Calculate Geometry Using Python

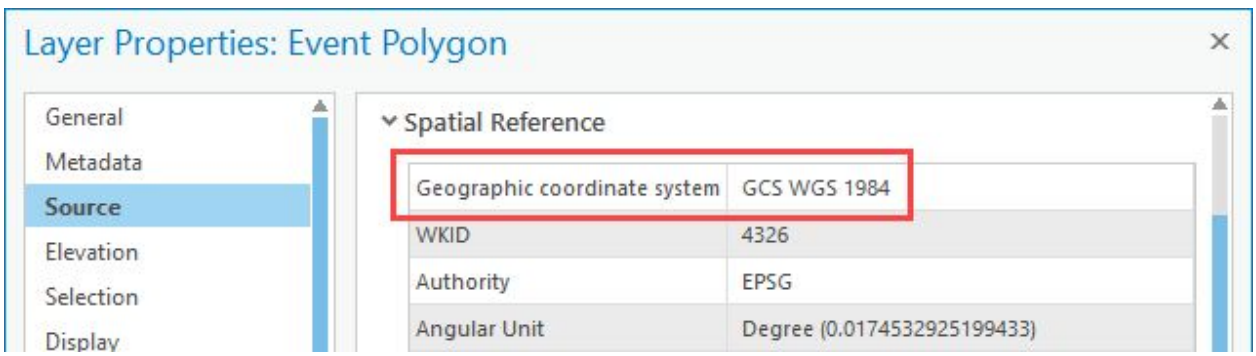
ArcGIS Pro

The Calculate Geometry function is only available in ArcGIS Pro 2.2+. When using an older version to calculate feature geometries to existing fields, a Python expression must be used.

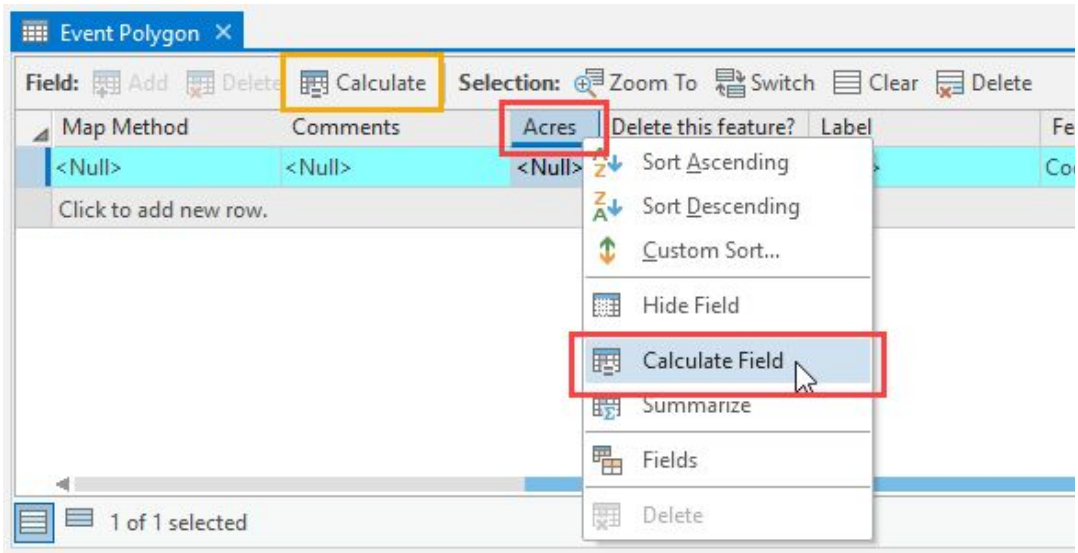
Calculate Acreage

These calculations will depend on the spatial reference of the *feature class, not the data frame*.

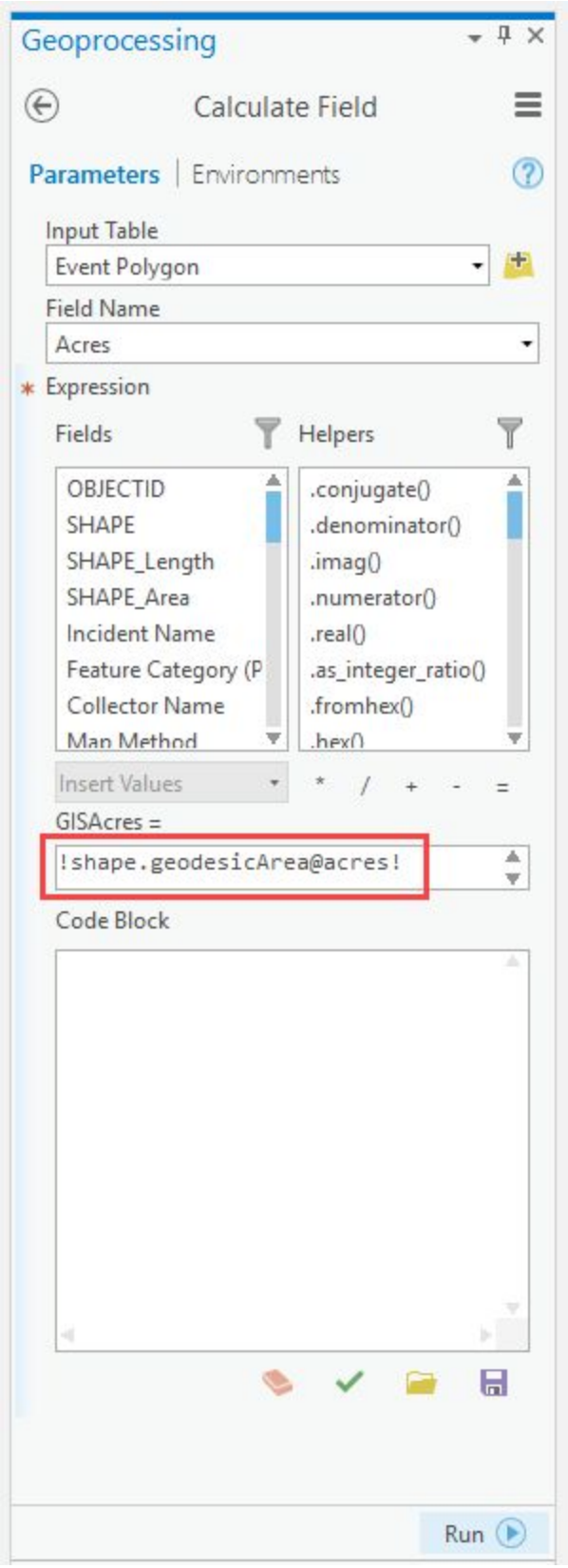
1. Open the layer properties by right-clicking the map in the table of contents and selecting Properties.
2. Check the layer Spatial Reference to be sure it has the **WGS84 Geographic** Coordinate System.



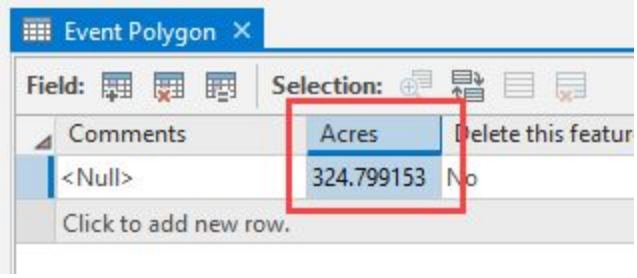
3. Open the Event Polygon attribute table, right-click the Acres field, and select Calculate Field. Alternatively, click Calculate and select Acres as the field name in the Geoprocessing pane.



4. Enter **!shape.geodesicArea@acres!** into the expression field and click Run.



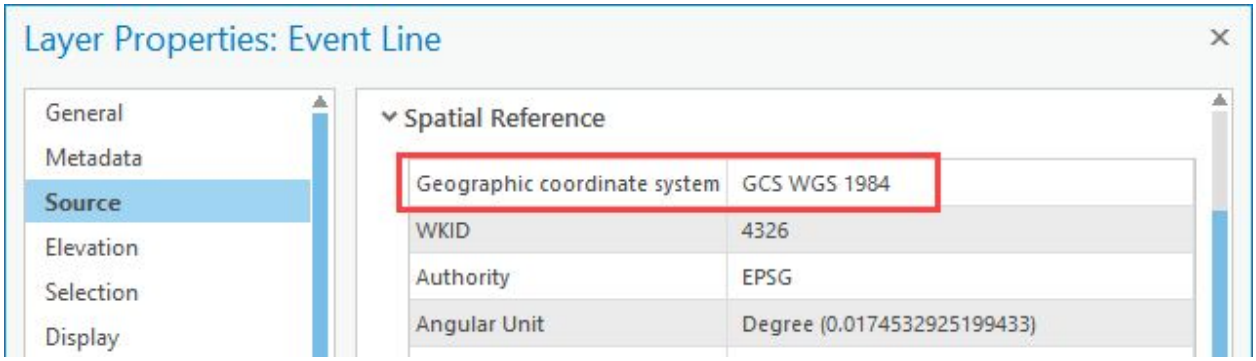
5. The Acres field will now be populated.



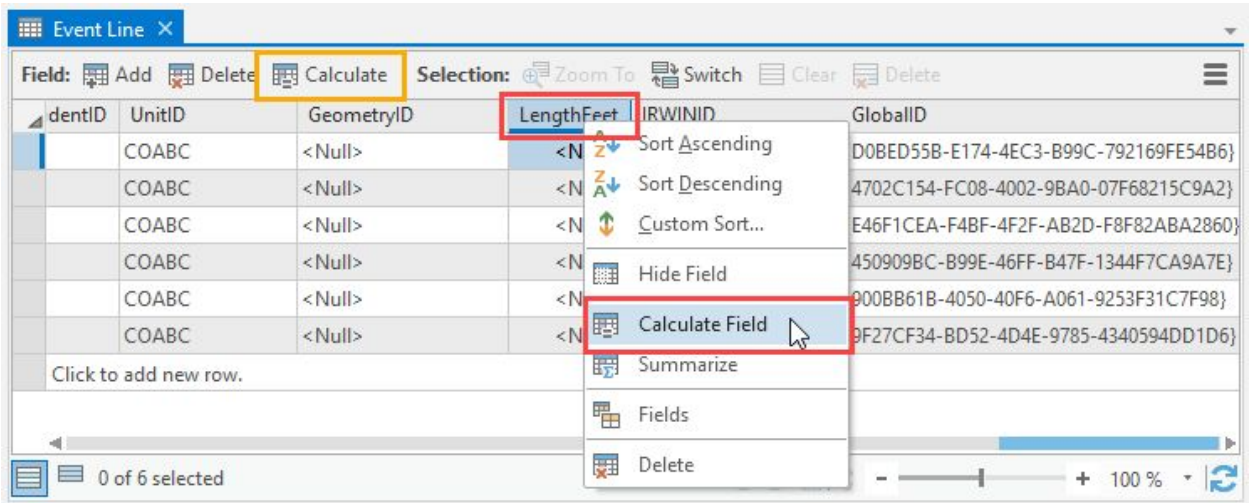
Calculate Line Lengths

These calculations will depend on the spatial reference of the *feature class, not the data frame*.

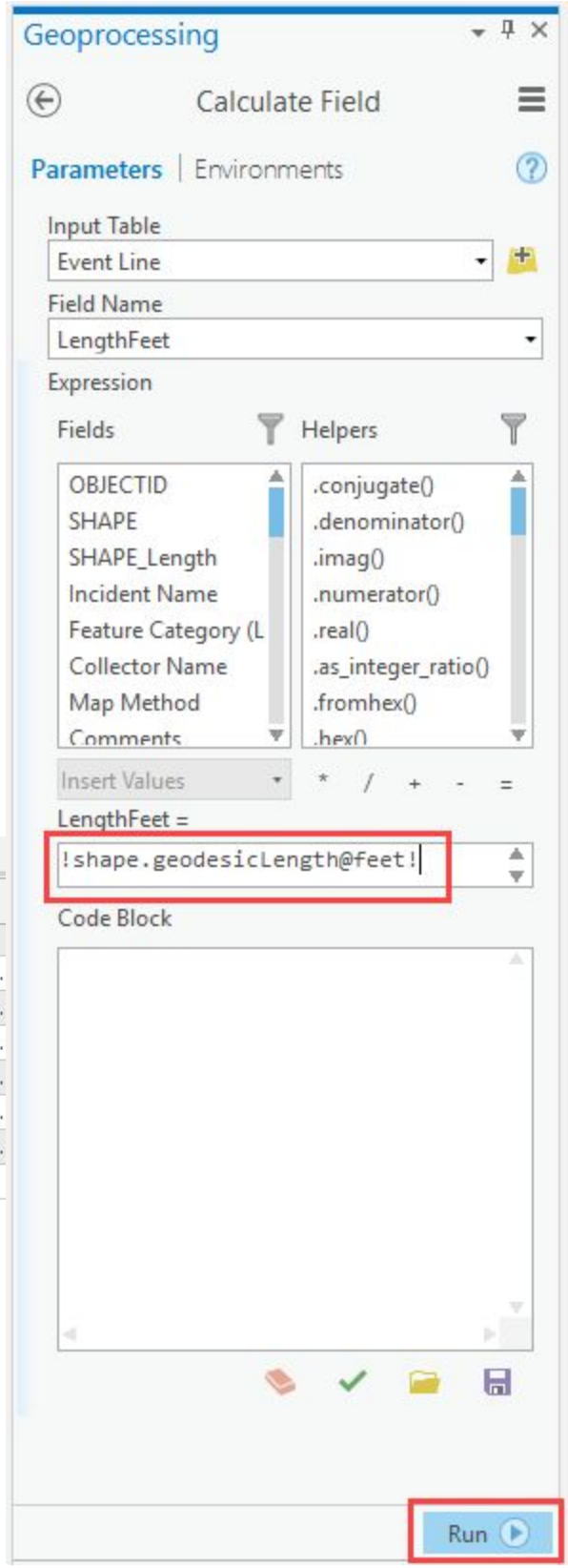
1. Open the layer properties by right-clicking the map in the table of contents and selecting Properties.
2. Check the layer Spatial Reference to be sure it has the **WGS84 Geographic** Coordinate System.



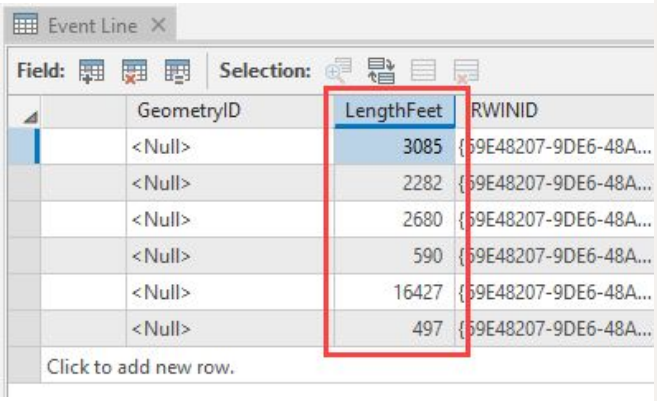
3. Open the Event Line attribute table, right-click the LengthFeet field, and select Calculate Field. Alternatively, click Calculate and select LengthFeet as the field name in the Geoprocessing pane.



4. Enter **!shape.geodesicLength@feet!** into the expression field and click Run.



5. The LengthFeet field will now be populated.

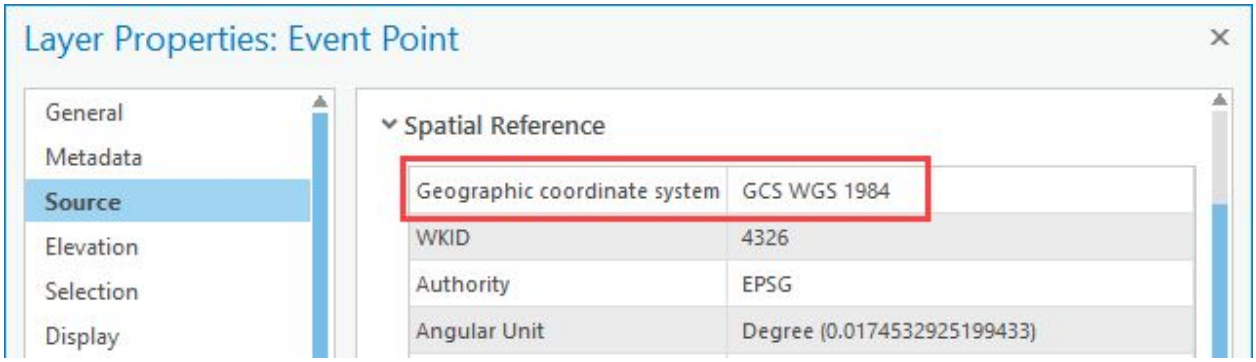


Calculate Latitude and Longitude for Point Features

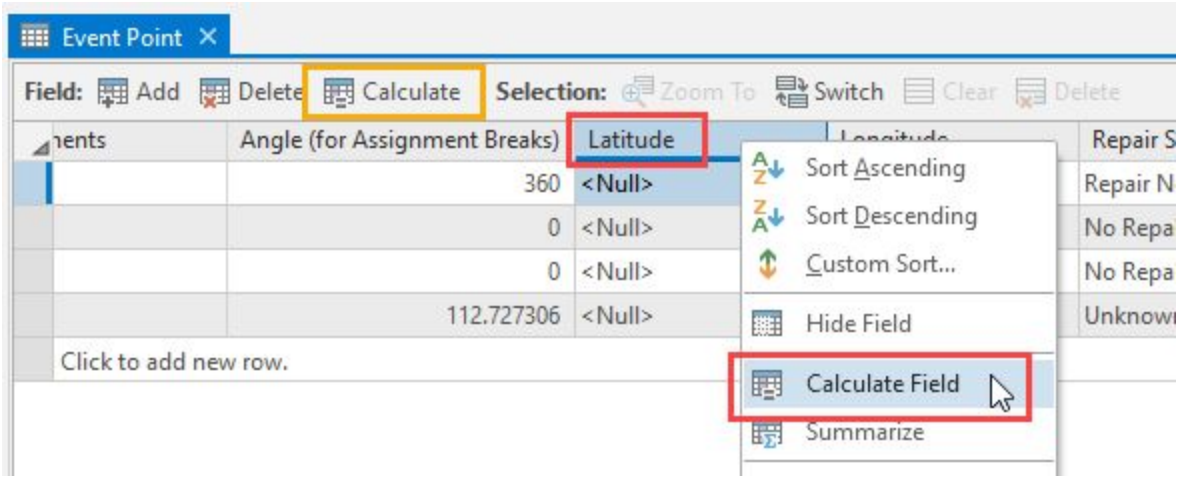
Lat/Long Expression Files (.cal)

These calculations will depend on the spatial reference of the *feature class, not the data frame*.

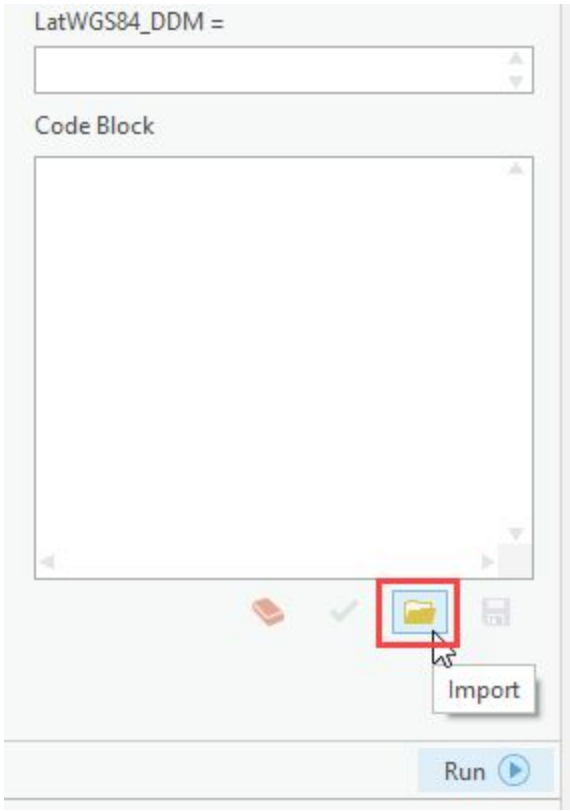
1. Open the layer properties by right-clicking the map in the table of contents and selecting Properties.
2. Check the layer Spatial Reference to be sure it has the **WGS84 Geographic** Coordinate System.



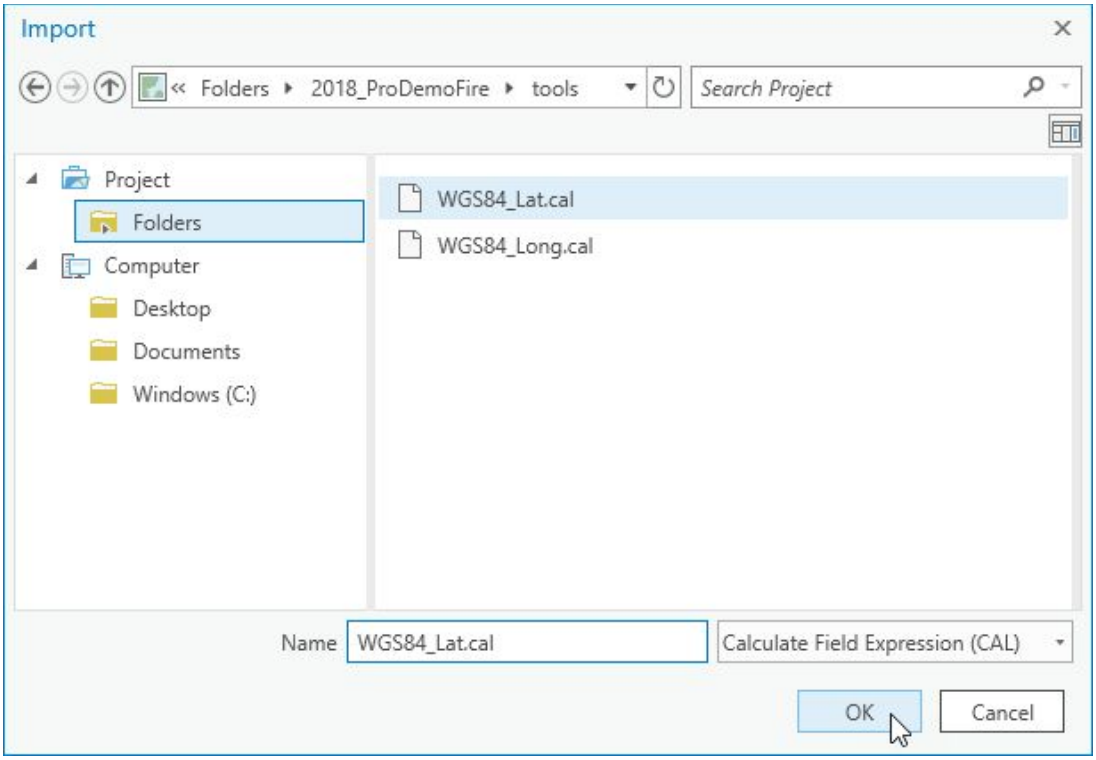
3. Open the Event Point attribute table, right-click the Latitude field, and select Calculate Field. Alternatively, click Calculate and select Latitude as the field name in the Geoprocessing pane.



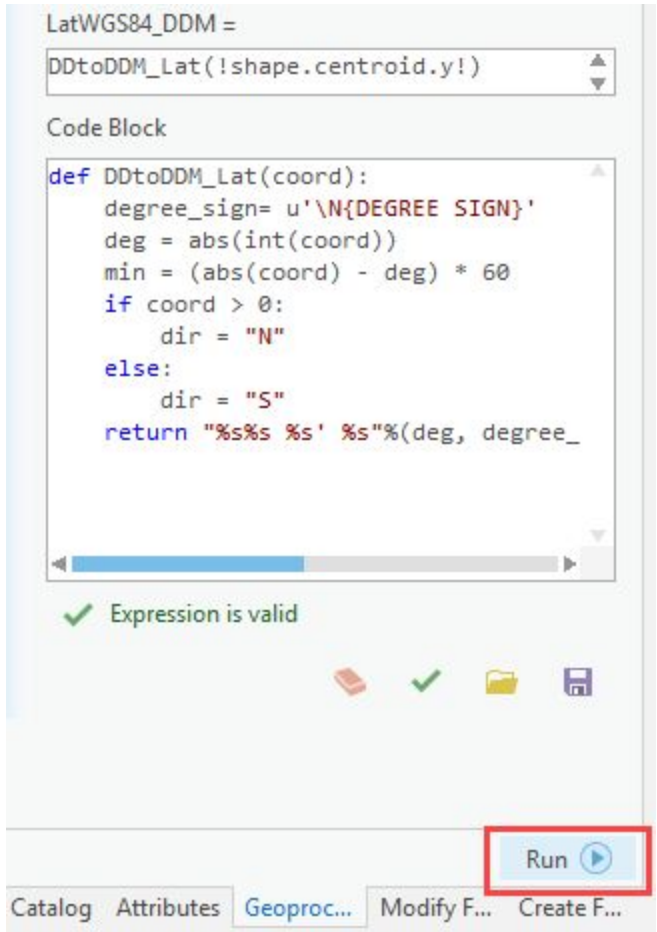
4. In the Calculate Field pane, click the Import button below the Code Block box.



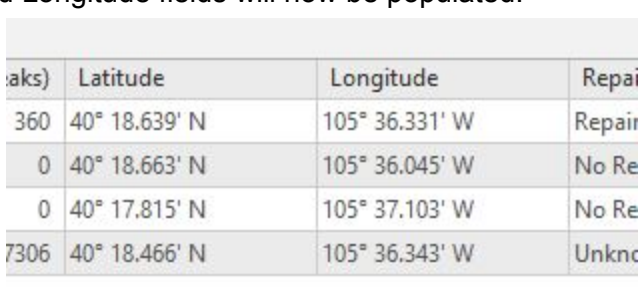
5. Navigate to and select the **WGS84_Lat.cal** expression file.



6. This will populate the expression and code block with the necessary script to calculate Latitude in WGS84 DDM. Click Run.



7. Repeat the same steps for the Longitude field but select the **WGS84_Long.cal** expression file instead.
8. The Latitude and Longitude fields will now be populated.

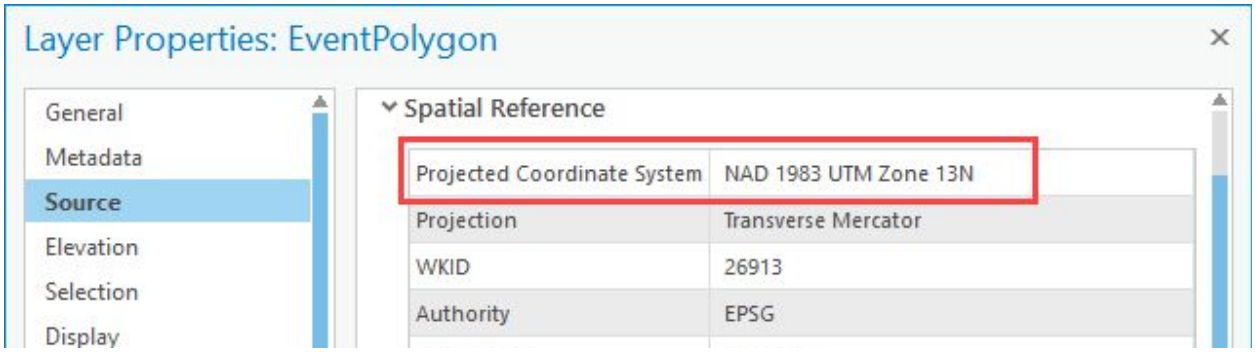


aks)	Latitude	Longitude	Repai
360	40° 18.639' N	105° 36.331' W	Repair
0	40° 18.663' N	105° 36.045' W	No Re
0	40° 17.815' N	105° 37.103' W	No Re
7306	40° 18.466' N	105° 36.343' W	Unknc

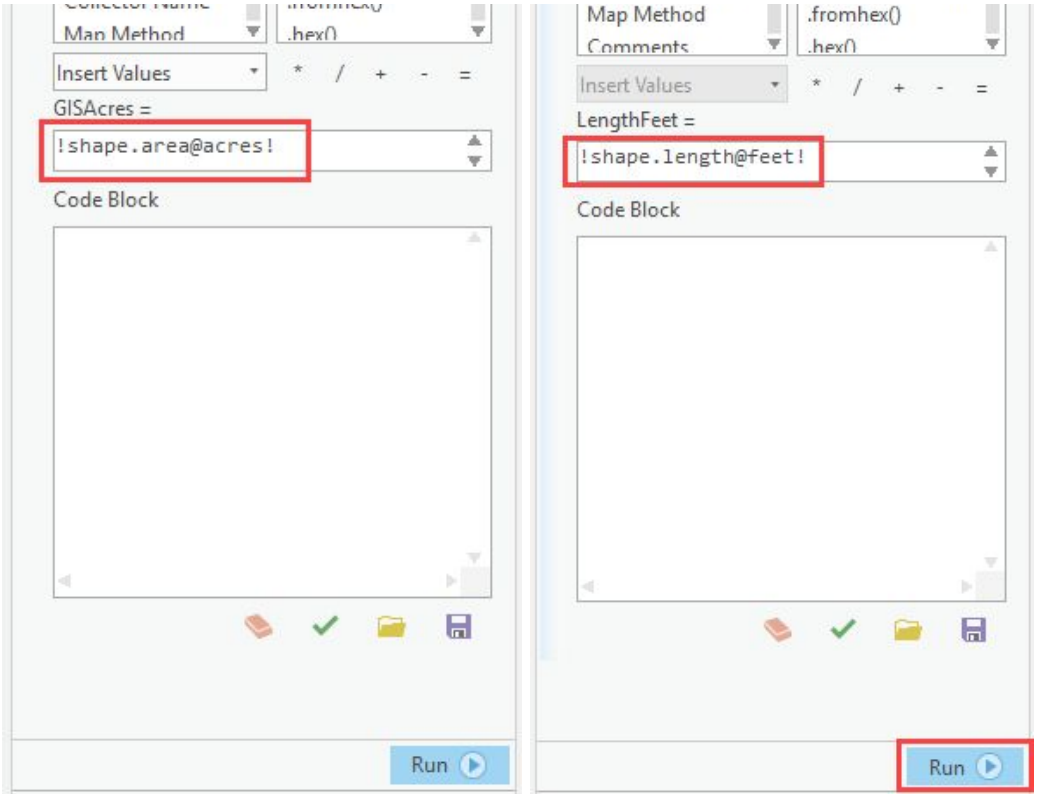
Calculating Geometry on Projected Feature Classes

If for some reason the Event Features must be converted to a projected coordinate system, feature geometries can still be calculated, it is just more complicated for Latitude and Longitude. It is strongly recommended that the following only be used if you have a strong understanding of projections are comfortable with Python.

When a feature class Spatial Reference specifies a **Projected** Coordinate System:



- To calculate Acres, type **!shape.area@acres!** into the expression field and click Run.
- To calculate LengthFeet, type **!shape.length@feet!** into the expression field and click Run.



- To calculate Latitude and Longitude in WGS84 Degrees Decimal Minutes is where things get a little tricky. Coordinates calculated this way are potentially subject to shifting because of reprojection and transformations are selected automatically.
- To calculate **Latitude**, use the following code or import the NotWGS84 expression files.

Expression

calc_notWGS84_long(!SHAPE!)

Code Block

```
def DDtoDDM_Long(coord):
    degree_sign= u'\N{DEGREE SIGN}'
    deg = abs(int(coord))
    min = (abs(coord) - deg) * 60
    if coord > 0:
        dir = "E"
    else:
        dir = "W"
    return "%s%s %s' %s"%(deg, degree_sign, "{:06.3f}".format(min), dir)

def calc_notWGS84_long(geom):
    inSr = geom.SpatialReference
    outSr = arcpy.SpatialReference(4326)

    pt = arcpy.Point(geom.centroid.x,geom.centroid.y)
    ptObj = arcpy.PointGeometry(pt,inSr)

    WGSgeom = ptObj.projectAs(outSr)

    x = WGSgeom.centroid.X

    return DDtoDDM_Long(x)
```

- To calculate **Longitude**, use the following code

Expression

calc_notWGS84_lat(!SHAPE!)

Code Block

```
def DDtoDDM_Lat(coord):
    degree_sign= u'\N{DEGREE SIGN}'
    deg = abs(int(coord))
    min = (abs(coord) - deg) * 60
    if coord > 0:
        dir = "N"
    else:
        dir = "S"
    return "%s%s %s' %s"%(deg, degree_sign, "{:06.3f}".format(min), dir)

def calc_notWGS84_lat(geom):
    inSr = geom.SpatialReference
    outSr = arcpy.SpatialReference(4326)

    pt = arcpy.Point(geom.centroid.x,geom.centroid.y)
    ptObj = arcpy.PointGeometry(pt,inSr)

    WGSgeom = ptObj.projectAs(outSr)

    y = WGSgeom.centroid.Y

    return DDtoDDM_Lat(y)
```