



Attribute Rules in Arcade

Sangho Kim & Hussein Nasser

2019 ESRI DEVELOPER SUMMIT
Palm Springs, CA

Agenda

- **Attribute Rule Overview**
 - Overview
 - rule types
 - Immediate calculation and constraint rules
- **Validation Service**
 - Overview
 - Workflow
 - Schema changes
- **Dictionary as Return Type**
- **SDK and Sample Code**
- **Summary and What's Coming in the Future?**

*Please silence cell phones
Questions at the end*



Overview of Attribute rules

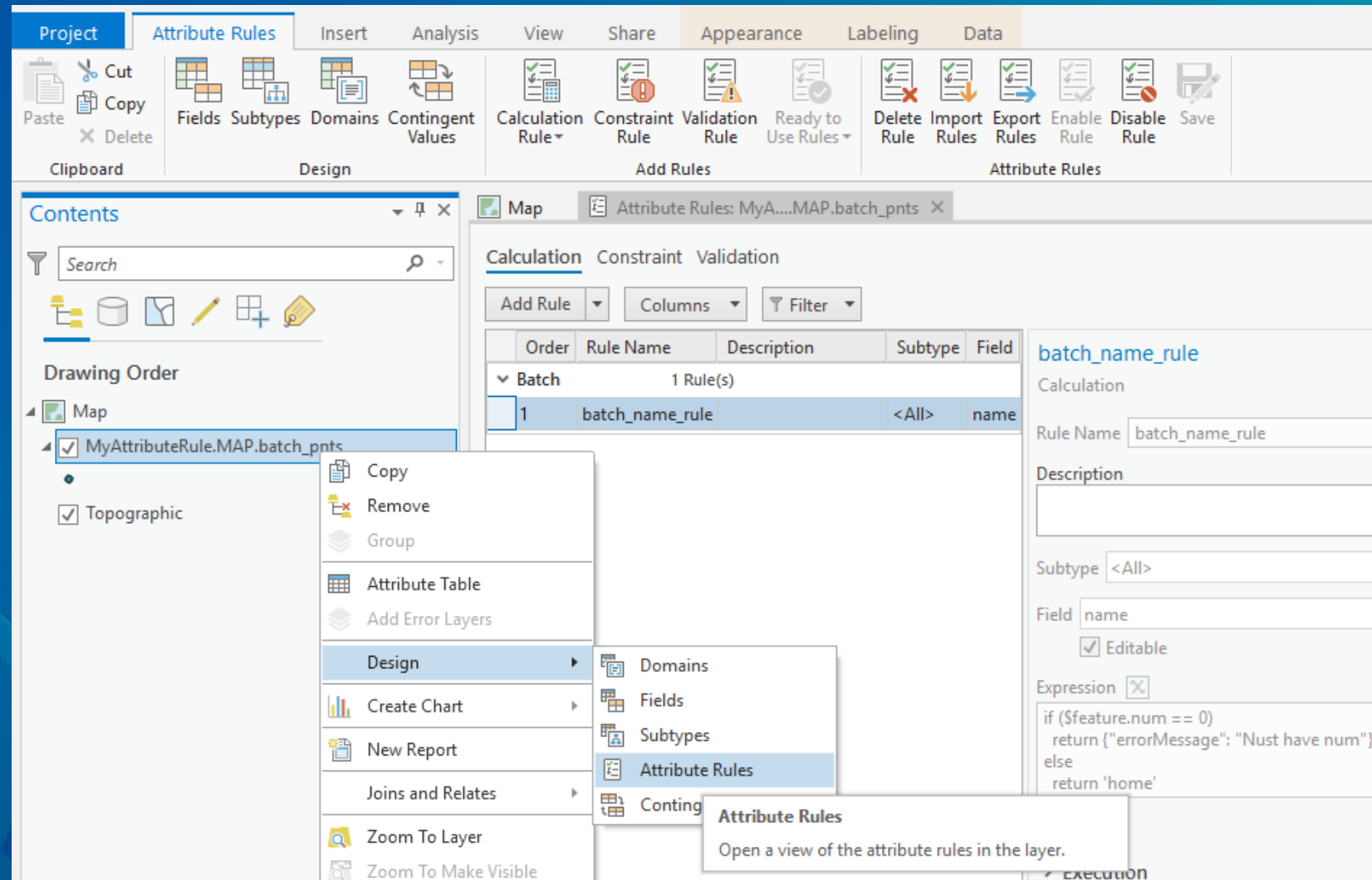
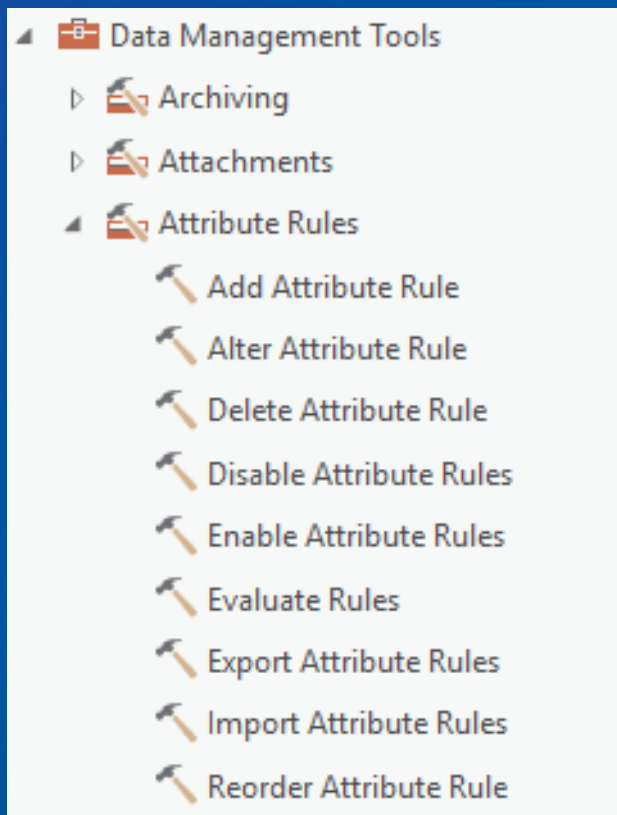
- **Extended Geodatabase capabilities**
 - Existing capabilities: e.g., subtype and domain
- **Evolving technology across ArcGIS platform**
 - Validation is based on service-based architecture
 - Currently available in Enterprise Geodatabase

Overview of Attribute rules

- **User-defined rules to enhance the editing experience and improve data integrity**
 - Automatically populate attributes
 - Constrain invalid edits
 - Perform quality assurance checks on existing features
- **Defined in feature classes or tables in Geodatabase using Arcade scripts**
 - When dataset is deleted, attribute rules are deleted together

Manage Attribute Rules

- Attribute Rules view
- Attribute Rules toolbox



Arcade Expression

- Attribute rules are specified using Arcade language
- One of the properties of rule definition
- Easy to use (i.e., similar to JavaScript) and has a rich library of functions
- Evolves with ArcGIS releases

```
// Similar to JavaScript
var myArray = [10,20,30,40,50,60,70];
var sum = 0;
for(var k in myArray)
{ sum += myArray[k]; }
return sum;

// Returns the number of features that intersect the polygon
var geom2 = Polygon({ ... });
Intersects($feature, geom2);
```

Rule Types

- Evaluated at edit operation

Rule type	Purpose	Return type	Script example
Immediate calculation	Populate attribute	Any value	<code>\$feature.field1 + \$feature.field2</code>
Constraint	Specify permissible value	True / False	<code>\$feature.field1 < \$feature.field2</code>

- Evaluated by validation service

Rule type	Purpose	Return type	Script example
Batch calculation	Populate attribute	Any value	<code>\$feature.field1 + \$feature.field2</code>
Validation	Validate existing data	True / False	<code>\$feature.field1 < \$feature.field2</code>

Immediate calculation rules

- **Used to automatically populate attribute**
- **Execution type is 'Batch' unchecked**
- **Target field that will the rule will be applied to**
 - **Can be user editable or read-only by 'Editable'**
- **Evaluated during edit operations**
 - **Insert, update, or delete**
- **Evaluation order between immediate calculation rules**

My Calculation Rule

Calculation

Rule Name

Description

Subtype

Field

☒ Editable

Expression

> Error

Triggers

Insert <input checked="" type="checkbox"/>	Update <input type="checkbox"/>	Delete <input type="checkbox"/>
---	------------------------------------	------------------------------------

Execution

Server Only <input type="checkbox"/>	Batch <input type="checkbox"/>	Disable <input type="checkbox"/>
---	-----------------------------------	-------------------------------------

Tags

Calculation

Constraint

Validation

Add Rule

Columns

Filter

	Order	Rule Name	Description
▼ Batch	1 Rule(s)		
1		batch_name_rule	
▼ Immediate	2 Rule(s)		
1		My Calculation Rule	Assign 1 + 1 to num attribute.
2		My Calculation Rule 2	Assign 2 + 2 to num attribute.

Evaluation Order Between Calculation Rule Types

Initial values:

field1	field2	field3
1	2	3

Evaluation order: Rule1, Rule2

Rule1: `field1 = field2 + field3`

Rule2: `field3 = field1`

Final results:

field1	field2	field3
5	2	5

vs

Evaluation order: Rule2, Rule1

Rule2: `field3 = field1`

Rule1: `field1 = field2 + field3`

Final results:

field1	field2	field3
3	2	1

Constraint rules

- Specify permissible attribute configurations
 - Ensure that specific conditions are met on a feature
- Arcade expression should return a true or false result
- Evaluated during edit operations
 - Insert, update, or delete
- If violated (i.e., return false),
 - A user-defined error number and message are returned
 - The edit is rolled back

My Constraint Rule

Constraint

Rule Name

Description

Subtype

Expression ☒

▼ Error

Error Number

Error Message

▼ Triggers

Insert ☒ Update ☒ Delete ☐

▼ Execution

Server Only ☐ Disable ☐

Tags



Demo

Immediate Calculation and Constraint Rules



Validation Service

Validation Service

- **Evaluates batch calculation and validation rules**
- **Useful for the following situations:**
 - **Users have existing dataset to evaluate**
 - **E.g., bulk uploading raw data and evaluate them**
 - **Users want to evaluate datasets at a specified time**
 - **E.g., evaluate editors' data at midnight**
 - **Script is too expensive to evaluate at edit operation**
 - **E.g., Arcade expression involves querying other features**

Workflow of Validation Service

- Add a batch calculation or a validation rule to dataset
 - Establish schema (i.e., attributes and tables) required for validation service
- Share the dataset as a web feature layer
 - Validation capability enabled
- Make edits to the web feature layer
- Evaluate rules and check features using Error Inspector
- Fix features with errors and re-evaluate

Capabilities

☐ WCS

☐ KML

☐ Version Management

☒ Validation

Error Inspector: Map1

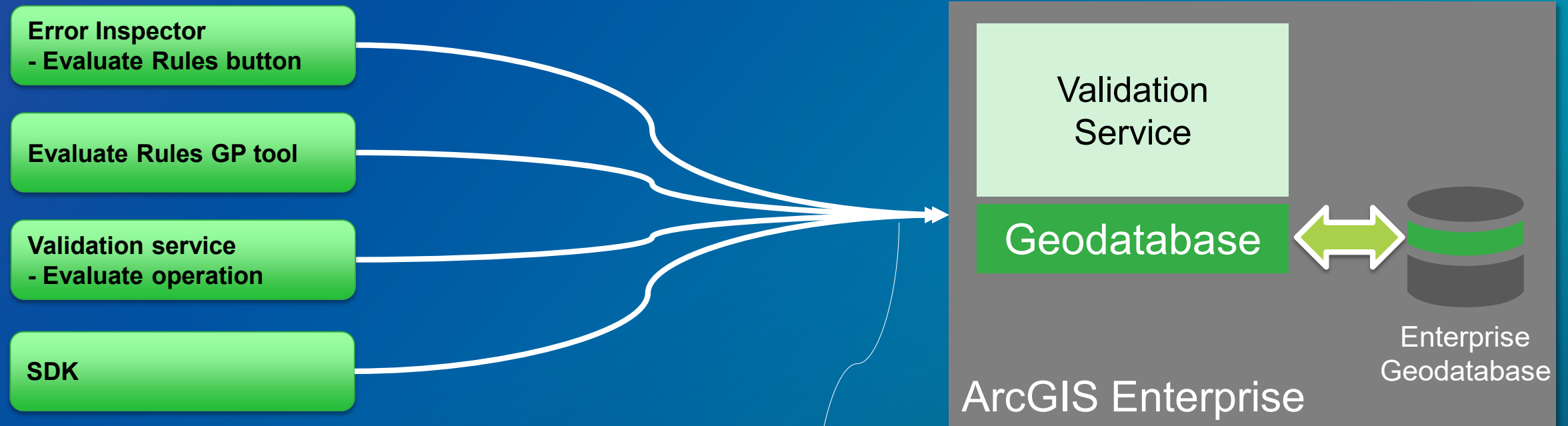
Source: ARTTest_Oracle Error I Evaluate Rules Filter: Map Extent Selection: Zoom To

Shape	Rule Type	Exception	Feature Class ID	Feature ObjectID	Feature GlobalID
	Validation		351	9	{87ACC70E-94FB-41E4-8A6A-BEE0A4E95249}
	Validation		351	7	{6C81E470-82DD-4D90-9BD6-C8963A284DE8}
	Validation		351	5	{E4B365B5-250A-4FE2-A767-4FB06600AEE9}
	Validation		351	8	{A6C1AAA2-5785-42ED-A619-D67FD1AC4A56}
	Validation		351	4	{6F92118F-8F35-42D1-8E3B-F5672D6AC920}
	Validation		351	6	{81540DAA-1ECF-42EA-9668-00974974B000}
	Validation		351	10	{59843753-0B00-48B1-B43A-76CA1C9F7FA6}
	Validation		351	3	{77057CC2-C933-40C0-B877-87B899D2DBB4}

Schema for Validation Service

- “Validation status” attribute is added to a dataset
- Validation status attribute represents the following status of each feature
 - Calculation required
 - Validation required
 - Has error(s)
- Error tables
 - GDB_ValidationPointErrors
 - GDB_ValidationLineErrors
 - GDB_ValidationPolygonErrors
 - GDB_ValidationObjectErrors

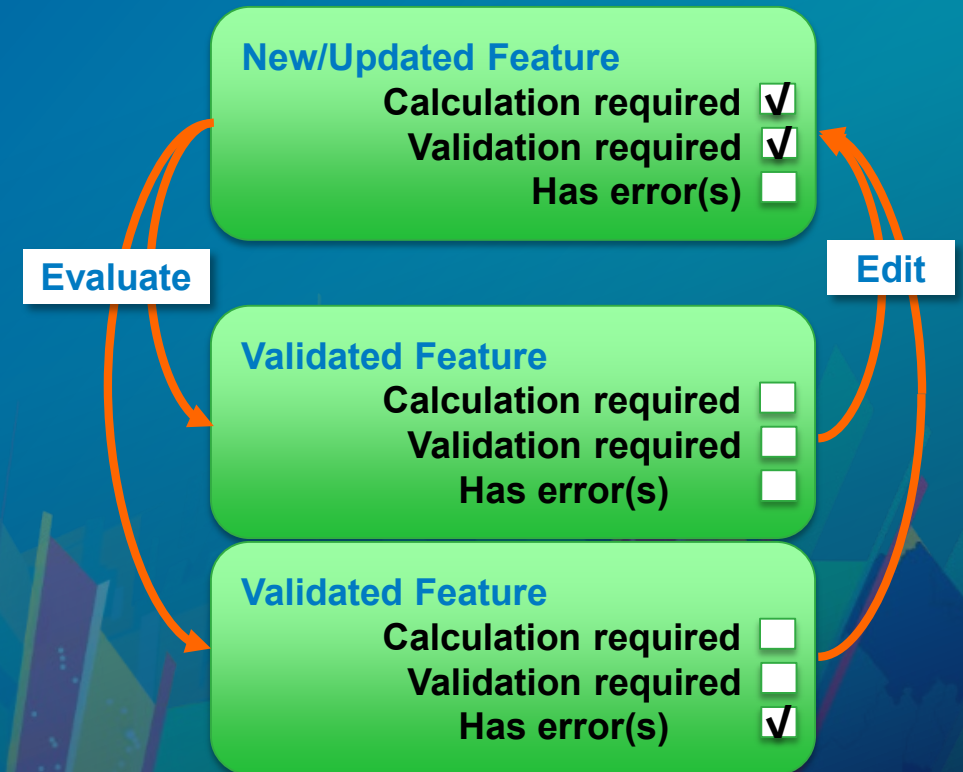
How to Evaluate



<https://myserver.esri.com/server/rest/services/serviceName/ValidationServer/evaluate>

Validation Status Changes

- When a feature is inserted or updated
 - Validation status attribute has calculation/validation required
- When a feature is evaluated by validation service
 - Validation status gets reset
 - Its 'Has error(s)' may be true or false



Lifespan of Error Features

- Analogous to the Geodatabase Topology model
- Error features are created when
 - A validation rule is violated or
 - A batch calculation rule failed while evaluating
 - Expression of a validation or a batch calculation rule returns an error by dictionary (i.e., errorMessage)
- Error features are deleted when
 - Its source feature gets fixed and no longer violates the rule
 - Its source feature gets deleted
 - Its associated rule gets deleted or disabled

Rule1: field1 < 10

field1

1

Error: field1 should be less than 10

Rule1: field1 < 10

field1

11

Error: field1 should be less than 10

Rule1: field1 < 10

~~field1~~

~~1~~

Error: field1 should be less than 10

~~Rule1: field1 < 10~~

field1

1

Error: field1 should be less than 10



Demo

Validation Service



Support Dictionary as Return Type

Support Dictionary as Return Type

- Dictionary return type is needed to support:
 - Mark other features as calculation/validation required
 - User defined error message from Arcade script

- **result**

```
return {'result' , 200}
```

- **calculationRequired, validationRequired**

```
return {'calculationRequired': [{ 'className' : 'MyTable', 'objectIDs' : [1,2,3,4]}] }  
return {'validationRequired': [{ 'className' : 'MyTable', 'globalIDs' : ['E23B...', 'BEC8...']}]}
```

- **errorMessage**

```
return {'errorMessage': 'Invalid phase.'}
```




Demo

Dictionary as Return Type



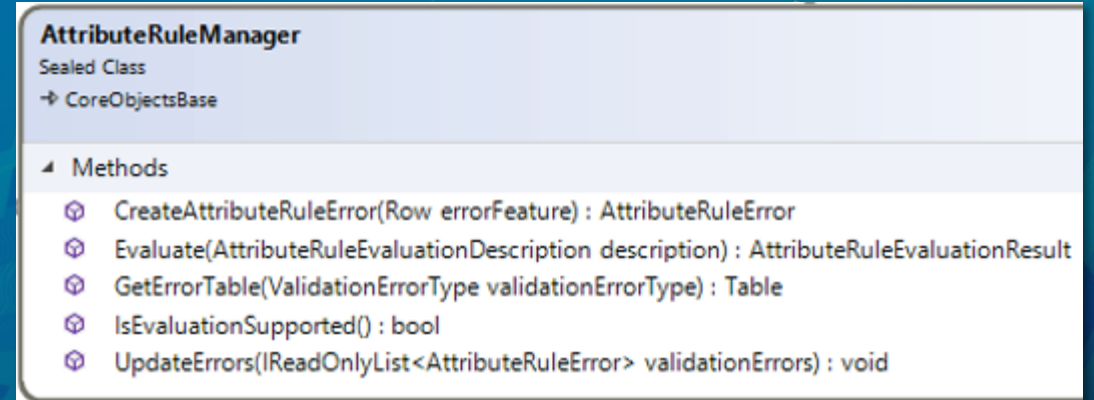
SDK and Sample Code

SDK

- You can evaluate and inspect/update errors by SDK
 - You cannot add/alter/delete attribute rules by SDK (no DDL)
 - No explicit SDK support for constraint and immediate calculation rules
- 

SDK

- **Attribute Rule Manager obtained from Geodatabase**
 - **IsAttributeManagerSupported()**
 - **GetAttributeRuleManager()**
- **Evaluation called from the Attribute Rule Manager**
 - **IsEvaluationSupported()**
 - **Evaluate()**
- **Additional SDK capabilities**
 - **Fetch error features**
 - **Fetch original features that generated errors**
 - **Mark errors as exceptions**



SDK and JavaScript Examples to Evaluate

```
using (Geodatabase geodatabase = new Geodatabase(new ServiceConnectionProperties(new Uri("https://tester.esri.com/...
using (AttributeRuleManager attributeRuleManager = geodatabase.GetAttributeRuleManager())
using (FeatureClass fc = geodatabase.OpenDataset<FeatureClass>(layerID))
{
    AttributeRuleEvaluationResult result = attributeRuleManager.Evaluate(
        AttributeRuleType.Validation, fc.GetExtent(), InvocationTarget.SynchronousService);
    Console.WriteLine("Number of errors => " + result.NumberOfErrors);
}
```

```
const evaluationType = ["calculationRules", "validationRules"]
const evaluationArea = {"xmin":-121.1, "ymin":37.94, "xmax":-121.30, "ymax":37.93, "spatialReference":4326}
const f = "json"
const url = "https://myserver.d.com/server/rest/services/ARTest/ValidationServer/evaluate"
const payload = {
    "method":"POST",
    body: 'token=${token}&evaluationType=${evaluationType}&evaluationArea=${evaluationArea}&f=${f}',
    headers: {'Content-Type': 'application/x-www-form-urlencoded'}
}

fetch(url, payload).then(a=>a.json()).then(console.log).catch(console.error)
```

Arcade Rule Examples

```
return "Tx-" + NextSequenceValue ("assetId") // E.g., "Tx-1"
```

```
// Calculate total voltage of transformers in the substation
var transformers = FeatureSetByName($datastore, "Transformers", ["objectId"], true)
var transformerSubset = Intersects(transformers, Geometry($feature))
var totalVoltage = 0;
for (var t in transformerSubset)
    totalVoltage += t.voltage

return totalVoltage
```

Substation (\$feature)



Summary and What's Coming in the Future?

- Enhance editing experience and data integrity
 - Immediate calculation and constraint rules with Pro 2.1
 - Validation service with Pro 2.3
-
- Support for FileGDB and mobile datasets
 - Support DML (i.e., inserts/updates/deletes)
 - Support copy and paste
 - Handle exceptions

Resources

- **Pro help for Attribute Rules:**

- <https://pro.arcgis.com/en/pro-app/help/data/geodatabases/overview/an-overview-of-attribute-rules.htm>

- **Arcade**

- <https://developers.arcgis.com/arcade/>

- **SDK**

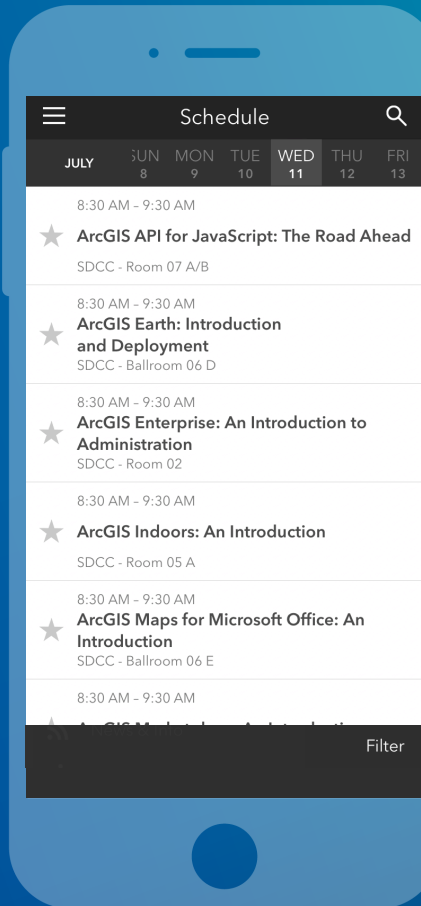
- <https://github.com/esri/arcgis-pro-sdk/wiki/ProConcepts-Geodatabase#attribute-rules>

Please Take Our Survey on the App

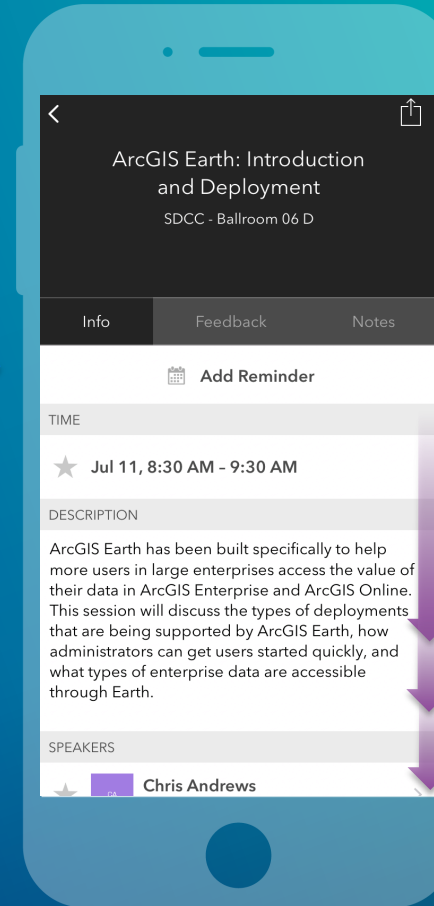
Download the Esri Events app and find your event



Select the session you attended



Scroll down to find the feedback section



Complete answers and select "Submit"

