



AutoCAD Utility Design: Bending the Rules

Dan Leighton
Principal Consultant – DL Consulting

Summary

The new generation of AutoCAD Utility Design introduced rules-driven model-based design to the utility industry. The rules engine within Utility Design provides enormous flexibility with a structured framework to create and edit the design rules. This lab will provide you with an overview of the rules engine, and then focus on several examples of practical rule definition for styling, material ordering, cost estimation, feature identifiers, and annotation. In addition, we will have a discussion of the practical considerations to define and manage rules.

Session Agenda

1	3:00–3:05	Session Introduction
2	3:05–3:20	Rules overview
3	3:20–3:55	Key topics and exercises <ul style="list-style-type: none">• Basics of rule editing• Style rules• Material ordering rules• Expression rules• Annotation rules• Analyzing advanced rules
4	3:55–4:00	Closing Thoughts
5	4:00–4:15	Q&A Session

Learning Objectives

At the end of this class, you will be able to...

- Explain the basics of rule creation and editing
- Describe the different types of AUD rules
- Create and modify styling rules
- Create and modify material ordering rules
- Create and modify expressions and annotation rules

AutoCAD Utility Design Rules Overview

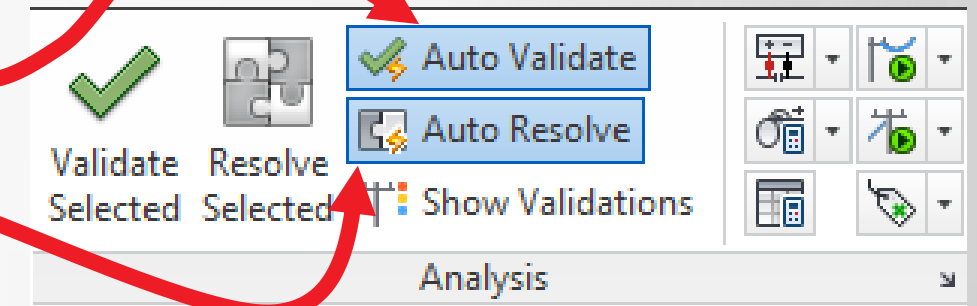
What does it mean to be “rules driven”?

- Utility Design rules control behavior

- Changes can occur behind the scenes
- Rules may prohibit certain design changes

- Two key settings affect rule activity

- Auto Validate updates Validation Report automatically
- Auto Resolve resizes components automatically



What can the rules do?

Control display **style**

- What symbols are used
- How are symbols scaled and colored

Generate the **material list**

- What to include in the order
- Calculate material price

Control component **annotation**

- Annotation display
- Annotation data / content

Design “sanity **checking**”

- Invalid component connections
- Missing connections

Automatically **resize components**










- Right-size conductors/transformers
- Right-size ducts

Automatically **add components**

- Add poles on long OH runs
- Add pole heads and guys

What rules do depend on your configuration

Nine rule categories

	Expressions	Define integer, real, logical, and text values
	Feature Identifier	Generate values for feature identifiers
	Validation	Check design & display issues in Validation Results
	Material	Determine what appears in the Material Editor
	Material Costing	Calculate cost of materials in the Material Editor
	Annotation	Control selection of callouts
	Sizing	Filters components so only valid models are considered
	Style	Defines how components are styled
	Analysis	Defines analysis and validation equations

Example: Expression Rules

+

Expressions

Define integer, real, logical, and text values

+

Feature Identifier

Generate values for feature identifiers

+

Validation

Check validation results

+

Material

Material Editor

+

Material Costing

Calculate material costs

+

Annotation

Control annotation

+

Sizing

Filter components

+

Style

Defines how components are styled

+

Analysis

Defines analysis and validation

-

Expressions

-

Boolean

f_x

Is Existing

f_x

Is New

f_x

Is Removal

f_x

Is Retired

([Status Internal] equals existing)

Example: Feature Identifier

+	Expressions	Define integer, real, logical, and text values
+	Feature Identifier	Generate values for feature identifiers
+	Validation	Check design & display issues in Validation Results
+	Material	Define material
+	Material Costing	Define material costing
+	Annotation	Control annotation
+	Sizing	Filter
+	Style	Define style
+	Analysis	Define analysis










All features (2)

Pad UM1-5C 3650
Pole / WP356 / FID: 3610

Original Feature Identifier display

Modified Feature Identifier for poles

Example: Validation Rules

	Expressions	Define integer, real, logical, and text values
	Feature Identifier	Generate values for feature identifiers
	Validation	Check design & display issues in Validation Results
	Material	Determine what appears in the Material Editor
	Material Costing	Calculate cost of materials in the Material Editor
	Annotation	
	Sizing	
	Style	
	Analysis	

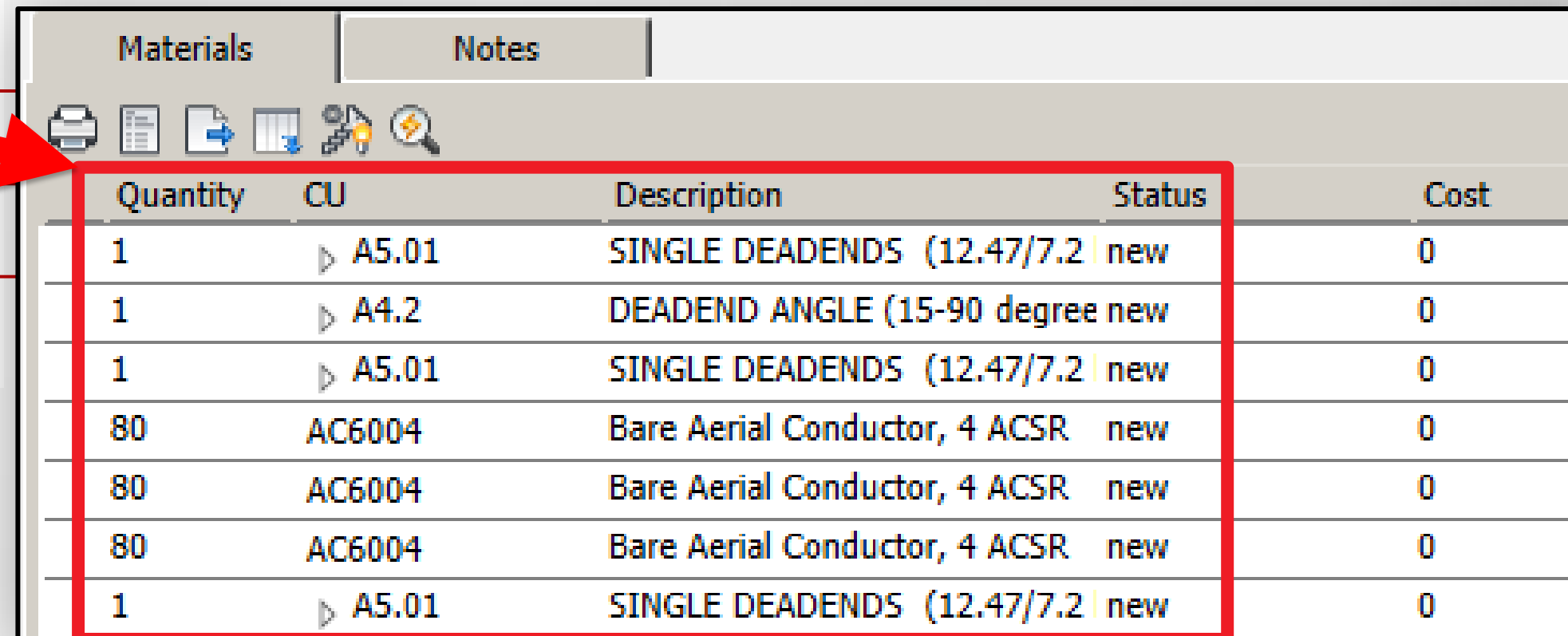
Rule Validation Message

Service Point Underground 5874

Method to determine demand is not specified.

Example: Material Rules

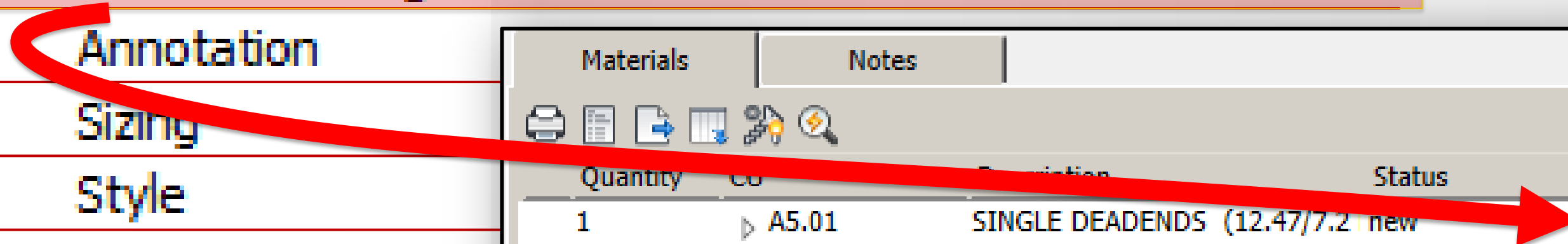
+	Expressions	Define integer, real, logical, and text values
+	Feature Identifier	Generate values for feature identifiers
+	Validation	Check design & display issues in Validation Results
+	Material	Determine what appears in the Material Editor
+	Material Costing	Calculate cost of materials in the Material Editor
+	Annotation	
+	Sizing	
+	Style	
+	Analysis	



Materials		Notes		
Quantity	CU	Description	Status	Cost
1	▶ A5.01	SINGLE DEADENDS (12.47/7.2	new	0
1	▶ A4.2	DEADEND ANGLE (15-90 degree	new	0
1	▶ A5.01	SINGLE DEADENDS (12.47/7.2	new	0
80	AC6004	Bare Aerial Conductor, 4 ACSR	new	0
80	AC6004	Bare Aerial Conductor, 4 ACSR	new	0
80	AC6004	Bare Aerial Conductor, 4 ACSR	new	0
1	▶ A5.01	SINGLE DEADENDS (12.47/7.2	new	0

Example: Material Costing Rules










+	Expressions	Define integer, real, logical, and text values
+	Feature Identifier	Generate values for feature identifiers
+	Validation	Check design & display issues in Validation Results
+	Material	Determine what appears in the Material Editor
+	Material Costing	Calculate cost of materials in the Material Editor

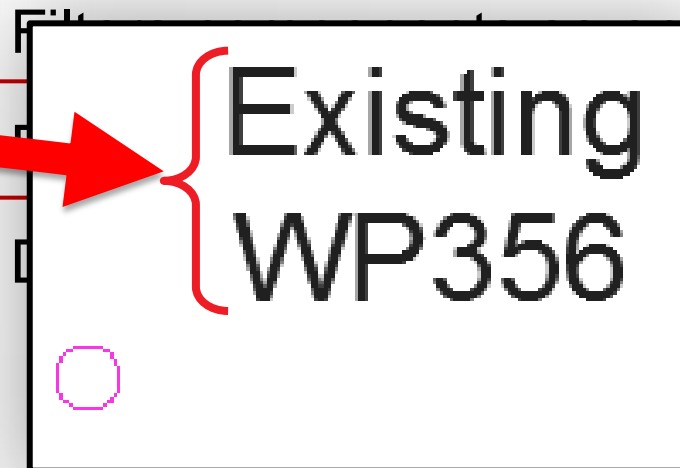


Materials		Notes	
Quantity	CU	Description	Status
1	▶ A5.01	SINGLE DEADENDS (12.47/7.2	new
1	▶ A4.2	DEADEND ANGLE (15-90 degree	new
1	▶ A5.01	SINGLE DEADENDS (12.47/7.2	new
80	AC6004	Bare Aerial Conductor, 4 ACSR	new
80	AC6004	Bare Aerial Conductor, 4 ACSR	new
80	AC6004	Bare Aerial Conductor, 4 ACSR	new
1	▶ A5.01	SINGLE DEADENDS (12.47/7.2	new










Cost
0
0
0
0
0
0
0

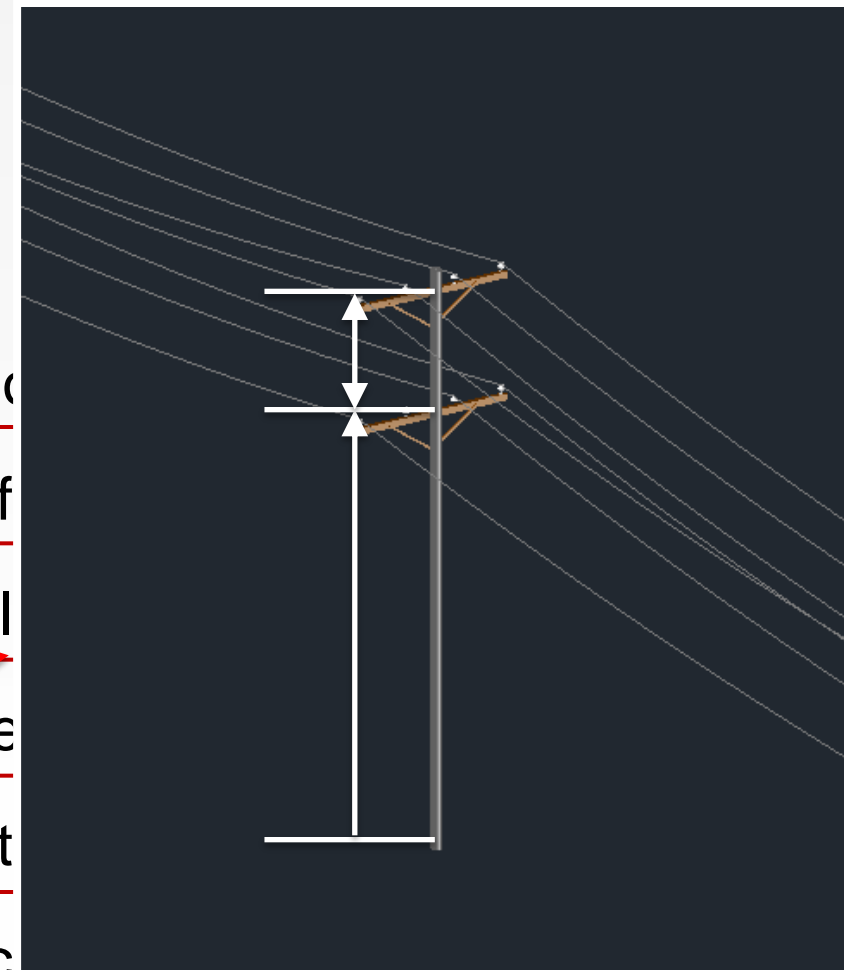
Example: Annotation Rules

	Expressions	Define integer, real, logical, and text values
	Feature Identifier	Generate values for feature identifiers
	Validation	Check design & display issues in Validation Results
	Material	Determine what appears in the Material Editor
	Material Costing	Calculate cost of materials in the Material Editor
	Annotation	Control selection of callouts
	Sizing	Only valid models are considered
	Style	are styled
	Analysis	ation equations



Example: Sizing Rules

	Expressions	Define integer, real, lo
	Feature Identifier	Generate values for f
	Validation	Check design & displ
	Material	Determine what appe
	Material Costing	Calculate cost of mat
	Annotation	Control selection of c
	Sizing	Filters components so only valid models are considered
	Style	Defines how components are styled
	Analysis	Defines analysis and validation equations



Example: Style Rules

+	Expressions	Define integer, real, logical, and text values
+	Feature Identifier	Generate values for feature identifiers
+	Validation	Check design & display issues in Validation
+	Material	Determine what appears in the Material Browser
+	Material Costing	Calculate cost of materials in the Material Browser
+	Annotation	Control selection of callouts
+	Sizing	Filters components so only valid models are shown
+	Style	Defines how components are styled
+	Analysis	Defines analysis and validation equations

If ([Pad Type] equals concrete)		
Then		
Select case where [Status Internal] equals		
new	Choose style Default	
existing	Choose style Existing	
Otherwise	Choose style Removal	
Else		
Select case where [Status Internal] equals		
new	Choose style New-NC	
existing	Choose style Existing-NC	
Otherwise	Choose style Removal-NC	

Example: Analysis Rules

<input type="checkbox"/>	Expressions	Define integer, real, logical, and text values
<input type="checkbox"/>	Feature Identifier	Generate values for feature identifiers
<input type="checkbox"/>	Validation	Check design & display issues in Validation
<input type="checkbox"/>	Material	Determine what appears in the Material Editor
<input type="checkbox"/>	Material Costing	Calculate cost of materials in the Material Editor
<input type="checkbox"/>	Annotation	Control selection of callouts
<input type="checkbox"/>	Sizing	Filters components so only valid models are shown
<input type="checkbox"/>	Style	Defines how components are styled
<input type="checkbox"/>	Analysis	Defines analysis and validation equations

☐ **Analysis**

- ☐ Clearance Check
- ☐ Flicker
- ☐ Guying
- ☐ Load
- ☐ Pole Attachments
- ☐ Pulling Tension
- ☐ Sag
- ☐ Voltage Drop
 - ☐ Equation for Meshed Voltage Drop
 - ☐ Equation for Radial Voltage Drop
- ☐ Validation
 - ☐ Conductor Voltage Drop
 - ☐ Light Voltage Drop
 - ☐ Service Point Voltage Drop
 - ☐ Transformer Voltage Drop

Two types of rules

Expressions (=Functions)

- Expressions that define Boolean, integer, real, or text values
- `IsExisting → ([Status Internal] equals existing)`

Sequential/Action Rules (=Methods)

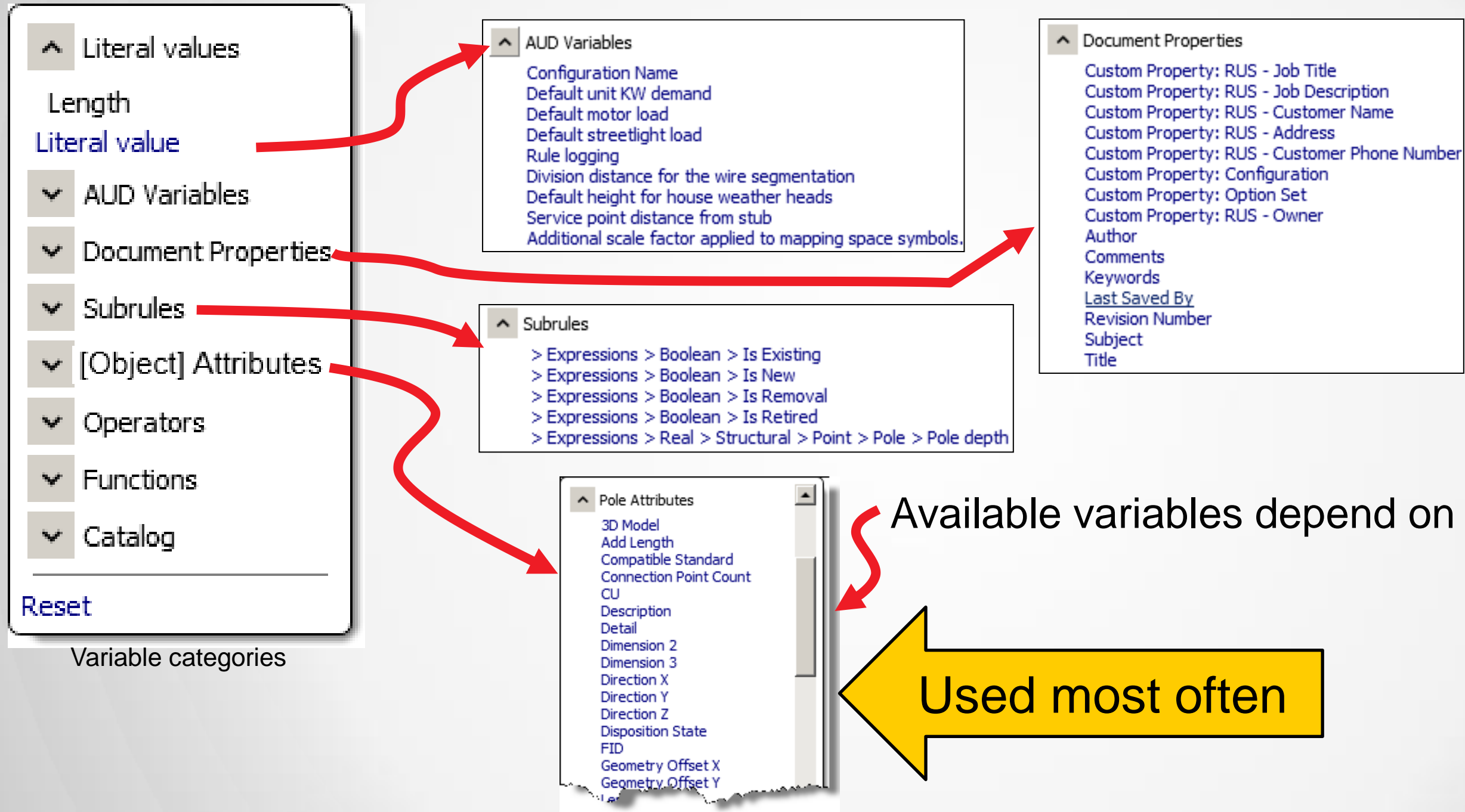
- Sequential/action rules execute a set of instructions
- `If ([Status] is not equal to existing)`
Then
`Add material Model Name: [Model Name] Quantity: 1 Status: [Status]`

Basics of Rule Editing

Exercise 1: Basic steps to create/edit a rule

1. Start AutoCAD Utility Design
2. Open the AUD_Rules.dwg file
3. Select the Exercise 1 view – You will see two Vaults
4. Start the Rule Editor
5. Locate and review the style rule for Vaults
6. Change the rule to set a different Style based on Status Internal
7. Note the change in the drawing

Variables within the rules engine (part 1)



Variables within the rules engine (part 2)

The diagram illustrates the structure of variables within the rules engine. It features a 'Variable categories' panel on the left with expandable sections: Literal values, AUD Variables, Document Properties, Subrules, [Object] Attributes, Operators, Functions, and Catalog. A 'Reset' button is located at the bottom of this panel. To the right, three panels display the contents of these categories: 'Operators' (containing arithmetic and comparison operators), 'Functions' (containing mathematical and logical functions), and 'Material Catalog' (containing a list of material categories and items). Red arrows indicate the mapping from the 'Variable categories' panel to the respective content panels: 'Operators' to the Operators panel, 'Functions' to the Functions panel, and 'Catalog' to the Material Catalog panel.

Variable categories

- Literal values
 - Length
 - Literal value
- AUD Variables
- Document Properties
- Subrules
- [Object] Attributes
- Operators
- Functions
- Catalog
- Reset

Operators

- +
-
- *
- /
- equals
- equals within tolerance
- not equal
- not equal within tolerance
- is more than
- is less than
- is at least
- is at most
- and
- or
- not
- exclusive or

Functions

- abs
- acos
- asin
- atan
- atan2
- attribute is not set
- attribute is set
- average
- ceil
- choose value where
- conditional number
- cos
- cosh
- e
- equality
- exp
- feature class caption
- first
- floor

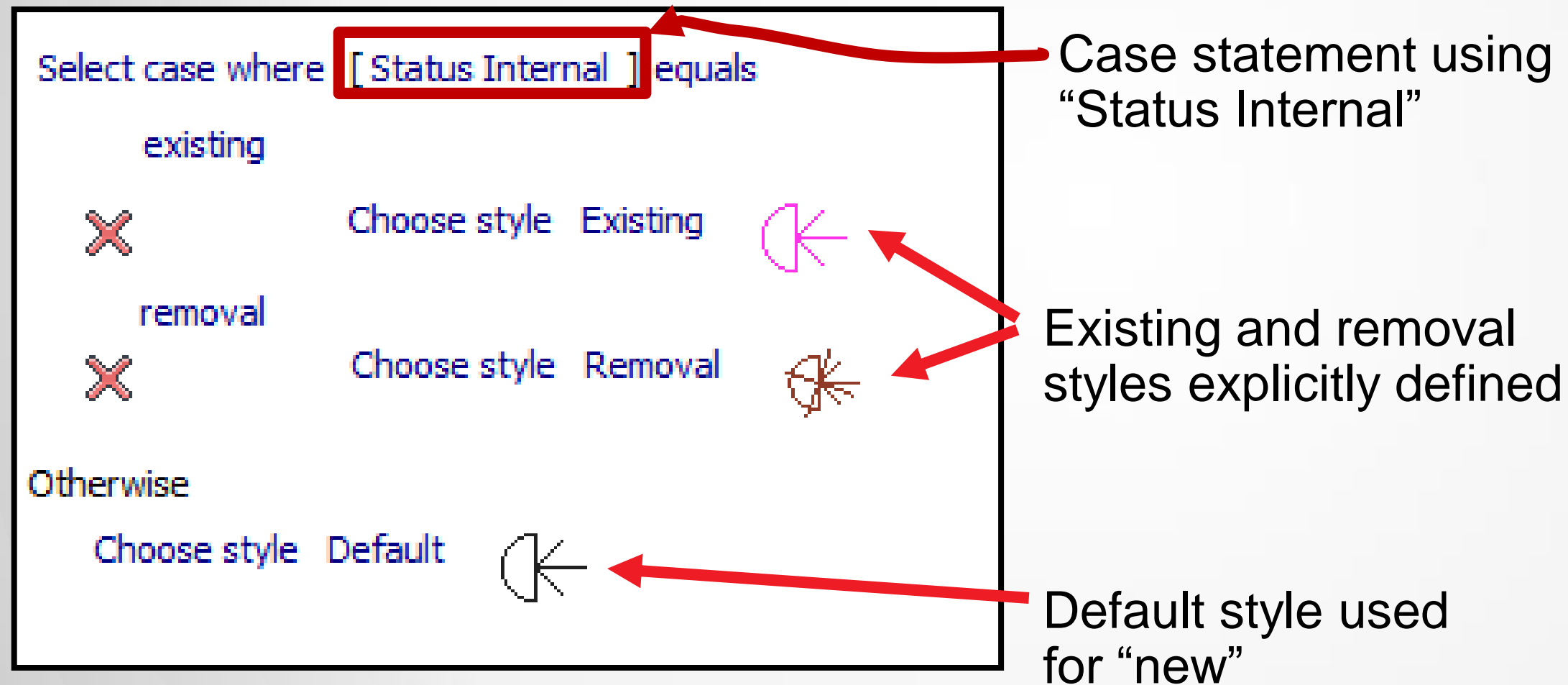
Material Catalog

Category	Items
Anchor	CU Description
Anode	F2.6 This is the default, all users Material catalog.
Auto Transformer	F3.6A PLATE TYPE ANCHORS
Brace	F5.1 ROCK ANCHORS
Cable	F6.8 SWAMP ANCHORS (POWER INSTALLED)
Capacitor	F1.6 EXPANDING TYPE ANCHORS
CATV Cable	
Communications Enclosure	
Communications Framing	
Communications Messenger	

Style Rules

The basic styling rule

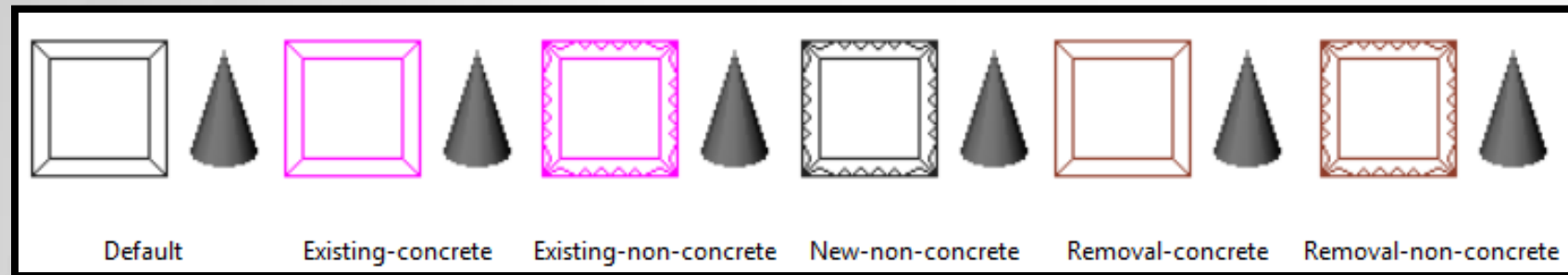
Determine which style (as defined within the Industry Model configuration) is used given various component values



Important stuff!

- ✓ Define styles before creating styling rules!
- ✓ Create a style for every display variation

Advanced style rule based on additional attributes



```
If ([[ Status Internal ] equals new ) and ([ Pad Type ] equals concrete ))  
Then  
  Choose style Default  
  
Else  
  Choose style by name "[ Status Internal ] - [ Pad Type ] <Add Text...> "
```

Explicit handling for
New Concrete Pads

Calculated style
name for all other
combinations

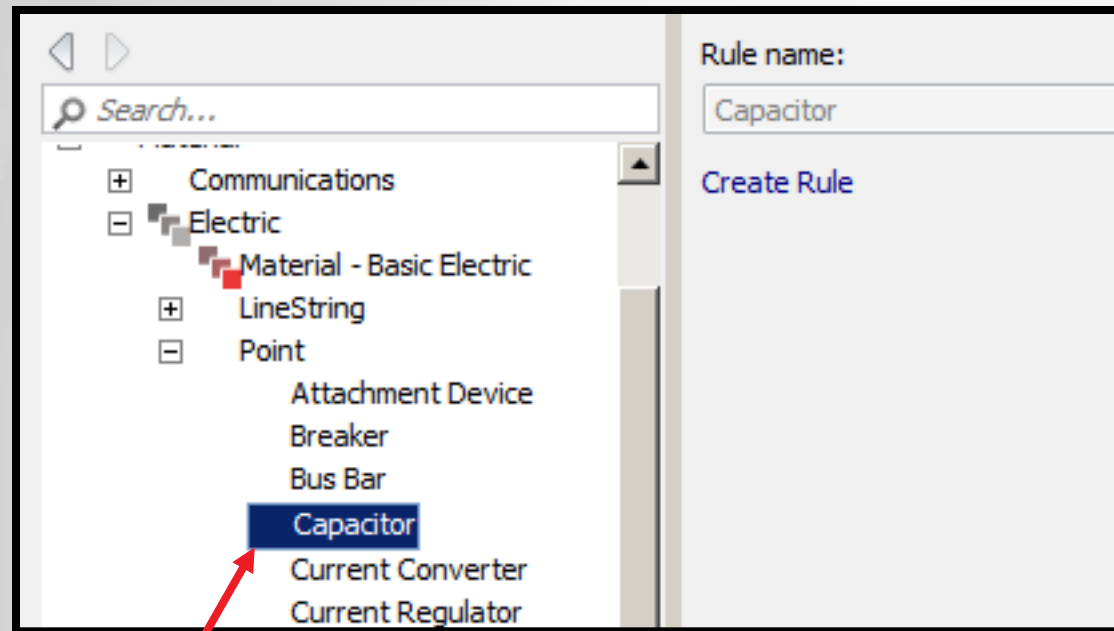
Exercise 2: Creating an advanced styling rule

1. Select the Exercise 2 and 3 view – you will see several Pads
2. Review the available styles for Pads
3. Start the Rule Editor
4. Review the existing style rule for Pads (then delete it)
5. Create a rule to “calculate” the style depending on two attributes
6. Review the results

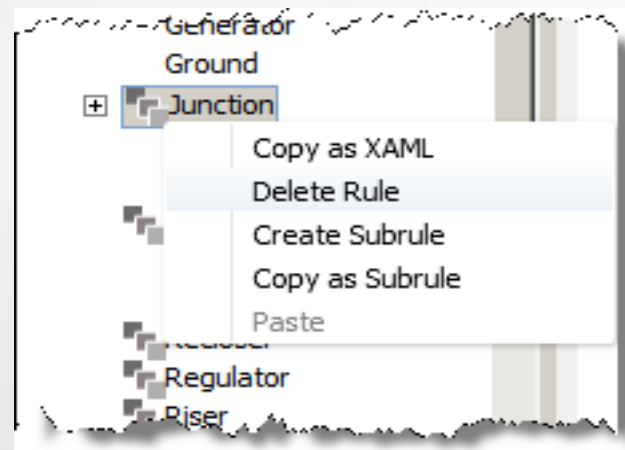
Beyond the Basics

Deleting rules

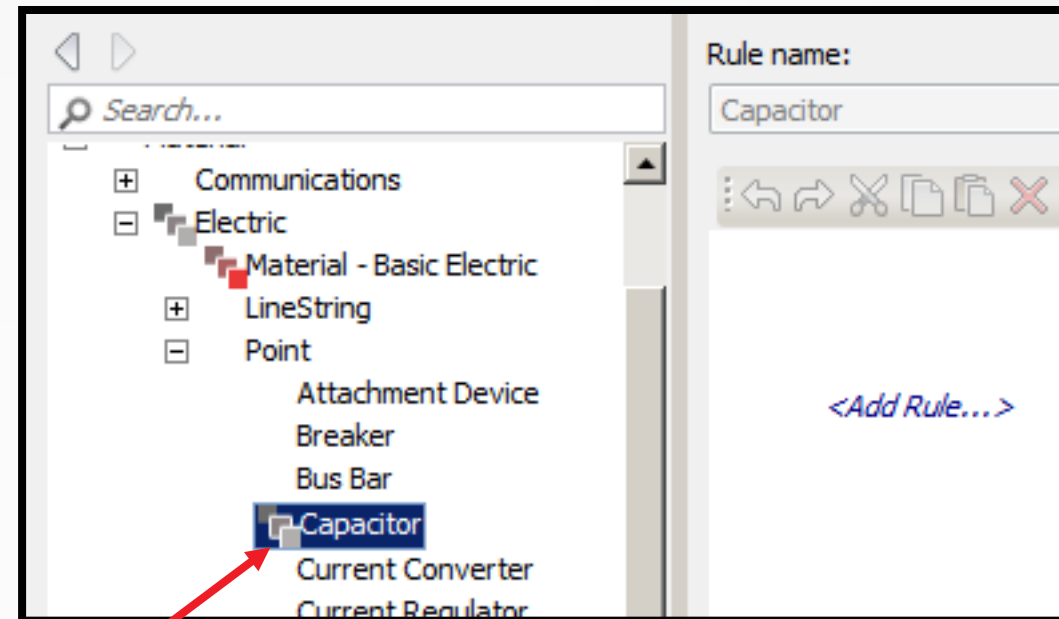
There is NO capacitor rule here



No icon here!



There IS a capacitor rule here



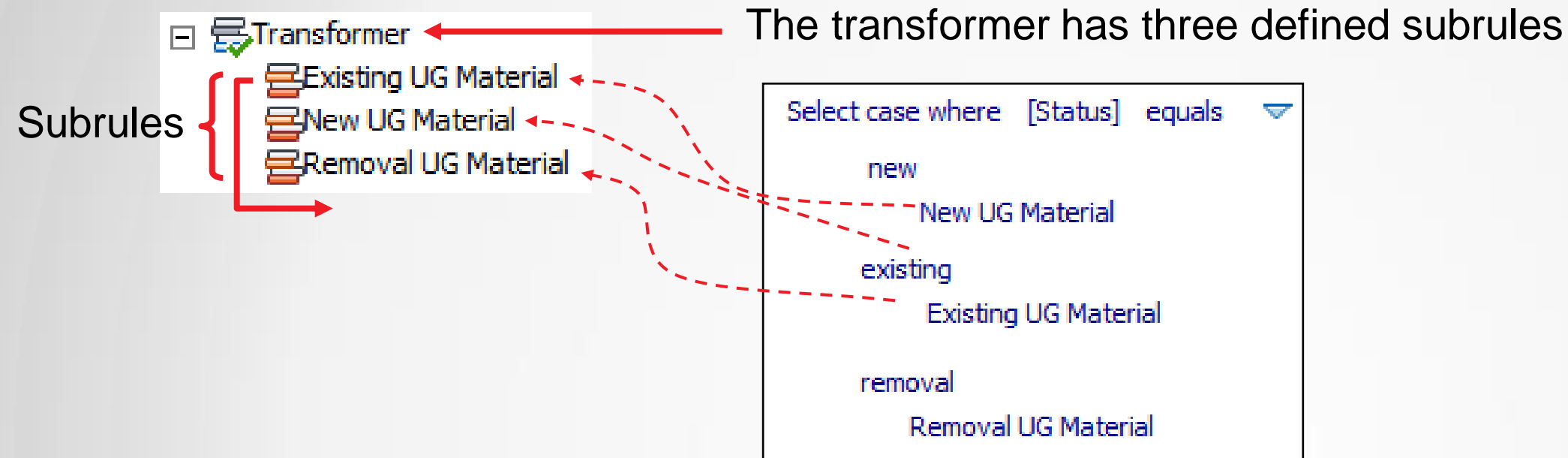
Note the icon

Delete rules by right clicking
on the feature class

Important stuff!

✓ If you delete a rule, make sure it's really deleted!

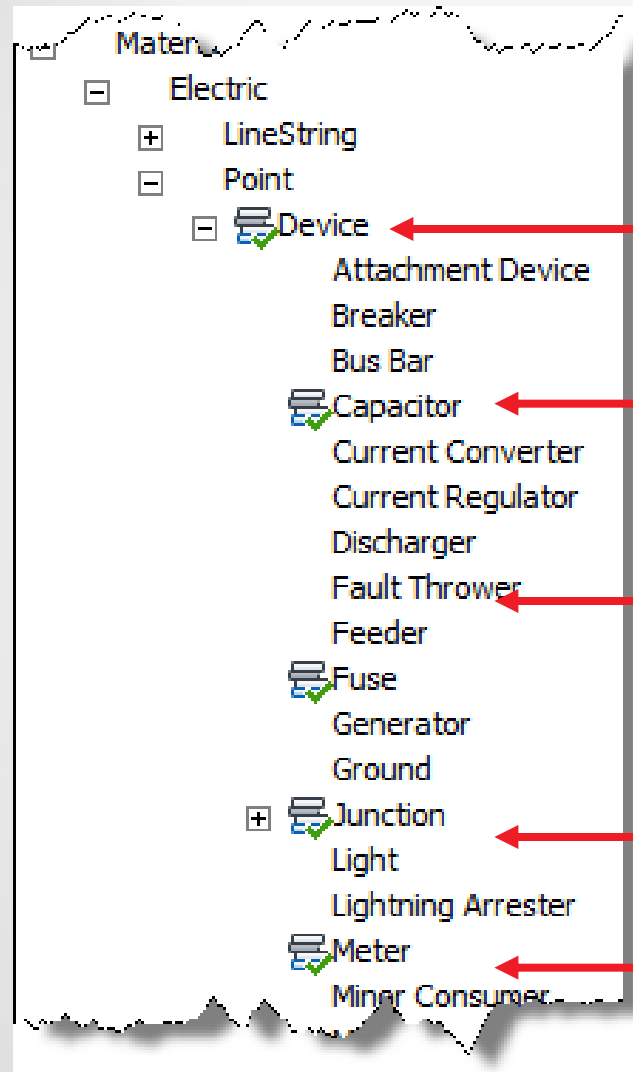
Subrules



Example code found in subrule “Existing UG Material”

```
If (( [Point New Primary Cable] is more than 0 ) and ( [Point Retired Primary Cable] equals 0 )) ▷  
  
If ( [Point New Primary Cable] equals [Point Retired Primary Cable] ) ▷  
  
If (( [Point Retired Primary Cable] is more than 0 ) and ( [Point New Primary Cable] equals 0 )) ▷
```

Rule scope within the rule hierarchy



This rule applies to all point devices
(it could be this typical “default” material rule)

The above rule is superseded by rules defined for these specific devices.
A component rule *completely* supersedes a rule at the device or structure level.
(for example here is a rule that adds a connector)

```
If ( [Status] is not equal to existing )
Then
    Add material Model Name: [Model Name] Quantity: 1 Status: [Status]
    <Set Attribute...>
```

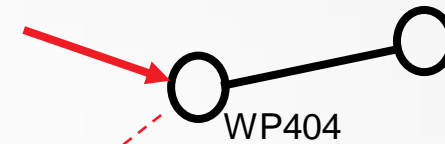
```
If ( [Status] is not equal to existing )
Then
    Add material Model Name: [Model Name] Quantity: 1 Status: [Status]
    <Set Attribute...>

    Add material Model Name: [UJ1-4] Quantity: 1 Status: [Status]
    <Set Attribute...>
```

Material ordering rules

Material Rules define what is in the Material List

1: For each object in the design...



2: ...a material rule executes...

Add material Model Name: [Model Name] Quantity: 1 Status [Status]

3: ...to populate the Material List table

Model Name	Description	Status	Quantity	Work Locati	Raw Cost	Labor Cost	Salva
WP404	40' CLASS 4 WOOD POLE	new	1		400	560	0
WP404	40' CLASS 4 WOOD POLE	removal	1		400	550	500
WP404	40' CLASS 4 WOOD POLE	replace	1		400	1,110	500
UJ1-2	SECONDARY CONNECTOR	new	3		0	0	0

Material rules must consider component status

Select case where [Status Internal] equals
new

[add entry to Material List to order stuff]

existing

[do nothing]

removal

[add entry to Material list to account for removal]

Done using the Add Material rule

Status Internal is the “real” status
(vs. just Status which can have variants)

Important stuff!

- ✓ Make sure there are material rules for everything included in a design

The Add material rule does two things

1: Add entry to Material List

Add material CU:[Model Name] Quantity: 1 Status: [Status]

CU	Quantity	Status	Description	Cost	Installer
WP356	1	new	35' CLASS 6 WOOD POLE	0	Utility

2: Set attributes associated with the Material List entry

Set attribute [Installer] to [ID_Installer_Name]
Set attribute ...

Exercise 3: Configuring a basic material rule

1. Select the Exercise 2 and 3 view showing pads with different status
2. Note there are currently no pads showing in the Material Editor
3. Define a Structural subrule for material ordering
4. Use the subrule to create a material rule for Pads
5. Review the results in the Material Editor

Creating/using expressions

Understanding Expression Rules

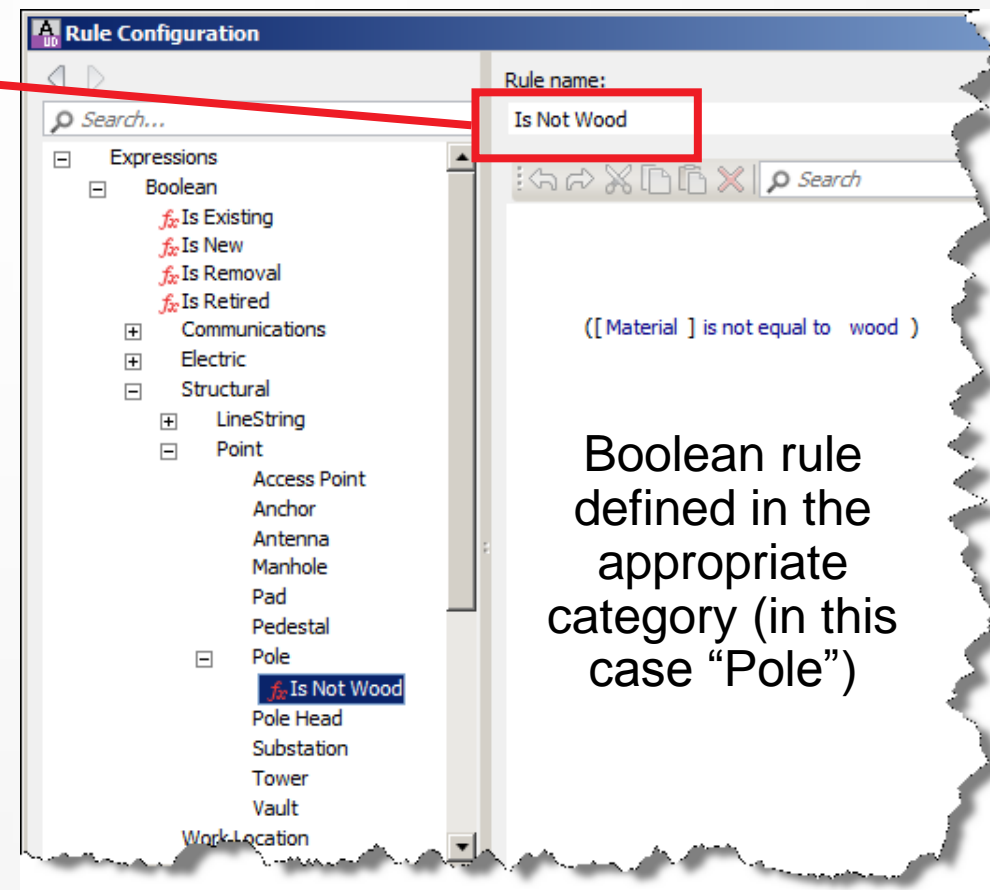
- What are expression rules?
 - Similar to function definitions within a programming language
 - Four types: Boolean, Integer, Real, and Text
 - Define new variables for use in other configuration elements
- Where are expression rules used?
 - To control Industry model attribute visibility
 - To set block attribute values in models and callouts
 - Within other rules
 - and more...

Uses for expressions: Controlling visibility

Visibility choices includes RULE!

Type	Caption	Unit	Name	Category	De	Is Optional	Visibility
Feature	FID	Number	FID		0	<input type="checkbox"/>	Hide
Precision: 18 Scale: 0							
Model	Length	foot	DIMENSION_1			<input checked="" type="checkbox"/>	Show
Model	Pole Type	Domain	ID_POLE_TYPE	Details		<input checked="" type="checkbox"/>	Hide
Model	Support Construction	Domain	ID_SUPPORT_CONSTR			<input checked="" type="checkbox"/>	Hide
Model	Maximum Vertical Load	pound-force	MAX_VERT_LOAD			<input checked="" type="checkbox"/>	Show
Model	Diameter	foot	DIMENSION_2	Details		<input checked="" type="checkbox"/>	Show
Model	XSize	foot	DIMENSION_3	Details		<input checked="" type="checkbox"/>	Show: Is Not Wood
Feature	Disposition State	Domain	ID_DISPOSITION_STAT			<input checked="" type="checkbox"/>	Hide
Feature	Direction X	foot	DIRECTION_3D_X			<input checked="" type="checkbox"/>	Show
Feature	Direction Y	foot	DIRECTION_3D_Y			<input checked="" type="checkbox"/>	Show
Feature	Direction Z	foot	DIRECTION_3D_Z			<input checked="" type="checkbox"/>	Show

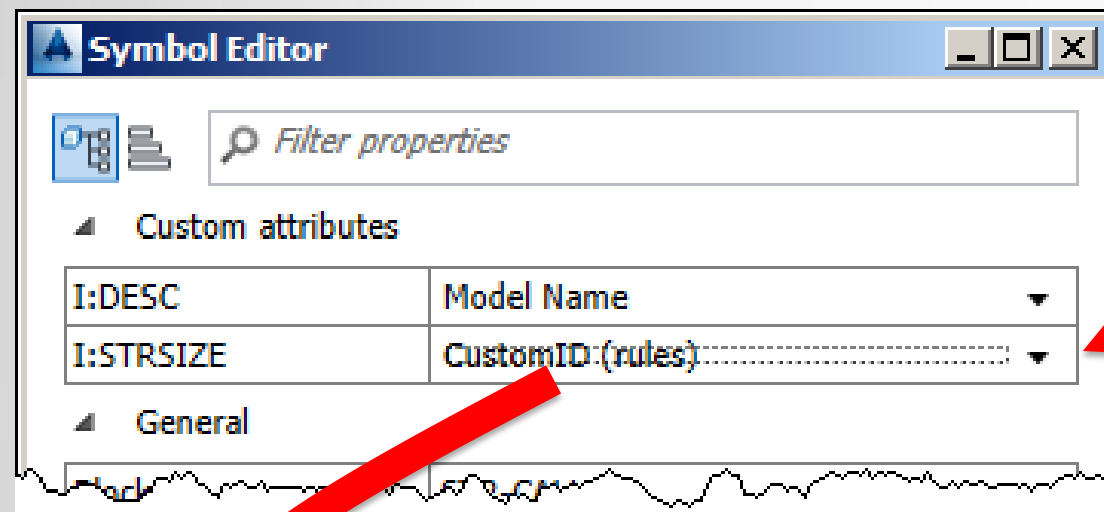
Type	Caption	Unit	Name	Category	De	Is Optional	Visibility
Custom Feature	Pole shape	Domain	NEW_ATTRIBUTE_0	Details	Ro	<input type="checkbox"/>	Show: Is Not Wood
Domain: Pole Shape							



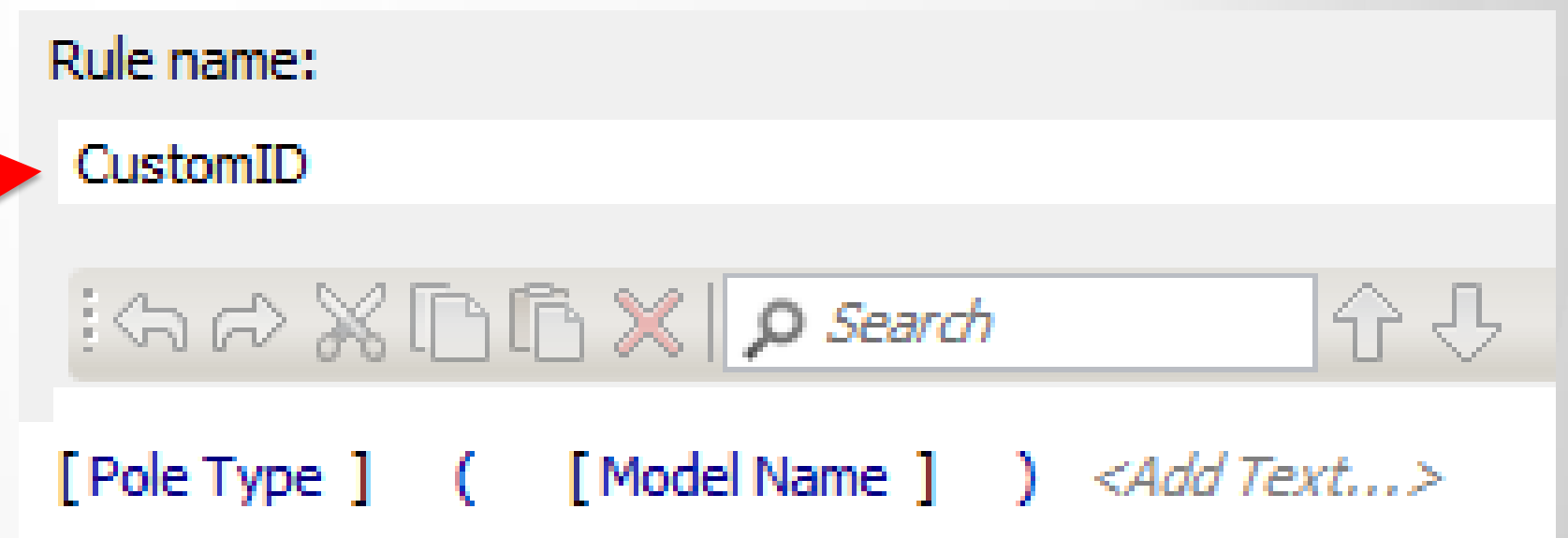
Creating / using expressions

Uses for expressions: Annotation label definitions

Custom attribute definition uses text rule



Rule is defined in Expressions section



Exercise 4: Create and use an Expression Rule

1. Select the Exercise 4 view to see three transformers with attributes
2. Define a new Text Expression rule
3. Use the text rule to improve display of the phase attribute
4. Observe the new display

Annotation rules

Annotation rules

How annotation rules control callouts

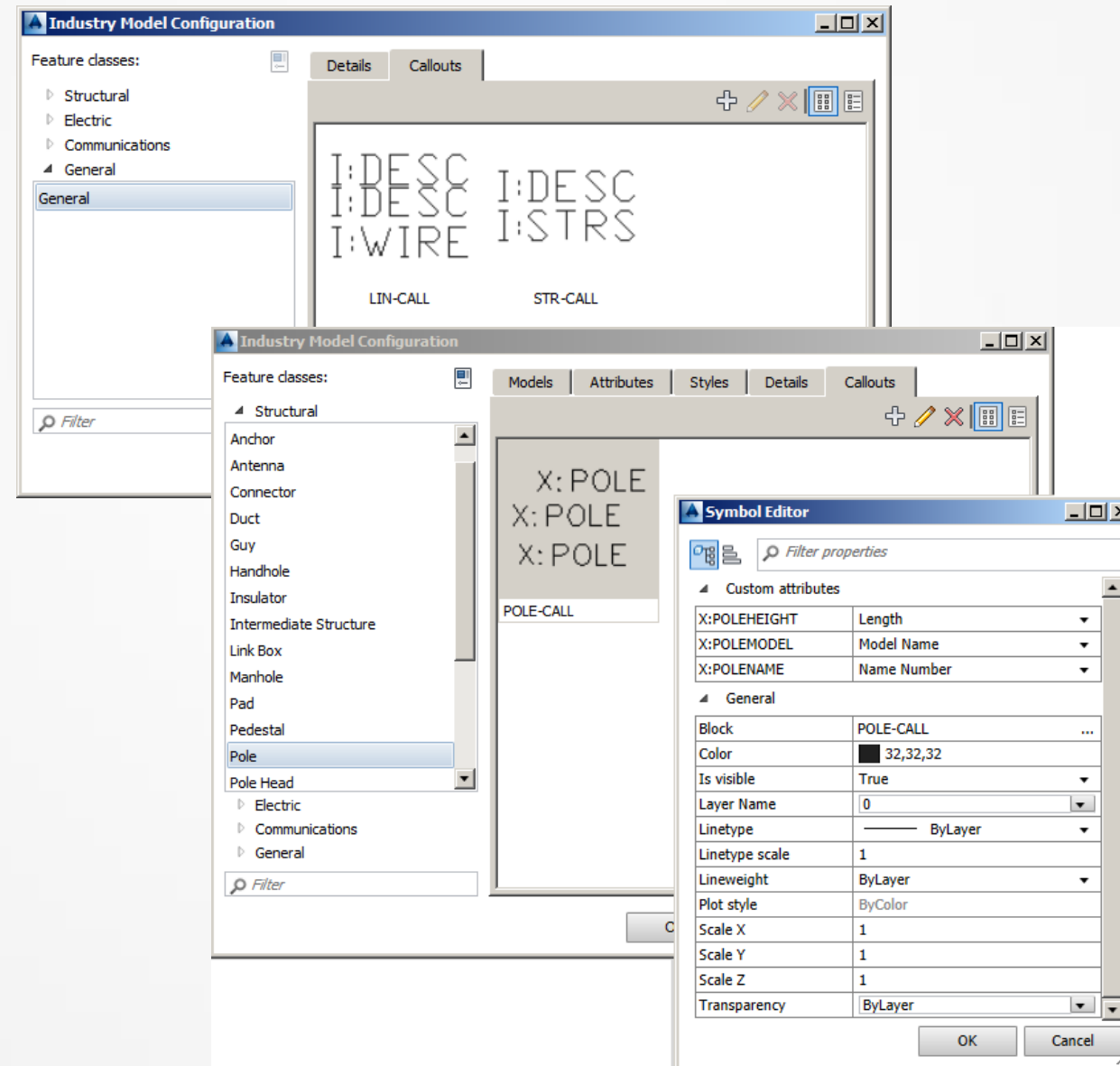
1: Annotation block definitions

- Determine # attributes
- Determine relative attribute placement
- Determine text style



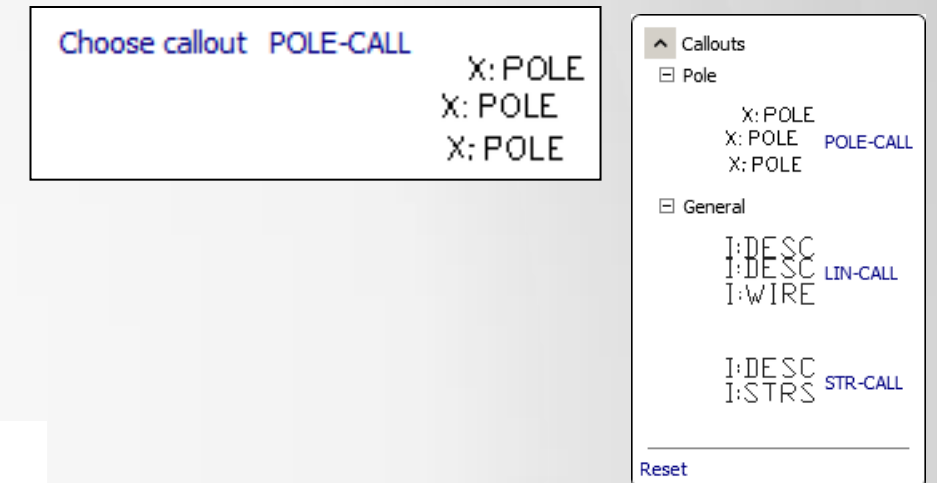
2: Model Definitions (Callout tab)

- Determine how attributes are assigned data
- Determine scaling and color



3: Annotation rules

- Determine which callout definitions is used



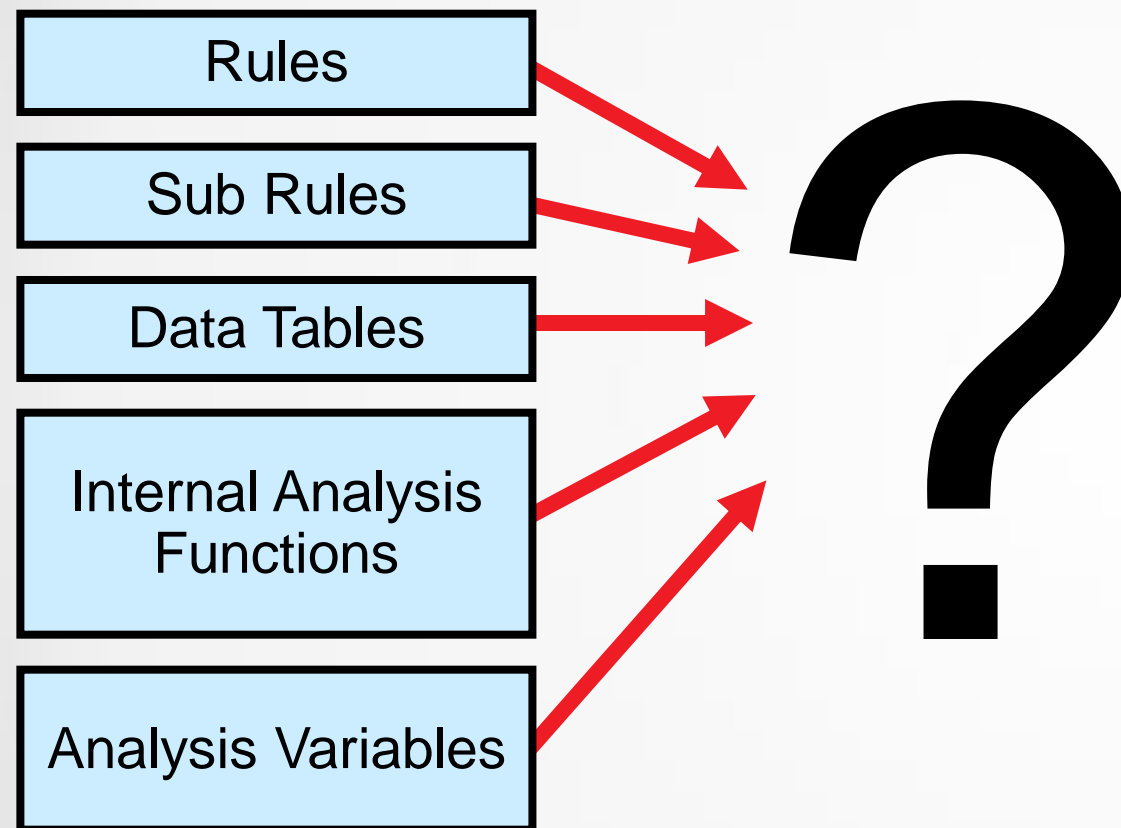
Exercise 5: Configuring annotation rules

1. Select the Exercise 5 view – you will see two lights
2. Configure two different callouts for lights
3. Create callout rule using a different callout based on light status
4. Add callouts and observe behavior

Analyzing advanced rules

Rule analysis

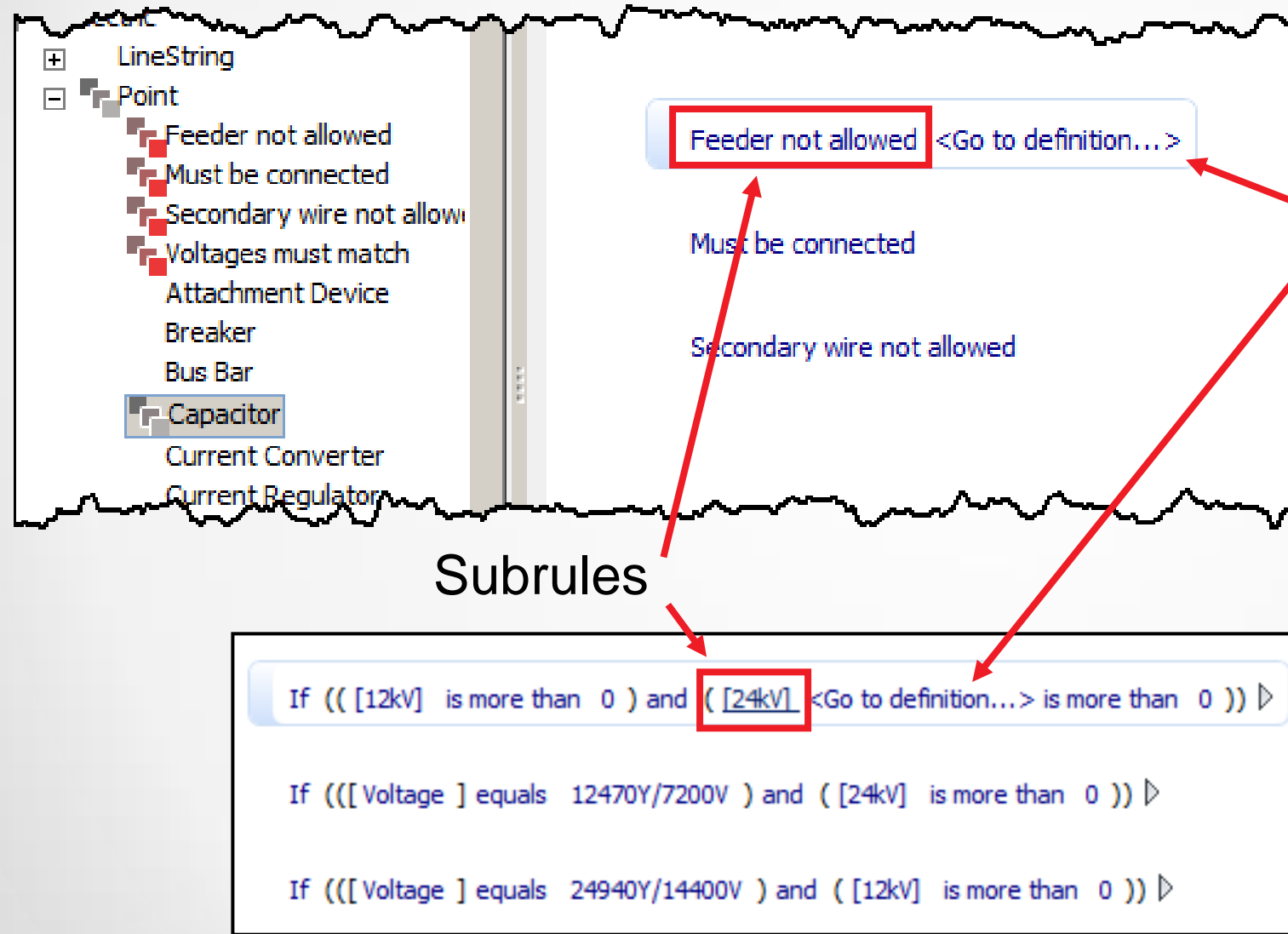
Behavior is driven by a combination of many factors



How do all these factors *combine* to create a specific behavior?

How do you trace rule operation to understand behavior and diagnose problems?

Tip 1: Use the “go to definition” function



When you hover mouse over subrules, “<Go to definition...>” appears

1. Helps you identify subrules!
2. Allows you to jump to the subrule definition.

Analyzing advanced rules

Tip 2: Track down variables

Where is this “Voltage” coming from?

The screenshot shows a rule editor interface. At the top, a rule is displayed: `If (([12kV] is more than 0) and ([24kV] <Go to definition...> is more than 0)) ▷`. Below this, another rule is partially visible: `If (([Voltage] equals 12470Y/7200V) and ([24kV] is more than 0)) ▷`. A red arrow points from the text "Where is this 'Voltage' coming from?" to the `[Voltage]` variable in the second rule. Below the rule editor, a dropdown menu is open, showing a list of categories: **Literal values**, **AUD Variables**, **Document Properties**, **Subrules**, **Point Attributes**, **Operators**, **Functions**, and **Catalog**. A red bracket groups the last four categories, with the text "It is probably one of these!" next to it. To the right of the dropdown, a "Valid Values" list is shown, containing various voltage values: unknown, low voltage, medium voltage, high voltage, public lighting voltage, 12V, 24V, 48V, 52V, 110V, 115V, 120V, 120/240V, 208V, 120/208V, 220V, 230V, 240V, 277V, 400V, and 460V. A red arrow points from the "Valid Values" list back to the `[Voltage]` variable in the rule.

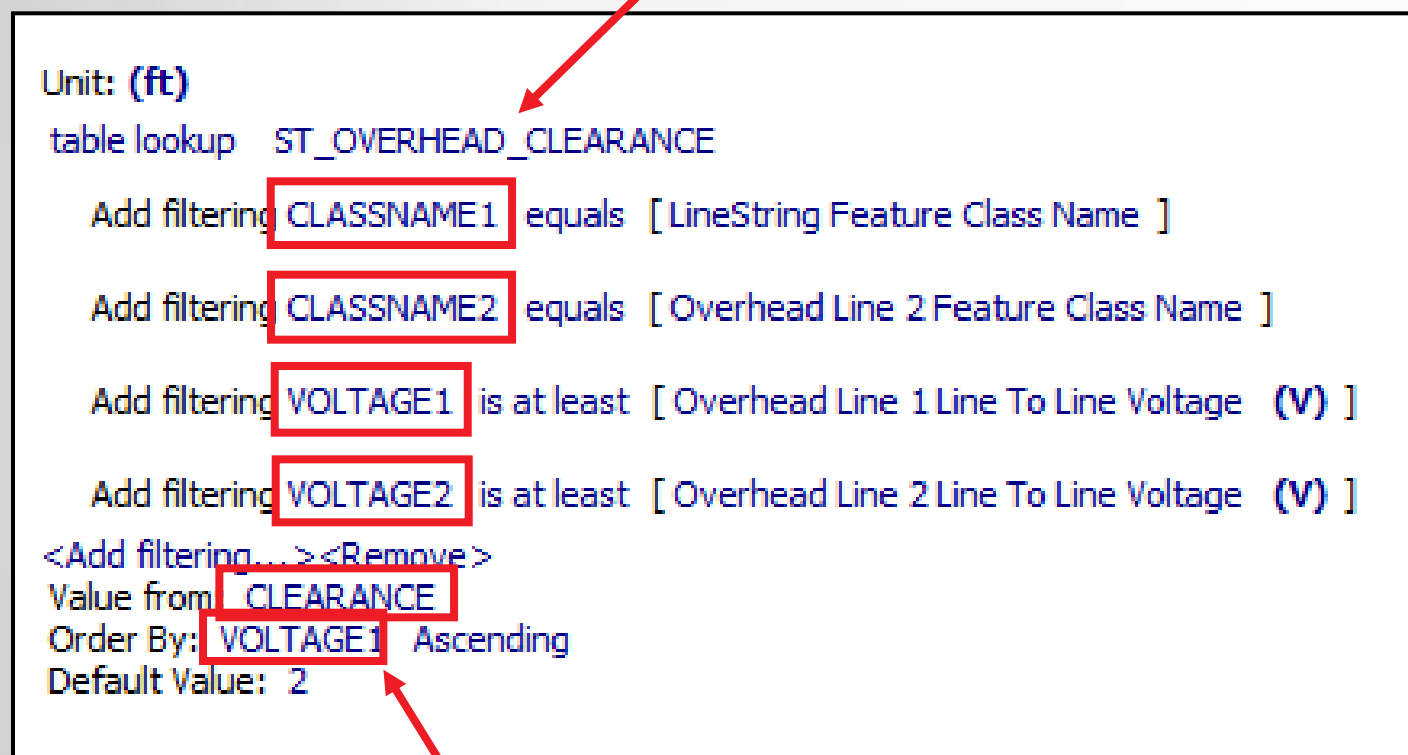
Look through the list of choices to find variables that appear in rules

Explore all the likely types until you find the source of the data

Clicking on the other side of the function can provide a clue...
...in this case Voltage must be a domain variable type

Tip 3: Watch out for “table lookup”

Note this is a “table lookup” clause...



```
Unit: (ft)
table lookup ST_OVERHEAD_CLEARANCE
  Add filtering CLASSNAME1 equals [ LineString Feature Class Name ]
  Add filtering CLASSNAME2 equals [ Overhead Line 2 Feature Class Name ]
  Add filtering VOLTAGE1 is at least [ Overhead Line 1 Line To Line Voltage (V) ]
  Add filtering VOLTAGE2 is at least [ Overhead Line 2 Line To Line Voltage (V) ]
  <Add filtering...><Remove>
  Value from CLEARANCE
  Order By: VOLTAGE1 Ascending
  Default Value: 2
```

...so these variables are defined in that table.

If you see a table referenced, variables in “Add filtering” or “Value from” rules are defined within that table.

Best to immediately look up the table to know what columns it contains.

Analyzing advanced rules

Tip 4: Watch for prefixes and multiple definitions

When there's a chance for ambiguity, a prefix is added to variable names

Unit: (ft)

table lookup ST_OVERHEAD_CLEARANCE

Add filtering CLASSNAME1 equals [LineString Feature Class Name]

Add filtering CLASSNAME2 equals [Overhead Line 2 Feature Class Name]

Add filtering VOLTAGE1 is at least [Overhead Line 1 Line To Line Voltage (V)]

Add filtering VOLTAGE2 is at least [Overhead Line 2 Line To Line Voltage (V)]

<Add filtering...><Remove>

Value from: CLEARANCE

Order By: VOLTAGE1 Ascending

Default Value: 2

Source Feature	Feature Attribute
LineString	Feature Class Name
Overhead Line 2	Feature Class Name
Overhead Line 1	Line To Line Voltage
Overhead Line 2	Line To Line Voltage

Literal values

Overhead Line 2 Line To Line Voltage

Literal value

AUD Variables

Document Properties

Subrules

LineString Attributes

Overhead Line 1 Attributes

Overhead Line 2 Attributes

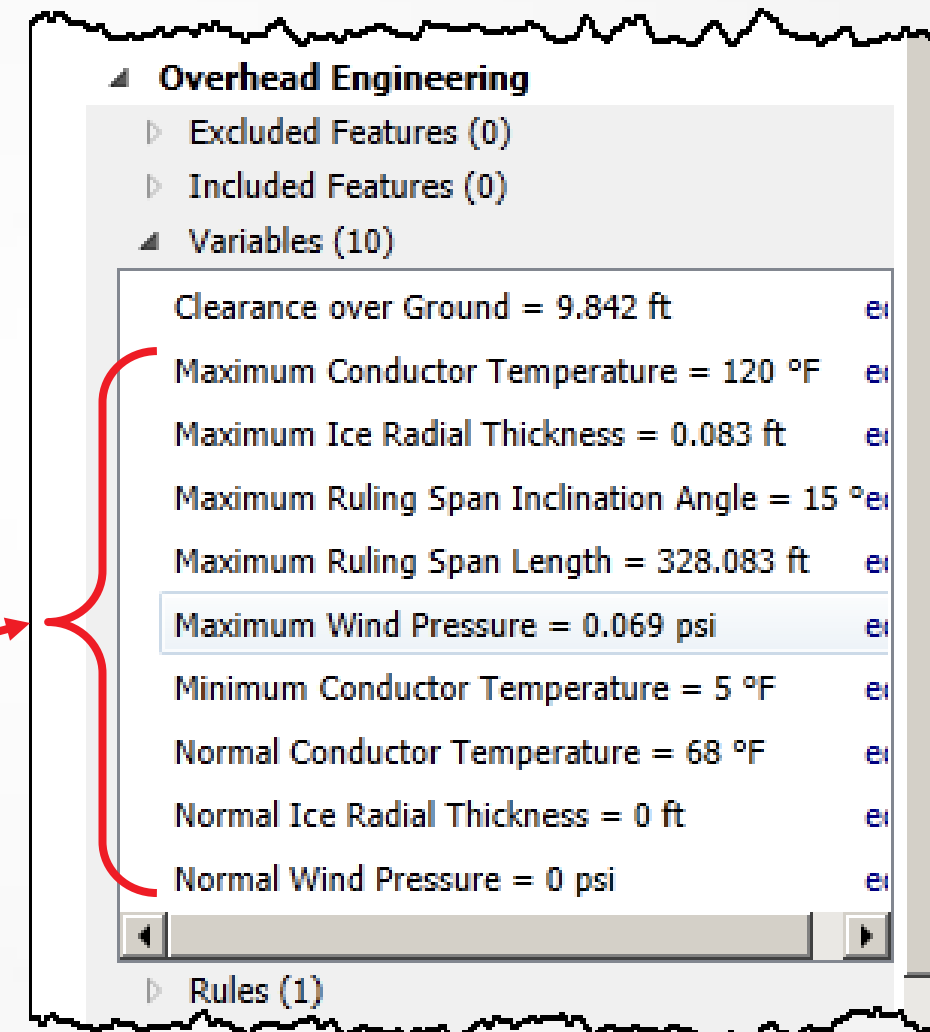
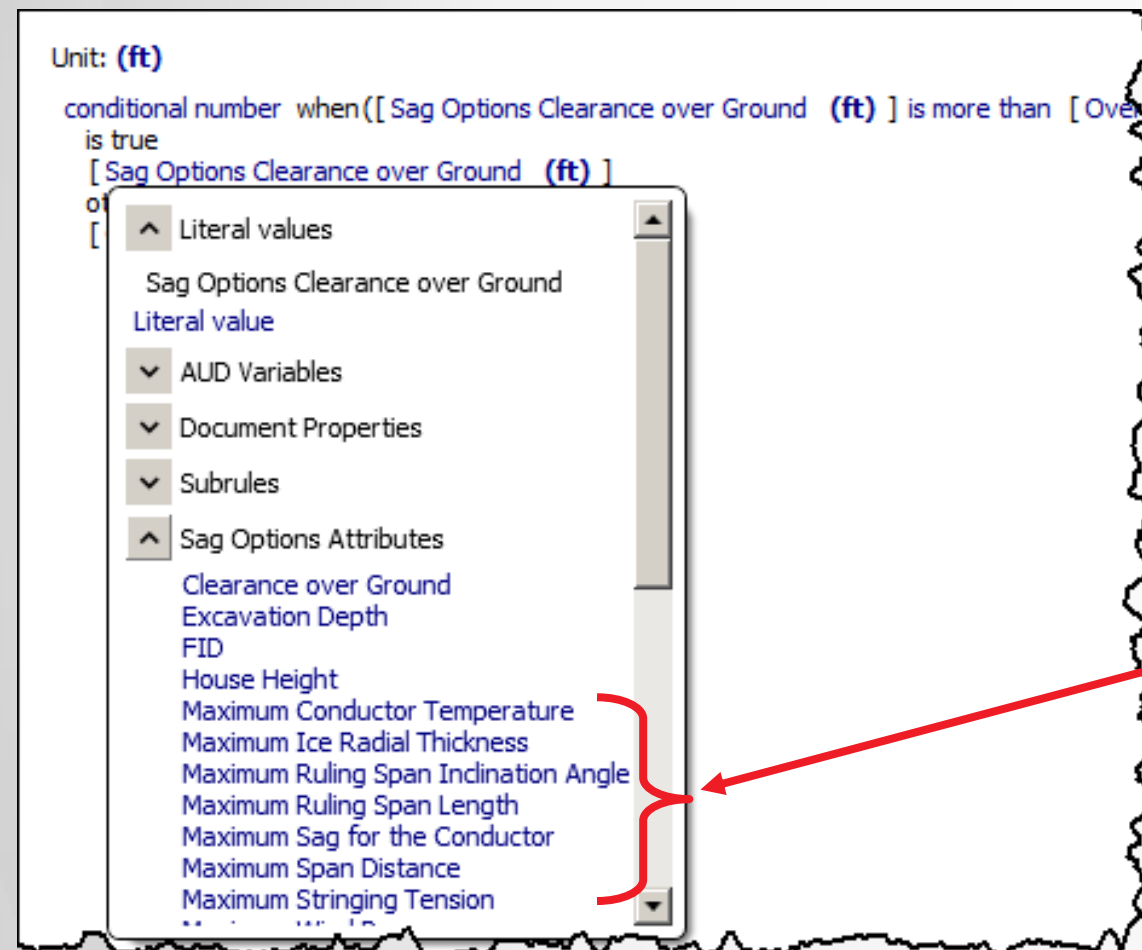
Operators

Where there are multiple feature classes involved, the feature class name is added as a prefix.

Analyzing advanced rules

Tip 5: Recognize internal variables

If you don't recognize something check the Analysis Variables



These are analysis variables (as found on the right) and calculated values from the internal analysis

Closing thoughts

Closing thoughts



When it comes to
creating rules, don't
just jump in and hope
for the best!

Planning for rule implementation

Q1: What *must* rules do accomplish in *your* implementation...

- ✚ ...for automatic identifiers?
- ✚ ...for automatic sizing?
- ✚ ...for validation?
- ✚ ...for automatic sizing?
- ✚ ...for material ordering?
- ✚ ...for material cost calculations?

(you don't need rules for things you don't care about!)

Q2: What “generic” rules can you create?

- ✚ Generic rules can be placed higher in the hierarchy

Q3: Are there parts of rules you keep writing over and over?

- ✚ If so, create subrules and place them high in the hierarchy

Planning for rule implementation

Create a checklist and rule matrix

- ✚ Without one, it's hard to review implementation status

Consider issues around each feature class

- ✚ What special things happen for new / existing / removal / replace
- ✚ Are there considerations where components intersect?

Test an example for each type of rule

- ✚ Know what works before you replicate rules across components

Decide where rules should be placed

- ✚ Example: Size mismatch – which component reports the issue?

Closing thoughts

Decide on your goals for using Utility Design

- ✚ Keep them modest!

Determine your implementation priorities

- ✚ Start with only the essentials

- ✚ Don't "boil the ocean"

Plan and test before implementing

- ✚ Try things and learn what works best for you

Take the time to fully understand Utility Design

Autodesk University Session Feedback

Your session feedback is very important to help improve Autodesk University!

Please complete session survey ASAP

Each completed session survey enters you in that day's drawing for a free AU2014 pass!

