

STREAMLINING DATA COLLECTION

IMPLEMENTING DATABASE TRIGGERS TO ACCELERATE DATA
COLLECTION AND MAPPING

John May, GISP
jmaygisp@gmail.com



TRIGGERS ENHANCE ARCGIS COLLECTOR

- Collector is a great tool
 - Enables users to interact with data
 - Minimizes learning curves
 - Delivers great UI
 - Puts the data manager in control



TRIGGERS ENHANCE ARCGIS COLLECTOR

- Using Collector in the field can present a few challenges
 - Form factor can be challenging when many fields are present.
 - Generally only 6 or 7 fields are viewable at a time.
 - Information to be collected may not be readily discernible.
 - For instance a user may not know what drainage basin they are standing in when they log a new point in a feature class.



EXAMPLE 1 AUTO POPULATE DATA

- Scenario
 - User is collecting information for a feature in the field
 - Some of the feature attributes are present in other feature classes
 - (example: Drainage Basins, Maintenance Yard boundaries, County Boundaries, etc)
 - The person in the field may not know the correct information off the top of their head



EXAMPLE 1 DATABASE TRIGGER

- What is a trigger?
 - A database trigger is procedural code that is automatically executed in response to certain events on a particular table or view in a database.
- In the ESRI world a trigger can also leverage database geometry functions.
 - This means you can perform geoprocessing tasks as part of a SQL statement.

EXAMPLE 1 SOLUTION

- Simple SQL query implemented as a trigger would automatically update the feature class attribute table.

SQL Example

```
UPDATE FeatureA SET FeatureA.DrainageBasin =  
FeatureB.DrainageBasin,  
FROM FeatureA  
INNER JOIN FeatureB  
ON FeatureA.shape.STWithin(FeatureB.shape) = 1  
WHERE DrainageBasin is NULL;  
(where clause is optional)
```

EXAMPLE 1 SOLUTION

- New feature

Location 25.58625691N, 80.38690608W

District Facility ID

State Project Number

FM Number

Facility Name

Facility Type

Side of Road

EXAMPLE 1 SOLUTION

- Saved data reveals information that was automatically generated by a trigger

D6SWF87000-00057
Location 25.58625691N, 80.38690608W
Edited by: May, John- CO, April 22, 2019

Attributes

Statewide Facility ID

D6SWF87000-00057

District Facility ID

TEST

State Project Number

1200

FM Number

Facility Name

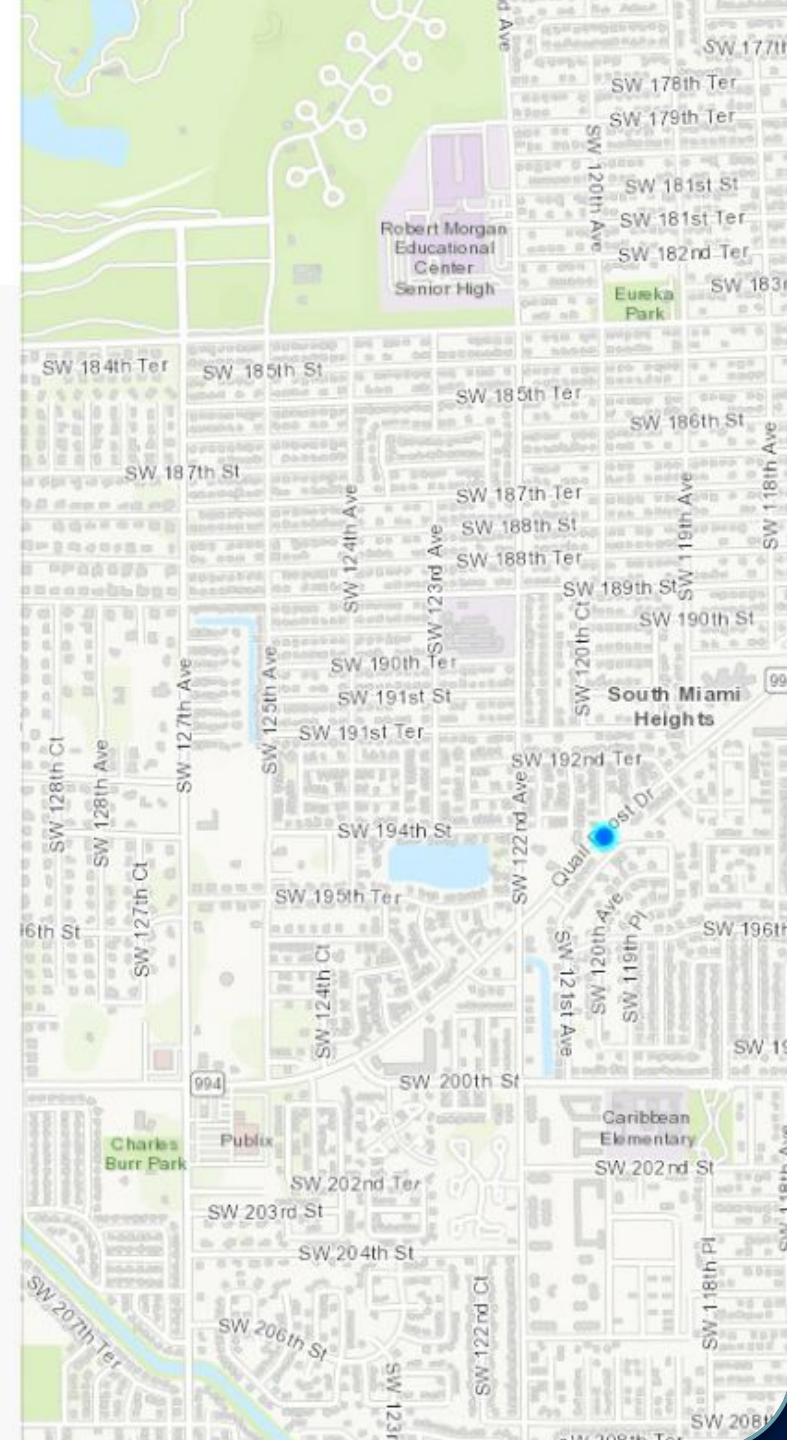
Pond Alpha

Facility Type

Wet Detention

County

MIAMI-DADE





WHY STOP AT 1 ATTRIBUTE

- Update multiple attributes at once.

```
UPDATE FeatureA SET FeatureA.DrainageBasin = FeatureB.DrainageBasin,  
FeatureA.County = FeatureB.County,  
FeatureA.CountyCode = FeatureB.CountyCode,  
FeatureA.PermitAgency = FeatureB.WMDName,  
FeatureA.MaintenanceYard = FeatureB.MaintYard,  
FeatureA.BMAPZone = FeatureB.Bmap,  
FeatureA.District = FeatureB.District  
FROM FeatureA  
INNER JOIN FeatureB  
ON FeatureA.shape.STWithin(FeatureB.shape) = 1  
WHERE FacilityID is NULL;
```



DON'T JUST COMPARE FEATURE CLASSES

- Retrieve a features last inspection date from a related table

```
UPDATE FeatureA set LastInspectionDate = C.InspectionDate
```

```
FROM FeatureA
```

```
INNER JOIN
```

```
(SELECT ParentGUID, FORMAT(InspectionDate,'d') AS InspectionDate from  
FeatureAInspections A
```

```
INNER JOIN
```

```
(select distinct ParentGUID as GUID, max(InspectionDate) as InspDate from  
FeatureAInspections group by ParentGUID) B
```

```
on A.parentguid = B.guid and A.InspectionDate = B.Inspdate) C
```

```
ON FeatureA.GLOBALID = C.PARENTGUID;
```



DON'T JUST COMPARE FEATURE CLASSES

- Automate adding Lat/Longs to your data

UPDATE FeatureA

SET Longitude = Shape.STX ,

Latitude = Shape.STY;



DON'T JUST COMPARE FEATURE CLASSES

- Automate feature naming

Update FeatureA set FacilityId = 'D1SWF'+
FeatureA.countycode + '-' + Format((NEXT Value for
FeatureASequence), '00000') where FacilityID is NULL;



DON'T JUST COMPARE FEATURE CLASSES

- Additional Scenarios
 - Is a point is within a buffered area
 - Is a point is within a distance of a feature
 - Does a line intersect another line or polygon
 - Does a feature fall completely within a polygon
 - Generate centroid values
 - Does a geometry touch another geometry
 - Union objects into a single result



THINGS TO CONSIDER

- Geoprocessing tasks against feature classes
 - Should you use 1 or many reference feature classes
- Think your triggers through
 - A trigger on table A could fire a trigger on table B
- Use triggers to maintain data integrity
 - Manage NULLS
 - Verify defaults
 - Easier to implement than Database changes



RESOURCES

- ESRI Resource URL for STGeometry Functions

<http://desktop.arcgis.com/en/arcmap/latest/manage-data/using-sql-with-gdbs/a-quick-tour-of-sql-functions-used-with-st-geometry.htm>

- Microsoft Resource URL

<https://docs.microsoft.com/en-us/sql/t-sql/spatial-geometry/ogc-methods-on-geometry-instances?view=sql-server-2017>