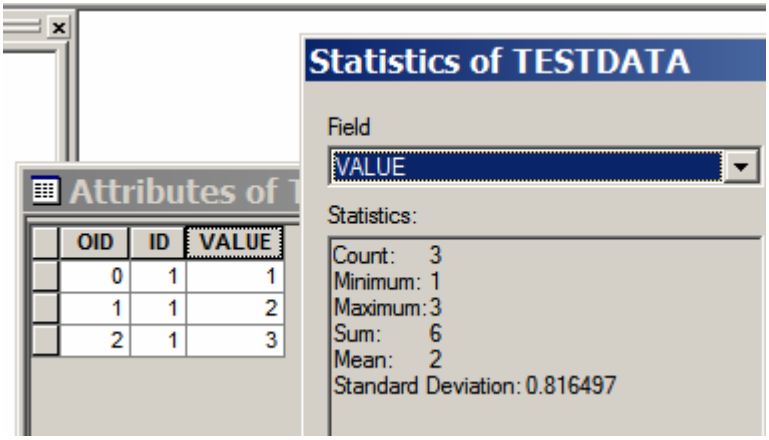
GIS Software Not Exactly Correct

Sometimes the results from GIS Geoprocessing are not exactly correct, which may or may not be significant, depending upon your application.

Here is an example, the standard deviation of a sample is often used in basic statistics…. $: \sqrt{\frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{N-1}}$

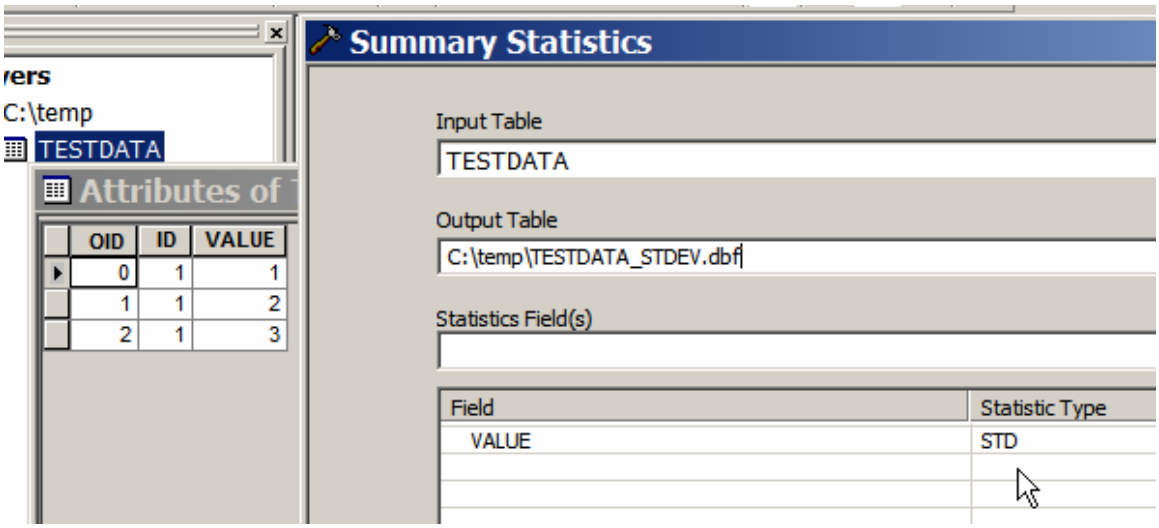
Thus for sample values of (1,2,3) the standard deviation of the sample is 1.0

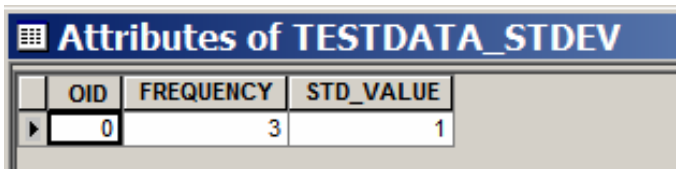
Yet if we rely on field statistics in ArcGIS, it always returns an incorrect sample standard deviation of 0.816 instead of 1.0:



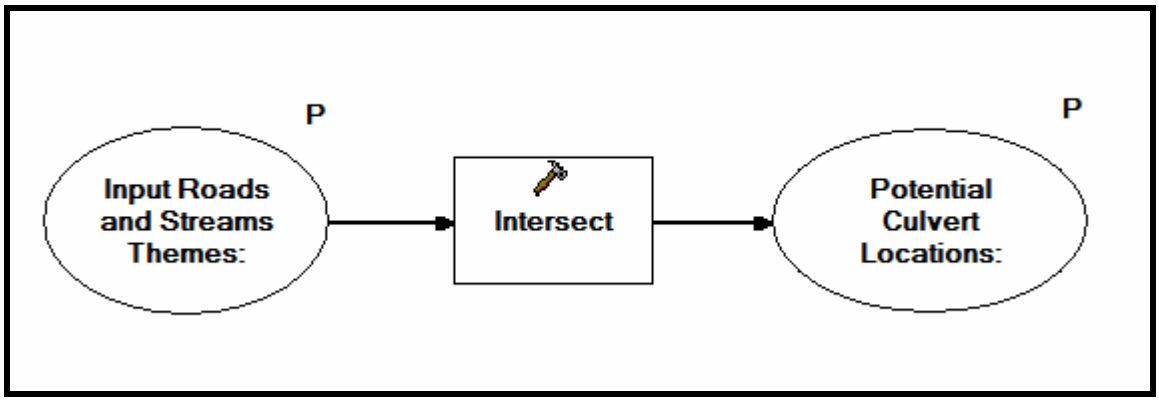
Thus if we used this standard deviation value to compute confidence intervals or standard errors, our sample statistics would be wrong.

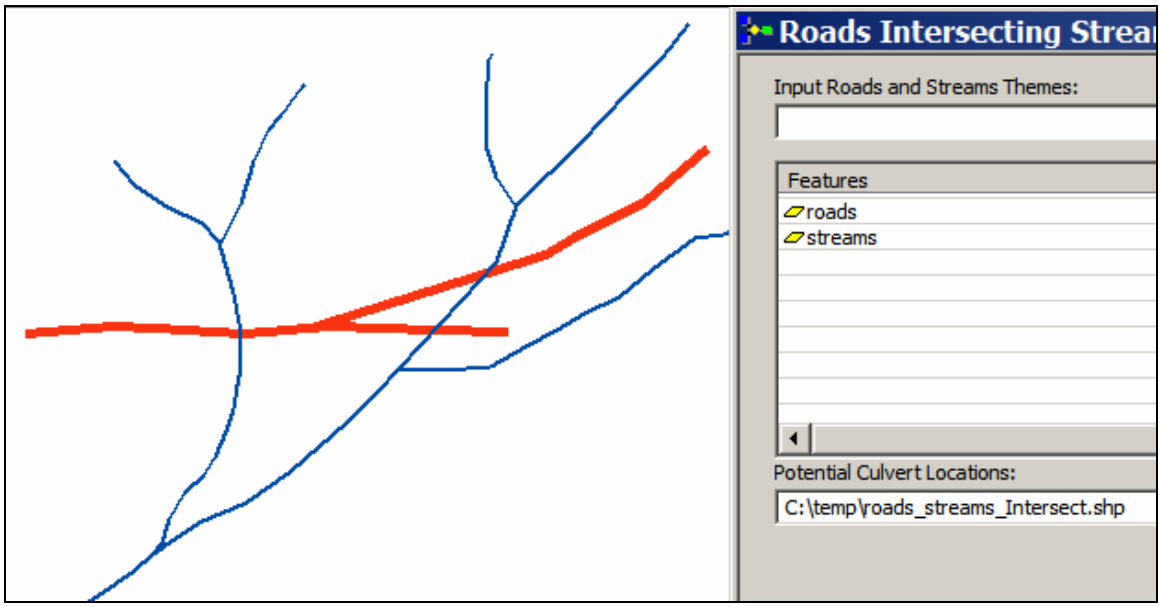
Once again, after you discover a blunder by using a simple example, you can usually find a solution to the problem. For example, the ***Summary Statistics*** tool correctly returns the sample standard deviation, so you can always use that tool.



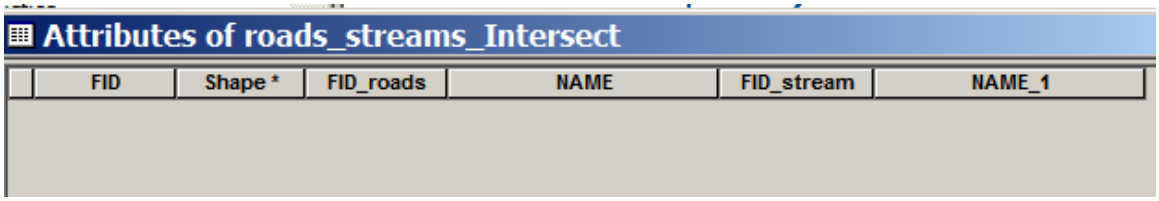


Here is another example. This model that attempts to find locations where roads intersect streams…locations that should be checked for culverts.

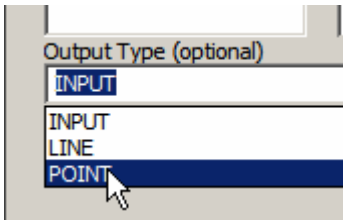


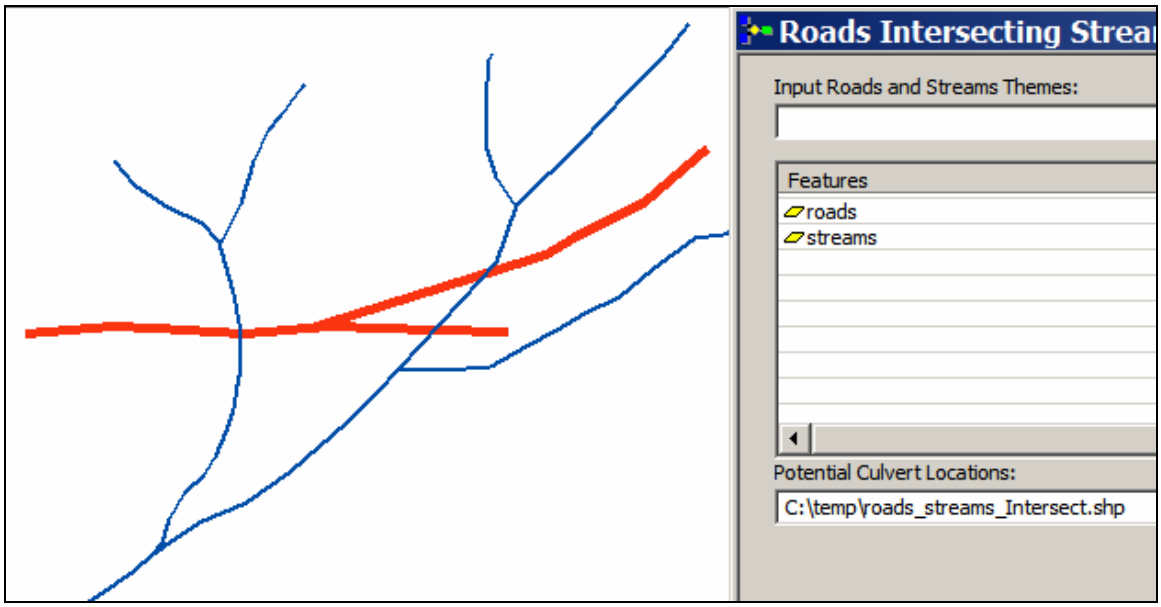


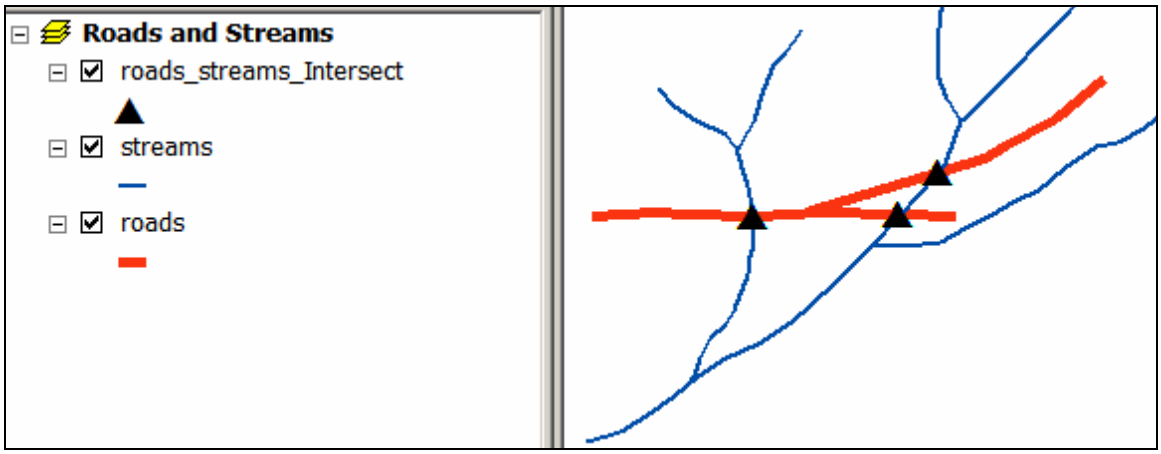
Yet the model returns no intersections!



By definition a line has length, but no width. Therefore the only time the Intersect tool will return the intersection of two lines is when they share the same line segments. So the solution in our model is to specify the output to be points (not the same type as input).

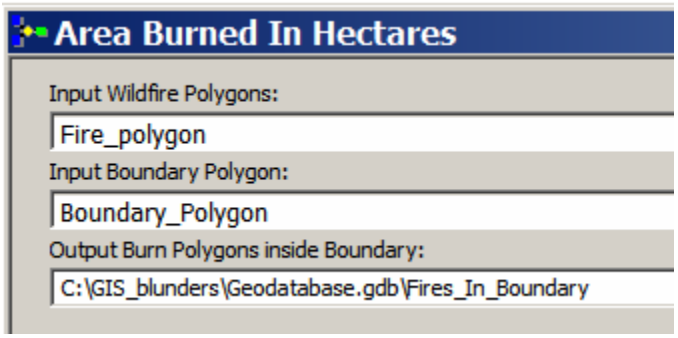



Roads Intersecting Strea
Input Roads and Streams Themes:
Features
roads
streams
Potential Culvert Locations:
C:\temp\roads_streams_Intersect.shp

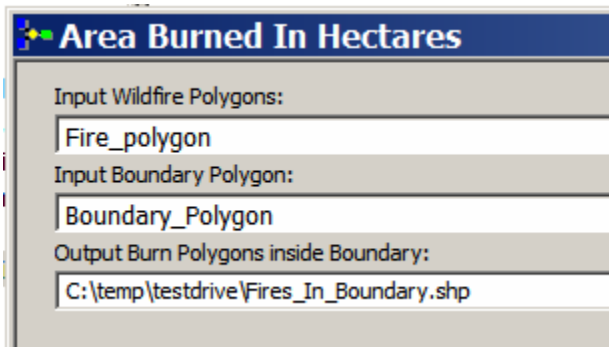

Roads and Streams
roads_streams_Intersect
streams
roads

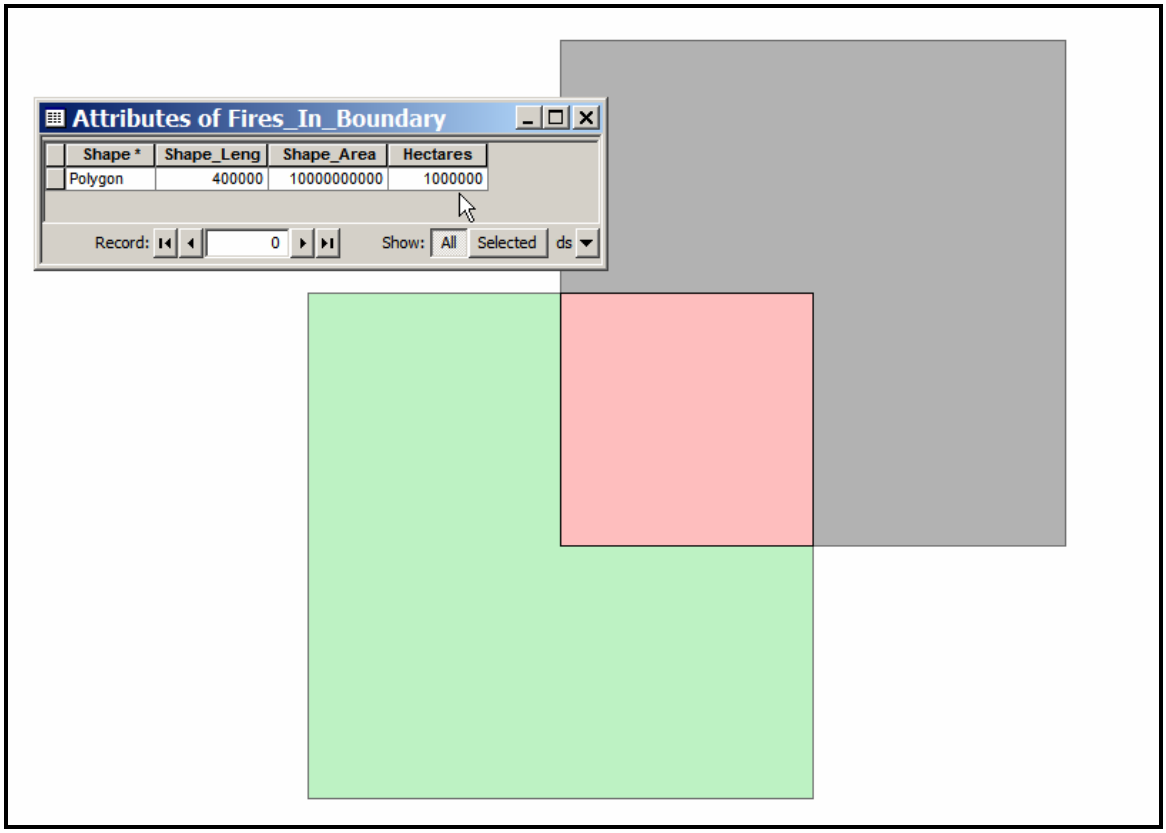
Blunders Depending On User Data

A third type of blunder is one that occurs inconsistently and varies depending on the GIS data being used. For example, here is a model to determine the area in hectares burned within any boundary polygon theme. We test drive it to make sure it works correctly…

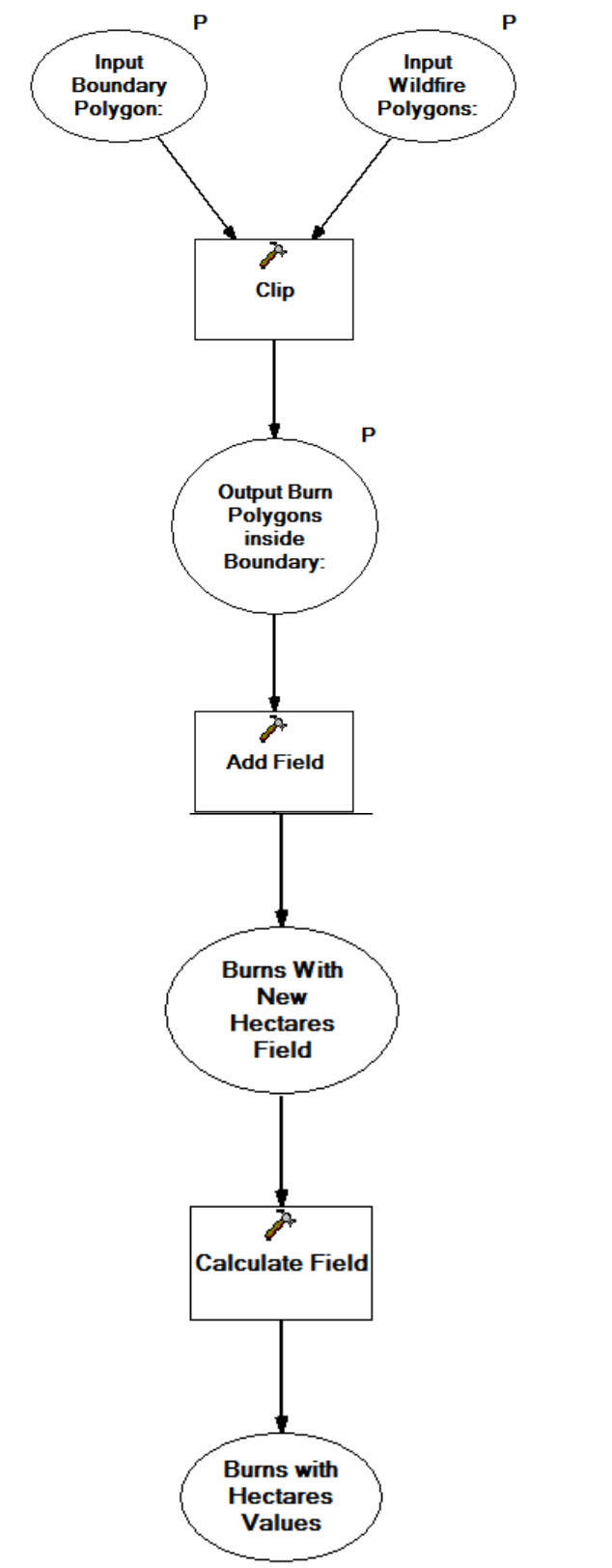


Yet another user test-drives the model and gets an incorrect answer:

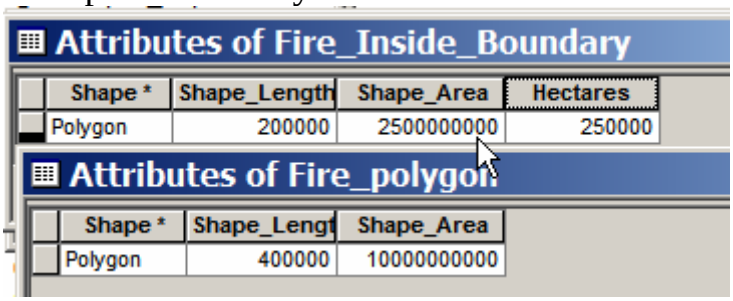




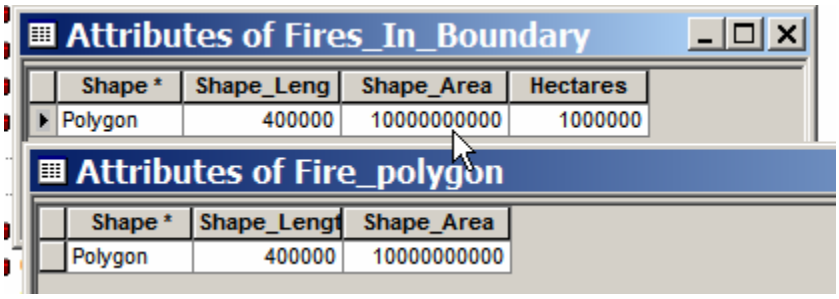
The model ***Clips*** the fire polygons using the Boundary Polygon, then Adds a ***Field*** named Hectares, then ***Calculates*** Hectares = Shape_Area / 10000


P
Input Boundary Polygon:
P
Input Wildfire Polygons:
Clip
P
Output Burn Polygons inside Boundary:
Add Field
Burns With New Hectares Field
Calculate Field
Burns with Hectares Values

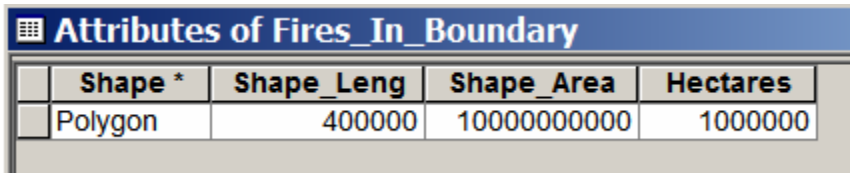
The results depend upon the data…if outputting to a geodatabase, ArcGIS updates the feature geometry and the *Shape_Length* and *Shape_Area* values are updated after the Clip and thus they are correct.



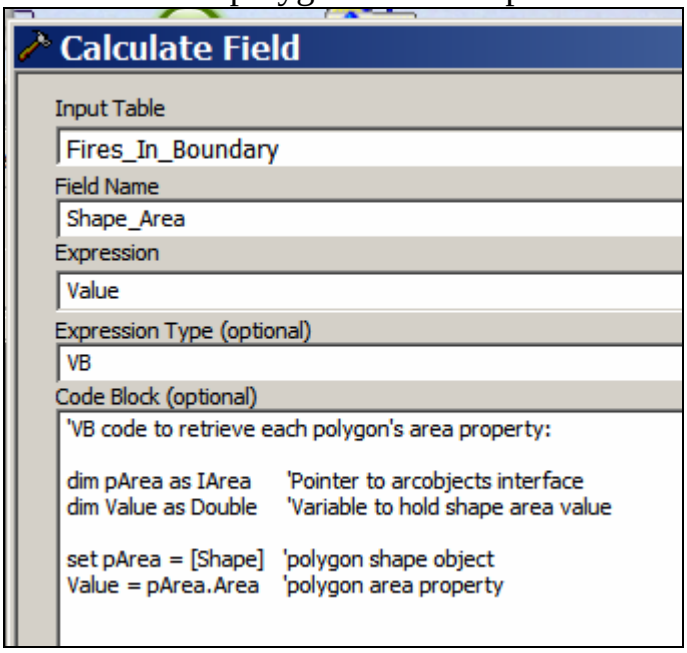
If outputting outside a geodatabase, the *Shape_Length* and *Shape_Area* values are not updated after Clipping and thus the field calculation Hectares = Shape_Area / 10000 is incorrect.



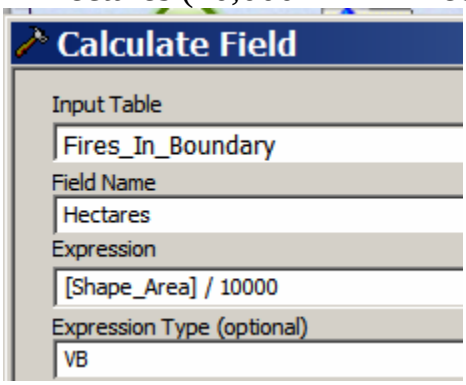
This problem can be solved by always retrieving the invisible area property from the [Shape] field when conducting your field calculation. For example, the three columns all contain incorrect values, since they were not updated after the fire polygon was clipped with the boundary polygon:

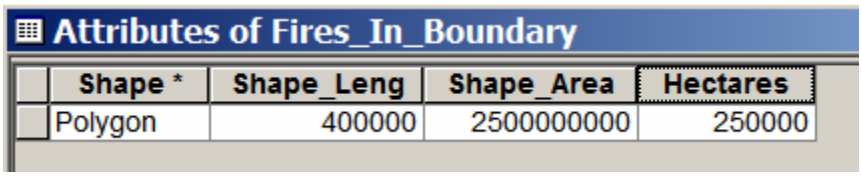


So retrieve the polygon's true shape area and update the Shape_Area field…

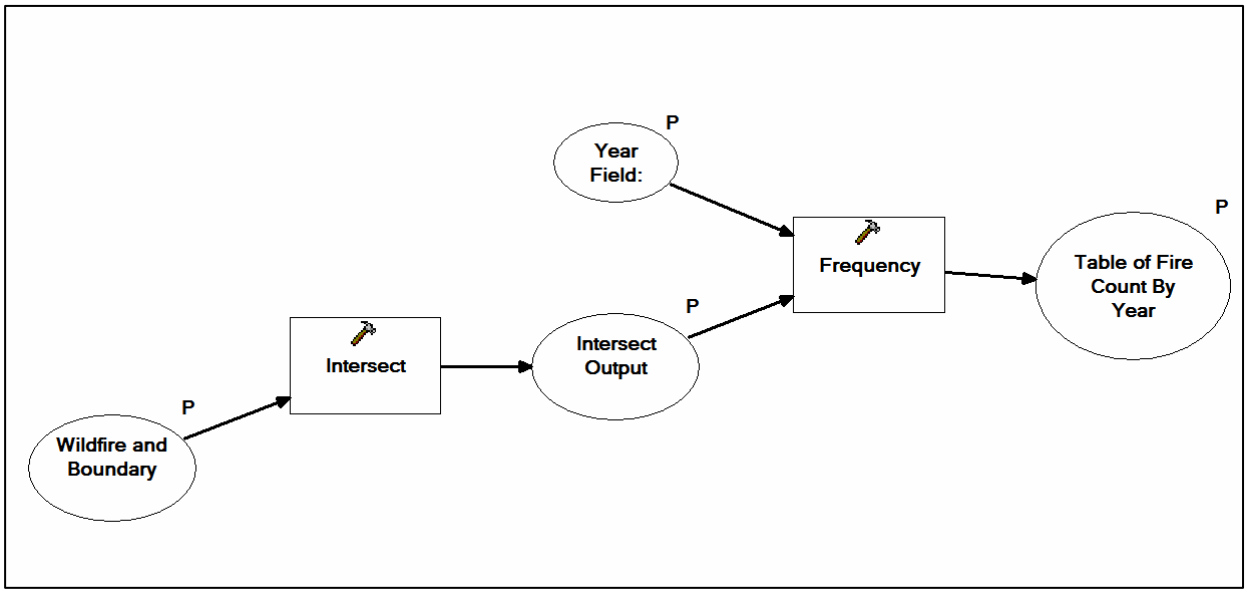


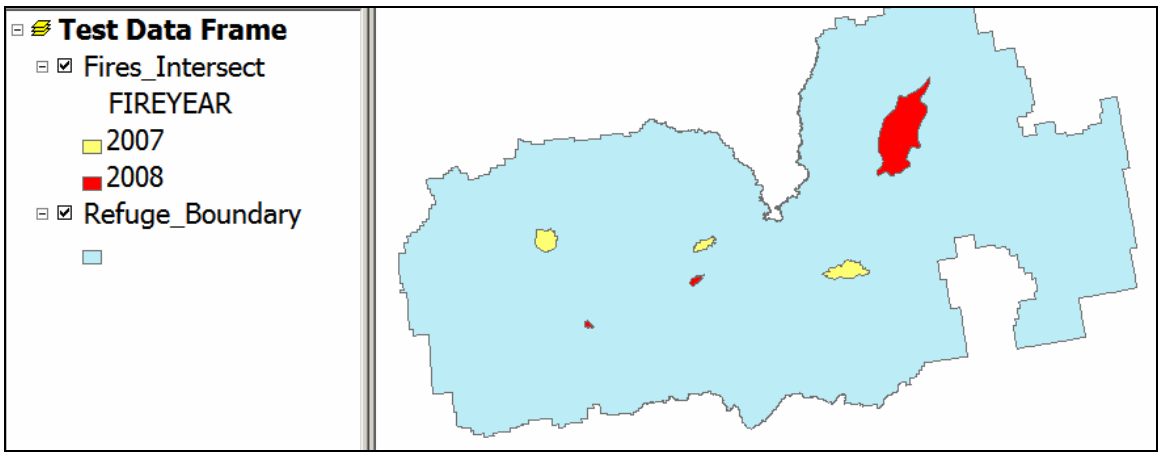
And then use the correct Shape_Area field to compute the correct polygon area in Hectares (10,000 m^2 = 1 Hectare):





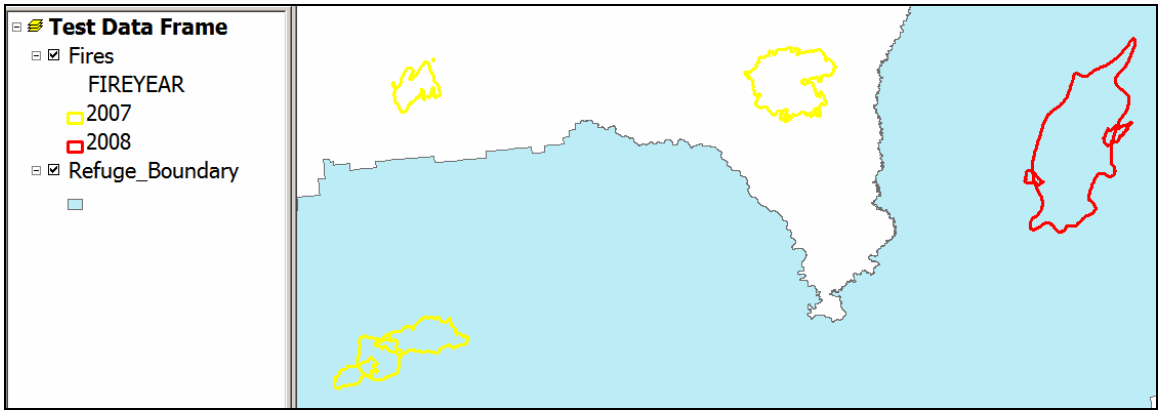
Here is a model to determine how many fires burned annually within a refuge boundary. It is correct depending upon the data used.





FREQUENCY	FIREYEAR
3	2007
3	2008

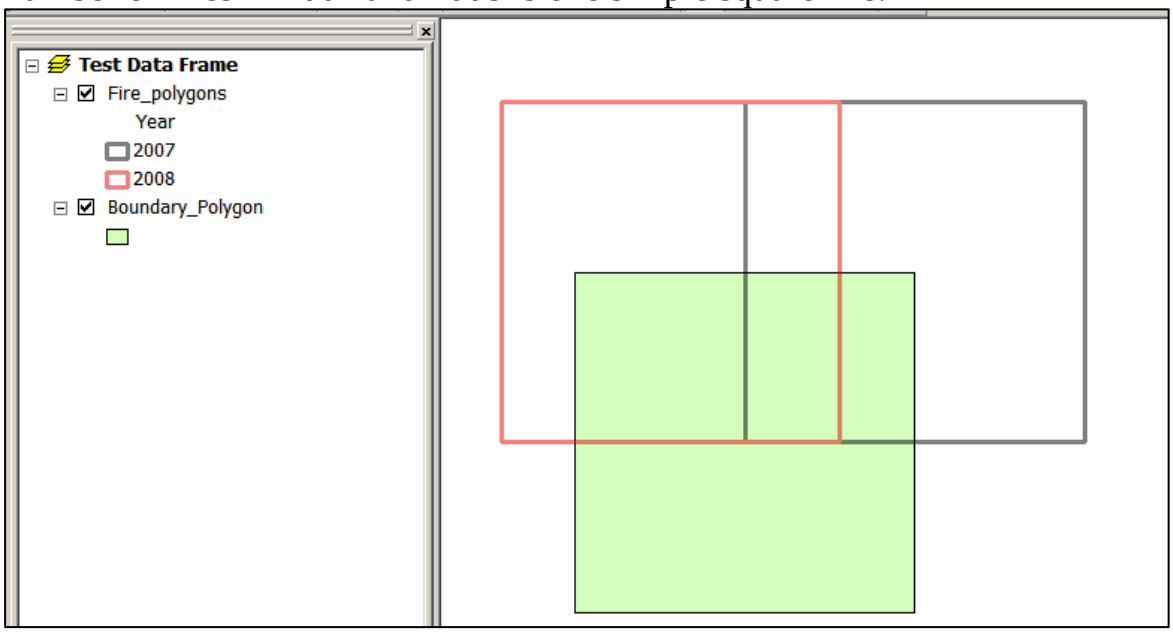
But if we move the fire polygons so that they overlap, will the model return correct number of fires? No!



FREQUENCY	FIREYEAR
7	2007
7	2008

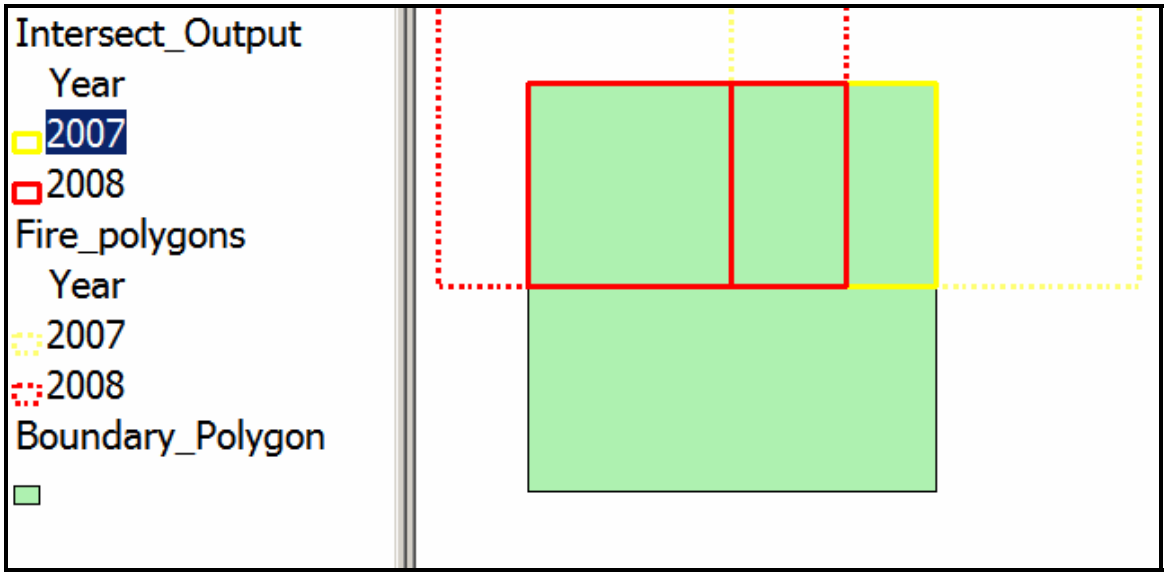
The Intersect tool creates a fire polygon for each original fire polygon inside the refuge boundary plus each overlap between any fire polygons in the refuge. Thus the model grossly overestimates the number of fires within the refuge for any given year.

Using simple squares as polygons, we can explore this problem. The number of fires in 2007 and 2008 is one simple square fire.

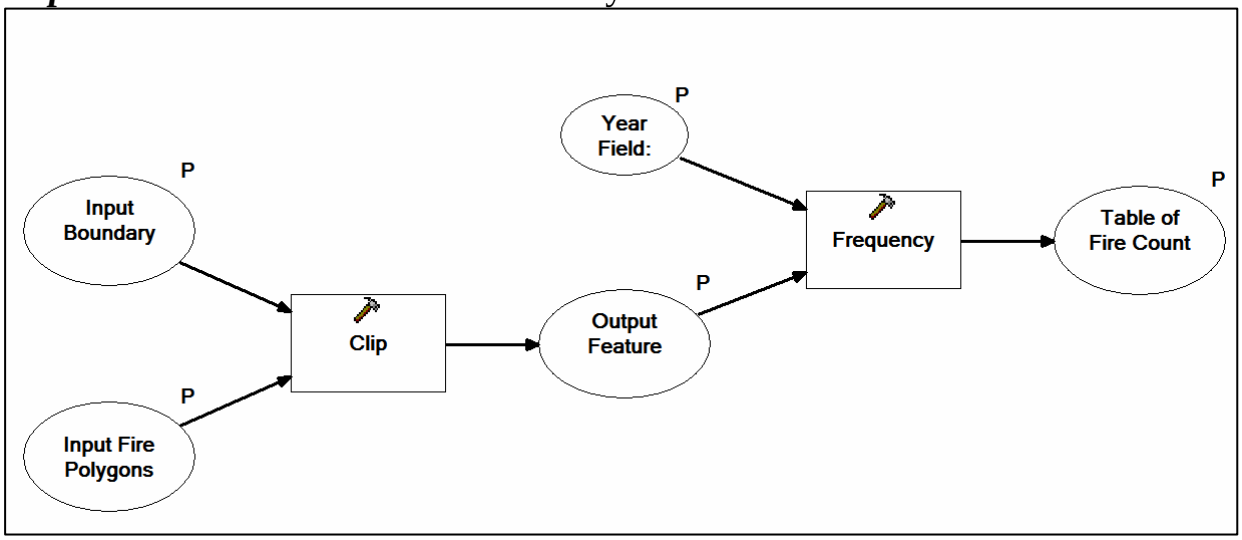


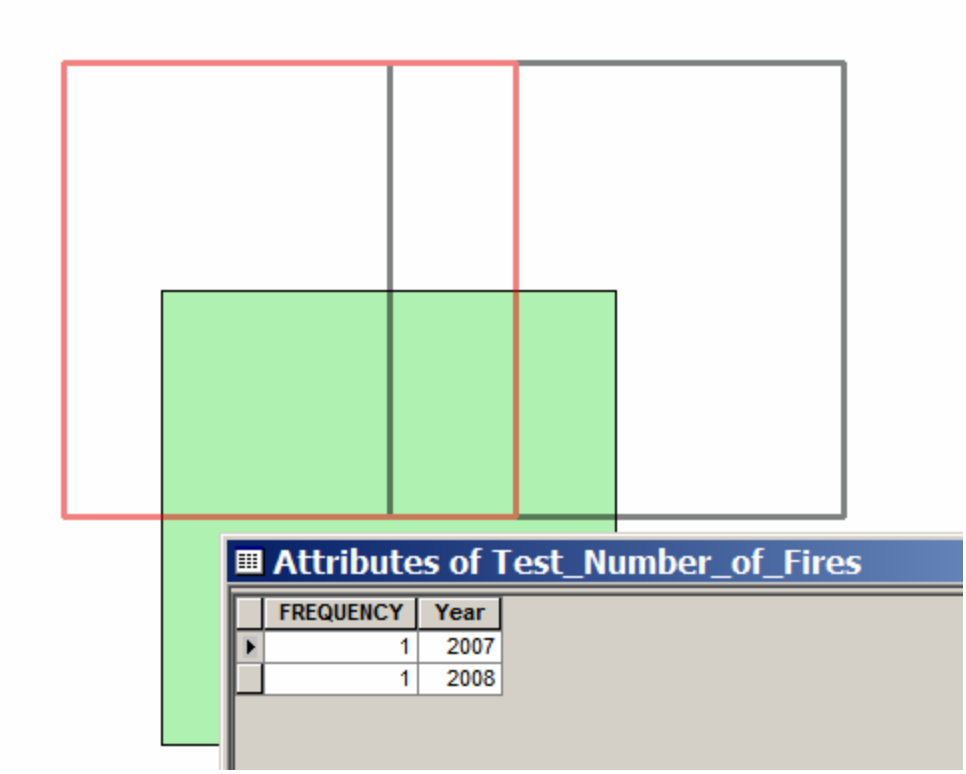
Yet the Intersect tool returns 4 fire polygons within the refuge boundary: each of the original fire polygon, but now as 2 fire polygons: the area that burned in the year and overlaps with another fire polygon, plus the area that burned in the year and did not overlap with another fire polygon.

FREQUENCY	Year
2	2007
2	2008

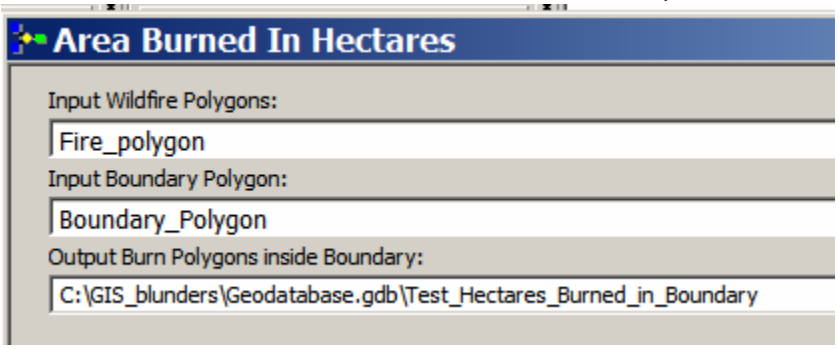


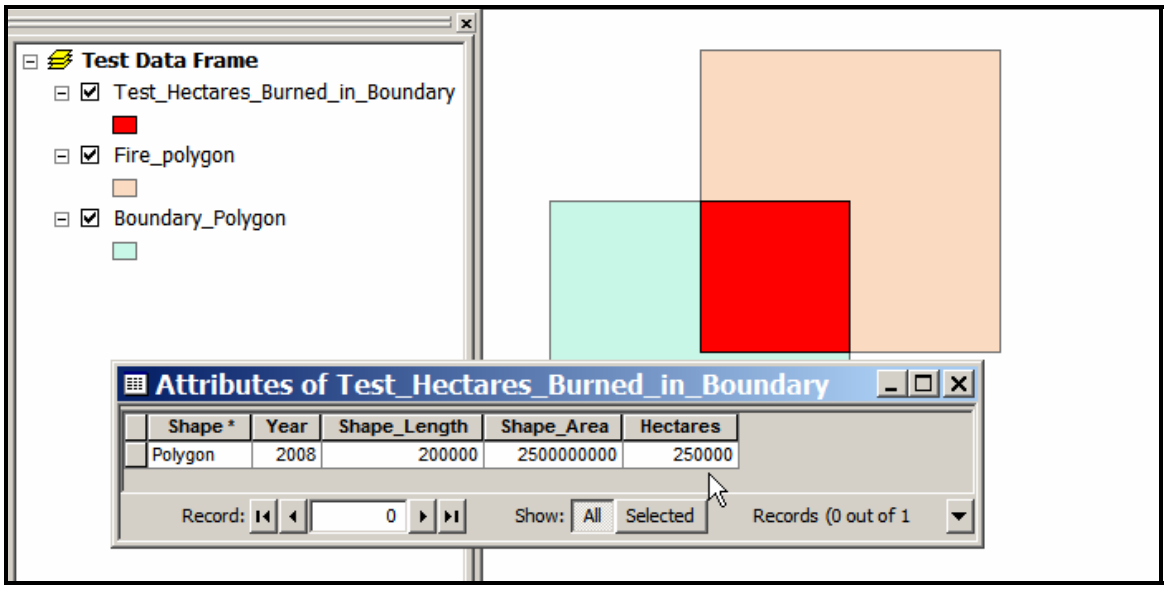
To avoid this blunder involving overlapping polygons, you could use the ***Clip*** tool instead of the ***Intersect*** tool in your model.





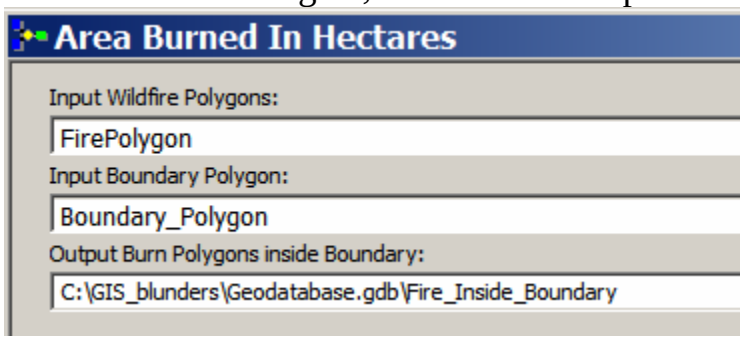
Here is one final example, where a blunder occurs depending on the data. Our model sometimes returns correct results, but in this example it does not.

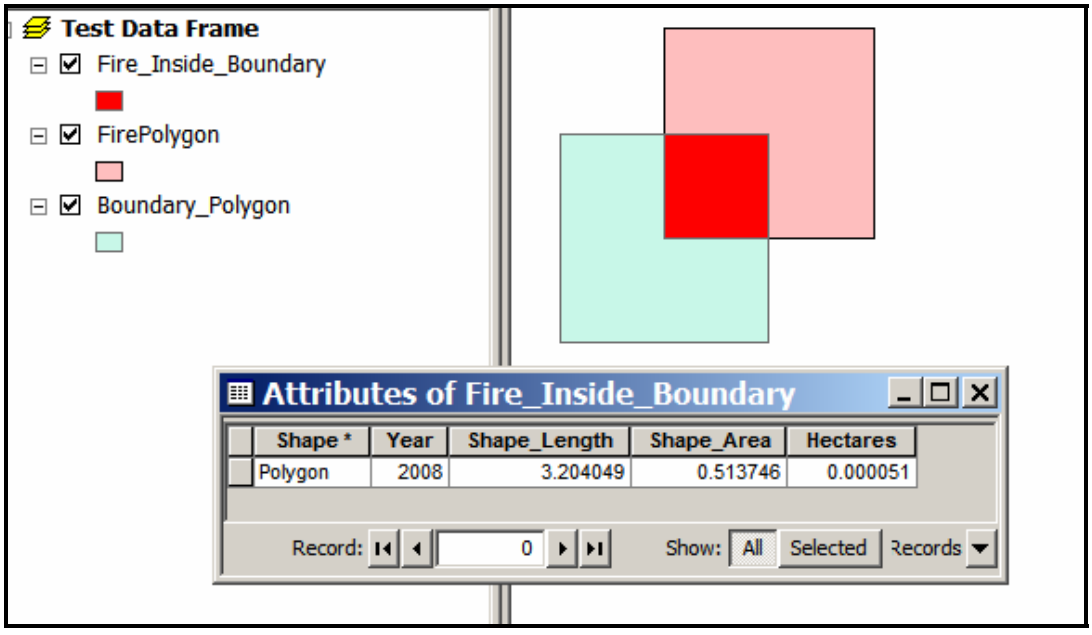




In the above example, the model output is correct.

We run the model again, this time the output is incorrect…





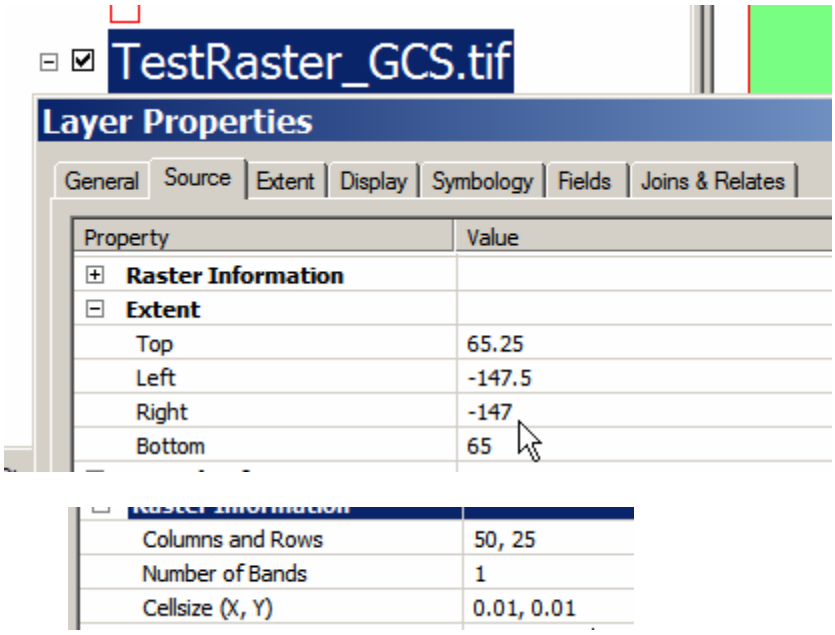
The geoprocessor inherits the coordinate system of the first polygon theme and outputs to that coordinates system. In the second example, the *FirePolygon* was actually in longitude/latitude and therefore the model outputs the *Fire_Inside_Boundary* in longitude/latitude rather than meters, even though the *Boundary_Polygon* coordinates are in meters. Thus the model will always return the correct hectares as long as both input polygon themes are in meters, and it will always be incorrect if the fire polygons are not in meters.

Raster Issues

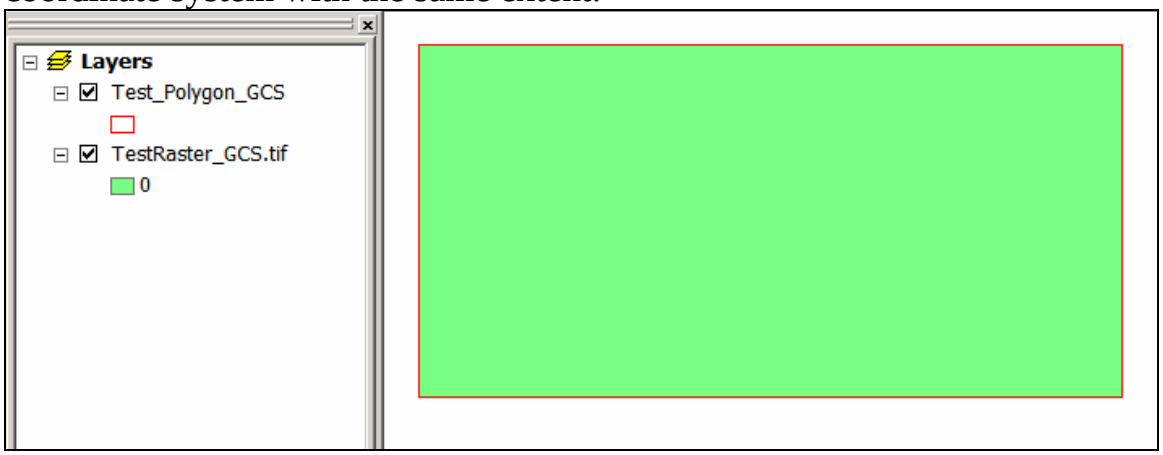
Rasters (images, grids, surfaces, etc) are fundamentally different than features (points, lines, and polygons). Because of this fundamental difference, geoprocessing tools may output different results from features versus raster data.

For example, we create a test polygon with four corners in longitude/latitude as follows

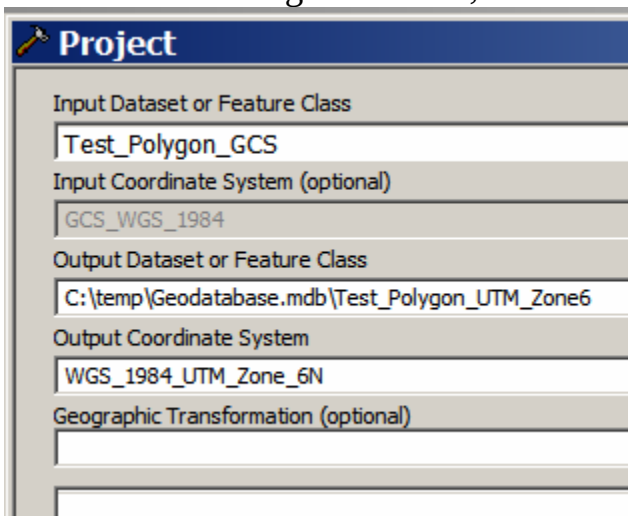
We then use this test polygon to create a test raster using the ***Polygon to Raster t***ool.

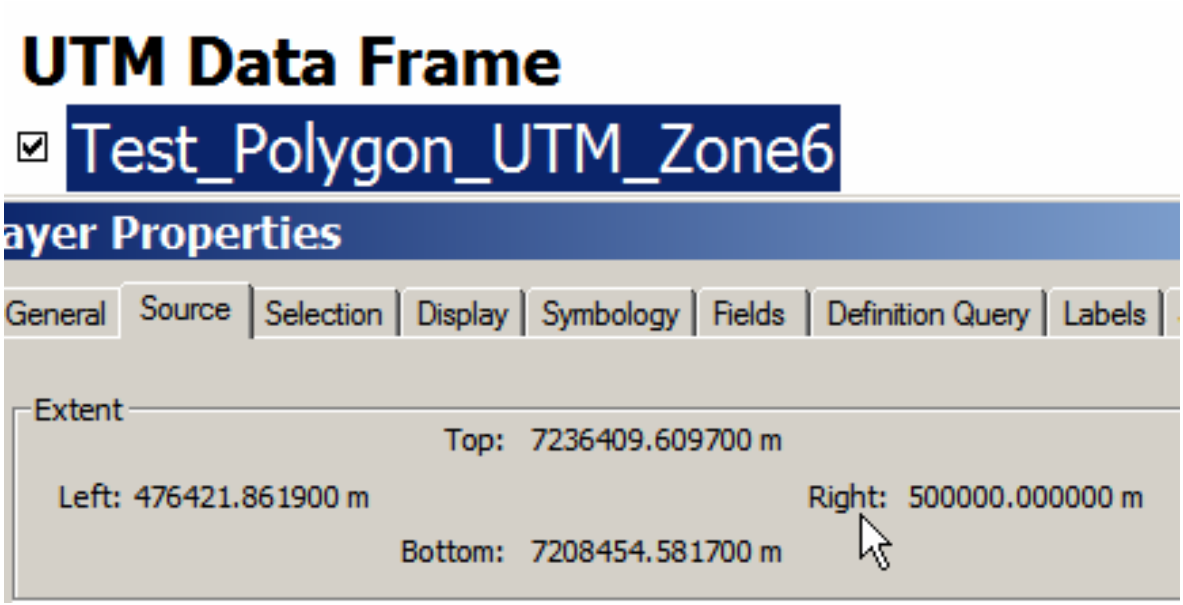


The test polygon and test raster are both in the same GCS WGS84 coordinate system with the same extent:



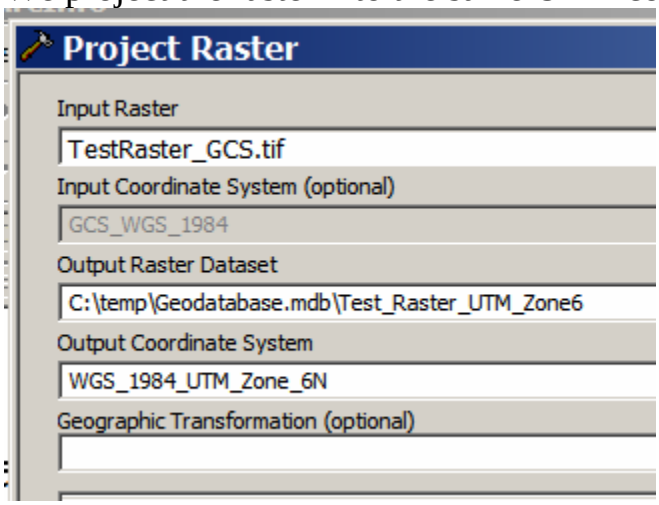
When projected to UTM Zone 6 WGS84, the polygon has the correct extent with the eastern edge at X=500,000 meters (center of UTM Zone 6).

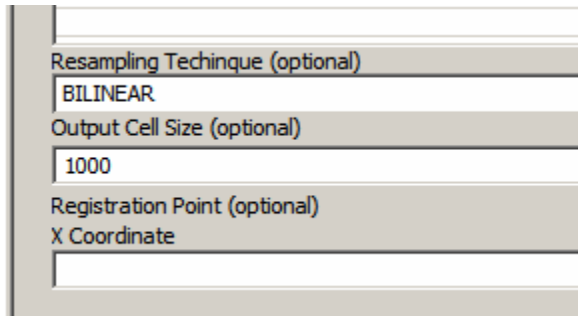




Perfect results extent!

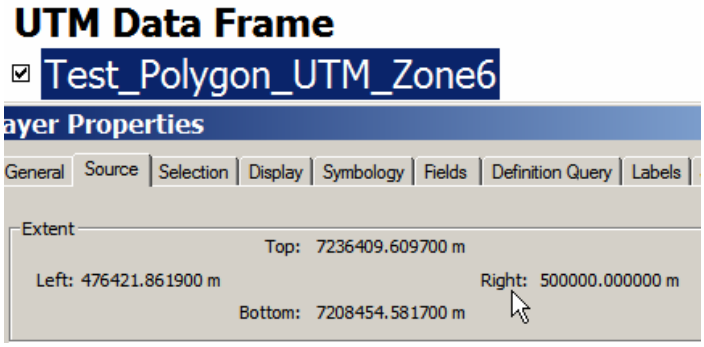
We project the raster into the same UTM coordinate system:





When projected to UTM Zone 6 WGS84, the raster does not have the correct extent with the eastern edge at X=500,000 meters (center of UTM Zone 6)… it is off by over 421 meters!

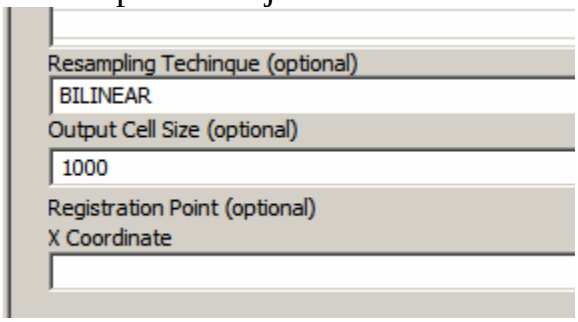
Does this mean there is a major bug in the Project Raster tool? No.
The polygon width and length are as follows:



Width = 500000 - 476421.861900 = **23578.1381 meters**

Length = 7236409.609700 - 7208454.581700 = **27955.028 meters**

The output of Project Raster tool was 1000 meter pixels.



23578.1381 / 1000 = 23.5781381 pixels

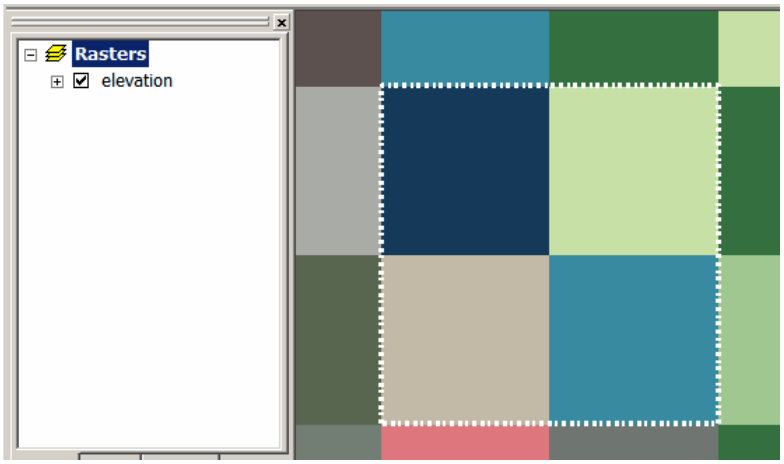
27955.028/1000 = 27.955028 pixels

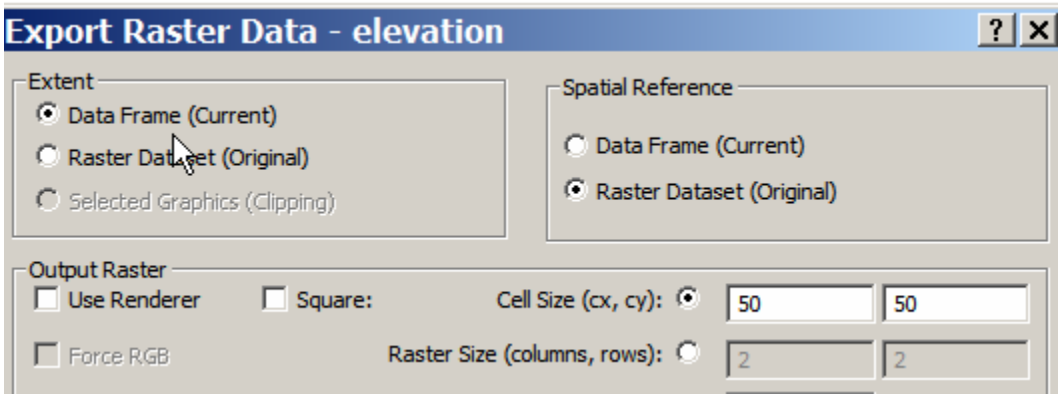
So the Project Raster tool had to output in whole pixels: 24 and 28 pixels….

perty	Value
Raster Information	
Columns and Rows	24, 28
Number of Bands	1
Cellsize (X, Y)	1000, 1000

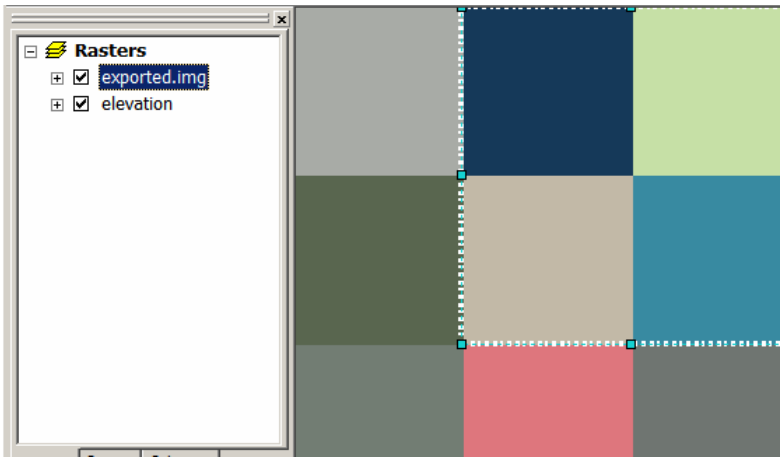
Thus output raster is forced to be wider and longer than polygon that is exactly correct.

Here is another example with the same basic problem. Here are 4 pixels displayed by zooming in on a large raster….we want to export these 4 pixels that are in the current data frame extent.





The exported raster is not identical, but is shifted in position in an attempt to fill in the data frame extent:



Conclusions

Always be cautious when using GIS tools for spatial analysis…the potential for Geoprocessing blunders exists and may have major consequences. Beware of software bugs, especially when upgrading to a new version of the software. Always test your analysis with simple GIS data so that you know the correct values to compare GIS output to. GIS blunders may occur inconsistently because of data issues such as polygons overlapping versus not overlapping, input themes from different coordinate systems, etc. A wise GIS analyst is a cautious GIS analyst.