Editing Tables with the Field Calculator

Tracy.Rijken@TeachMeGIS.com

Presentation Overview

- Getting to Know the Field Calculator
- Simple Calculations
- Function Calculations
- Advanced Calculations
Getting to Know the Field Calculator

What is it and why do we love it?

Field Calculator

Use the Field Calculator to modify records in a table
Field Calculator Expression

Calculate values for a column

Build an expression to calculate the value
- Same value
- Combination of values

Choose parser (language)
- VB Script
- Python

Some Things to Know

Modifying Data
- Need write access to data
- Existing field
- In or out of edit session
  - Undo-able inside

Selection
- Only modifies selected records
- Modifies all records if none selected
- No checkbox option in 10.0

Results
- Geoprocessing → Results
  - Error messages
  - Re-run
Simple Calculations

Static Values

Value type must match destination column type
- String
  - Quotes
  - Follow parser rules
- Number
  - No quotes

“Inner Corridor Technologies”
Copy Values

- Source type must match destination column type
- Fields
  - !<fieldname>! in Python
  - [<fieldname>] in VB Script

Combine Values

- Merge multiple columns
- Combine attribute values and static values
- Data types matter
- Syntax depends on parser
  - + for Python
  - & for VB Script

+Township! + "N-" + !Range! + "W"
Combination Tricks

<table>
<thead>
<tr>
<th>EXPRESSION</th>
<th>RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>!PointX! + “, ” + !PointY!</td>
<td>223.43, 5356.5</td>
</tr>
<tr>
<td>!County_Name! + “ County, ” + !State!</td>
<td>Harris County, TX</td>
</tr>
<tr>
<td>!Company! + “\n” + “!Address!”</td>
<td>TeachMeGIS 3000 Wilcrest</td>
</tr>
<tr>
<td>[Company] &amp; vbNewLine &amp; “!Address”</td>
<td>TeachMeGIS 3000 Wilcrest</td>
</tr>
</tbody>
</table>

Function Calculations

Number & String
Number Functions

- **Basic math**
  - + - / *
- **Built-in Functions**
  - round()
  - abs()
  - max()
  - min()

```
round(!PROD!, 1)
```

Number Tricks

<table>
<thead>
<tr>
<th>EXPRESSION</th>
<th>RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>!Elevation_ft! * 0.3048</td>
<td>Feet → Meters</td>
</tr>
<tr>
<td>!Elevation_m! * 3.2808399</td>
<td>Meters → Feet</td>
</tr>
<tr>
<td>!Area_km2! * 247.105381</td>
<td>km² → Acres</td>
</tr>
<tr>
<td>!Area_acres! * 0.00404685642</td>
<td>Acres → km²</td>
</tr>
<tr>
<td>round(!PROD! * !Royalty!, 2)</td>
<td>$</td>
</tr>
<tr>
<td>!Part1! / !Total!</td>
<td>%</td>
</tr>
</tbody>
</table>
String Functions

- Built-in Functions
  - `.upper()`
  - `.lower()`
  - `.title()`

- The field name goes before the `.function()`

!Company!.upper()

String Slicing

- Slice returns a **substring**:
  - 2 inputs (start, end)
  - All inputs are 0-based
  - Negative inputs count backwards from end
  - Non-inclusive

!API![0:7]
Type Conversion

- Convert values to correct data type
  - str()
  - int()
  - float()

float( !GasSales! )

Advanced Calculations

Show Codeblock
Show Codeblock Calculations

Write function in Pre-Logic Script Code section
Call function (with inputs) in Expression section
Returned value is written to the table

Sequential Numbers

TASK:
- Populate a column with sequential numbers

numVal = 0
def SequentialNums(numStart):
global numVal
if (numVal < numStart):
    numVal = numStart
else:
    numVal = numVal + 1
return numVal
Pick Non-Null Value

**TASK:**
- Populate a column with either API or UWI (whichever exists)

```python
def PickNonNullString(val1, val2):
    valOut = val2
    if val1:
        valOut = val1
    return valOut

PickNonNullString(str(API!), str(UWI!))
```

Any questions?

Tracy.Rijken@TeachMeGIS.com