Calculate Field

ArcMap 10.3
|
Other versions

- 10.8
- 10.7
- 10.6
- 10.5
- 10.4
- 10.3

Summary

Calculates the values of a field for a feature class, feature layer, or raster.

View examples of using Calculate Field

Usage

- Python expressions can be created using properties from the Geometry object including type, extent, centroid, firstPoint, lastPoint, area, length, isMultipart, and partCount (for example, !shape.area!).
- Python expressions can use the geometry area and length properties with an areal or linear unit to convert the value to a different unit of measure (for example, !shape.length@kilometers!). If the data is stored in a geographic coordinate system and a linear unit is supplied (for example, miles), the length will be calculated using a geodesic algorithm. Using areal units on geographic data will yield questionable results as decimal degrees are not consistent across the globe.
    - Areal unit of measure keywords:
        - ACRES | ARES | HECTARES | SQUARECENTIMETERS | SQUAREDECIMETERS | SQUAREINCHES | SQUAREFEET | SQUAREKILOMETERS | SQUAREMETERS | SQUAREMILES | SQUAREMILLIMETERS | SQUAREYARDS | SQUAREMAPUNITS | UNKNOWN
    - Linear unit of measure keywords:
        - CENTIMETERS | DECIMALDEGREES | DECIMETERS | FEET | INCHES | KILOMETERS | METERS | MILES | MILLIMETERS | NAUTICALMILES | POINTS | UNKNOWN | YARDS
- Python expressions can be used to calculate the geodesic area or length of a feature by using geodesicArea or geodesicLength properties combined with areal or linear units of measure (for example, !shape.geodesicArea@hectares! or !shape.geodesicLength@miles!).
- When used with a selected set of features, such as those created from a query in Make Feature Layer or Select Layer By Attribute, this tool will only update the selected records.
- The calculation can only be applied to one field per operation.
- Be aware that this tool honors the Output Extent environment setting. Only those features within the extent will have their field values calculated. The environment setting has no effect on non-spatial data such as tables.
- Existing field values will be overwritten. A copy of the input table should be made if you want to preserve the original values.
- For Python calculations, field names must be enclosed in exclamation points (!fieldname!).

  For VB calculations, field names must be enclosed in square brackets ([fieldname]).
- To calculate strings to text or character fields, in the dialog box the string must use double quotation marks ("string"), or in scripting, the string using double quotation marks must also be encapsulated in single quotation marks ('"string"').
- This tool can also be used to update character items. Expressions using a character string should be wrapped using single quotation marks, for example, [CHARITEM] = 'NEW STRING'. However, if the character string has embedded single quotation marks, wrap the string using double quotation marks, for example, [CHARITEM] = "TYPE'A'".
- To calculate a field to be a numeric value, enter the numeric value in the Expression parameter; no quotation marks around the value are required.
- The arcgis.rand() function is supported by this tool when a Python expression is specified. The arcgis.rand() function has been created for ArcGIS tools and should not be confused with Python's random module. The syntax for the available distributions for the arcgis.rand() function can be found at The distribution syntax for random values.
- The expression and code block are connected. The code block must relate back to the expression; the result of the code block should be passed into the expression.
- The Code Block parameter allows you to create complex expressions. You can enter the code block directly on the dialog box or as a continuous string in scripting.
- The Python math module and formatting are available for use in the Code Block parameter. You can import additional modules. The math module provides number-theoretic and representation functions, power and logarithmic functions, trigonometric functions, angular conversion functions, hyperbolic functions, and mathematical constants. To learn more about the math module, see the Python help.
- Saved VB .cal files from previous versions of ArcGIS may work or require minimal modifications. If you have VBA code from past releases that use ArcObjects, you will

need to modify your calculations to work.

- When calculating joined data, you cannot calculate the joined columns directly. However, you can directly calculate the columns of the origin table. To calculate the joined data, you must first add the joined tables or layers to the map. You can then perform calculations on this data separately. These changes will be reflected in the joined columns.
- Field calculations with a VB Expression type are not supported on 64-bit products, including ArcGIS for Desktop—Background Geoprocessing (64-bit) and ArcGIS for Server. To successfully use Calculate Field in these products, expressions should be converted to Python, or in the case of Background Geoprocessing (64-bit), background processing can alternatively be disabled.
- Python expressions that attempt to concatenate string fields that include a null, or divide by zero, will return a null for that field value.
- Calculate Field examples

Syntax

```
CalculateField_management (in_table, field, expression, {expression_type}, {code_block})
```

Parameter	Explanation	Data Type
in_table	The table containing the field that will be updated with the new calculation.	Raster Catalog Layer; Mosaic Layer; Raster Layer; Table View
field	The field that will be updated with the new calculation.	Field
expression	The simple calculation expression used to create a value that will populate the selected rows.	SQL Expression
expression_type (Optional)	Specify the type of expression that will be used. • VB —The expression will be written in a standard VB format. This is the default. • PYTHON —The expression will be written in a standard Python format. Use of geoprocessor methods and properties is the same as creating a 9.2 version geoprocessor. • PYTHON_9.3 —The expression will be written in a standard Python format. Use of geoprocessor methods and properties is the same as creating a 9.3 version geoprocessor. **Caution:** Field calculations with a VB Expression type are not supported on 64-bit products, including ArcGIS Pro, ArcGIS for Desktop—Background Geoprocessing (64-bit) and ArcGIS for Server. To successfully use Calculate Field in these products, expressions should be converted to Python, or in the case of Background Geoprocessing (64-bit), background processing can alternatively be disabled.	String
code_block (Optional)	Allows for a block of code to be entered for complex expressions.	String

Code sample

CalculateField example (Python window)

The following Python window script demonstrates how to use the CalculateField function in immediate mode.

```
import arcpy
arcpy.env.workspace = "C:/data"
arcpy.AddField_management("vegtable.dbf", "VEG_TYP2", "TEXT", "", "", "20")
arcpy.CalculateField_management("vegtable.dbf", "VEG_TYP2",
                                '!VEG_TYPE!.split(" ")[-1]', "PYTHON_9.3")
```

CalculateField example: Calculate centroids

Use CalculateField to assign centroid values to new fields.

```
# Name: CalculateField_Centroids.py
# Description: Use CalculateField to assign centroid values to new fields

# Import system modules
import arcpy

try:
    # Set environment settings
    arcpy.env.workspace = "C:/data/airport.gdb"

    # Set local variables
    inFeatures = "parcels"
    fieldName1 = "xCentroid"
    fieldName2 = "yCentroid"
    fieldPrecision = 18
    fieldScale = 11

    # Add fields
    arcpy.AddField_management(inFeatures, fieldName1, "DOUBLE",
                              fieldPrecision, fieldScale)
    arcpy.AddField_management(inFeatures, fieldName2, "DOUBLE",
                              fieldPrecision, fieldScale)

    # Calculate centroid
    arcpy.CalculateField_management(inFeatures, fieldName1,
                                    "!SHAPE.CENTROID.X!",
                                    "PYTHON_9.3")
    arcpy.CalculateField_management(inFeatures, fieldName2,
                                    "!SHAPE.CENTROID.Y!",
                                    "PYTHON_9.3")
except Exception:
    e = sys.exc_info()[1]
    print(e.args[0])
```

CalculateField example: Calculate ranges

Use CalculateField with a code block to calculate values based on ranges.

```
# Name: CalculateField_Ranges.py
# Description: Use CalculateField with a codeblock to calculate values
#  based on ranges

# Import system modules
import arcpy

# Set environment settings
arcpy.env.workspace = "C:/data/airport.gdb"

# Set local variables
inTable = "parcels"
fieldName = "areaclass"
expression = "getClass(float(!SHAPE.area!))"
codeblock = """def getClass(area):
    if area <= 1000:
        return 1
    if area > 1000 and area <= 10000:
        return 2
    else:
        return 3"""

# Execute AddField
arcpy.AddField_management(inTable, fieldName, "SHORT")

# Execute CalculateField
arcpy.CalculateField_management(inTable, fieldName, expression, "PYTHON_9.3",
                                codeblock)
```

CalculateField example: Calculate random values

Use CalculateField to assign random values to a new field.

```
# Name: CalculateField_Random.py
# Description: Use CalculateField to assign random values to a new field

# Import system modules
import arcpy

# Set environment settings
arcpy.env.workspace = "C:/data/airport.gdb"

# Set local variables
inFeatures = "parcels"
fieldName = "RndValue"
expression = "arcgis.rand('Integer 0 10')"

# Execute AddField
arcpy.AddField_management(inFeatures, fieldName, "LONG")

# Execute CalculateField
arcpy.CalculateField_management(inFeatures, fieldName, expression, "PYTHON_9.3")
```

Environments

- Current Workspace
- Extent

Licensing information

- ArcGIS for Desktop Basic: Yes
- ArcGIS for Desktop Standard: Yes

- ArcGIS for Desktop Advanced: Yes

Related topics

- An overview of the Fields toolset