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** • **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

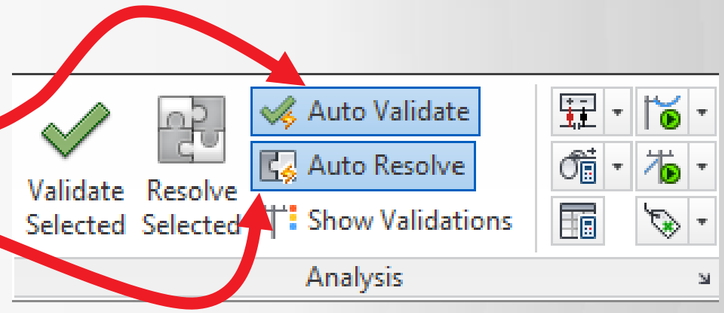
At the end of this class, you will be able to…

- 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

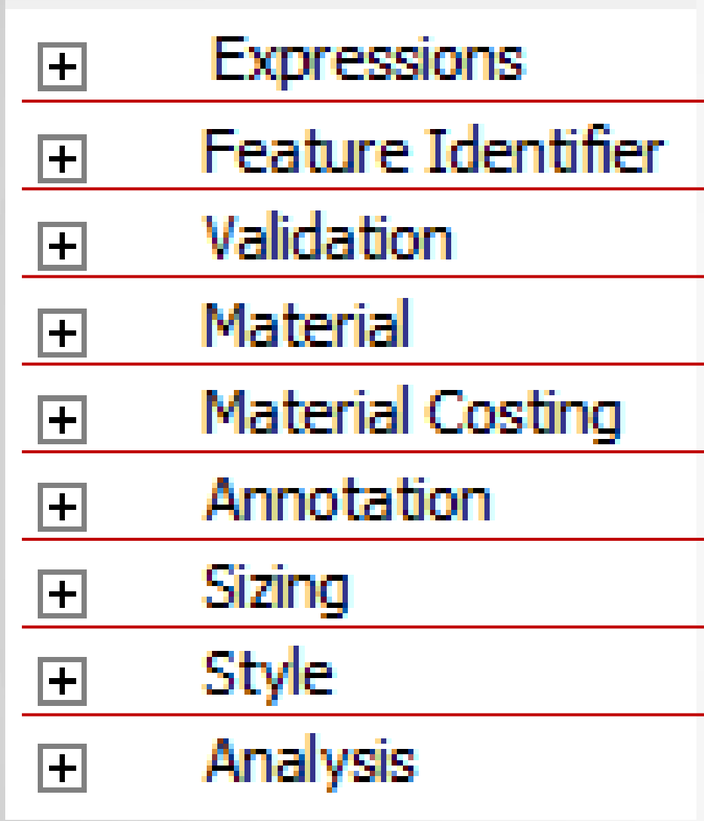
Automatically **add components**

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

What rules do depend on your configuration

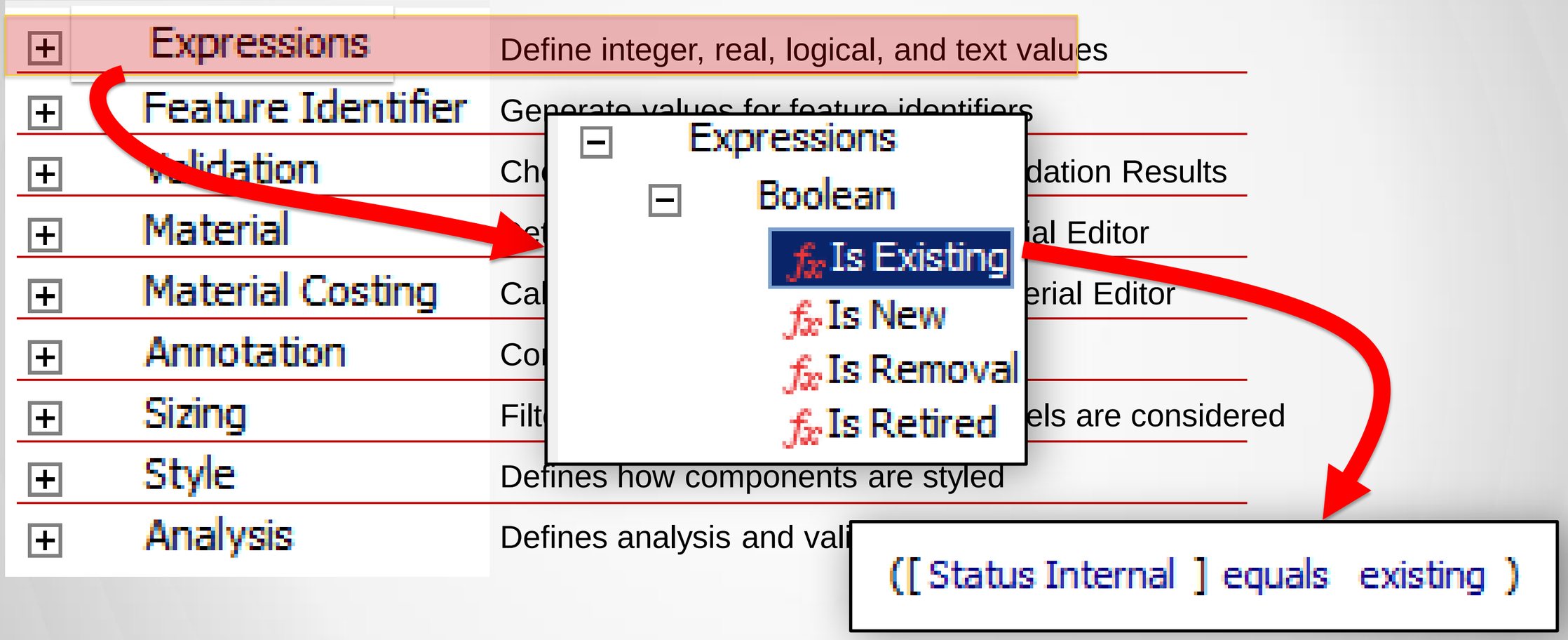
Rules overview

Nine rule categories

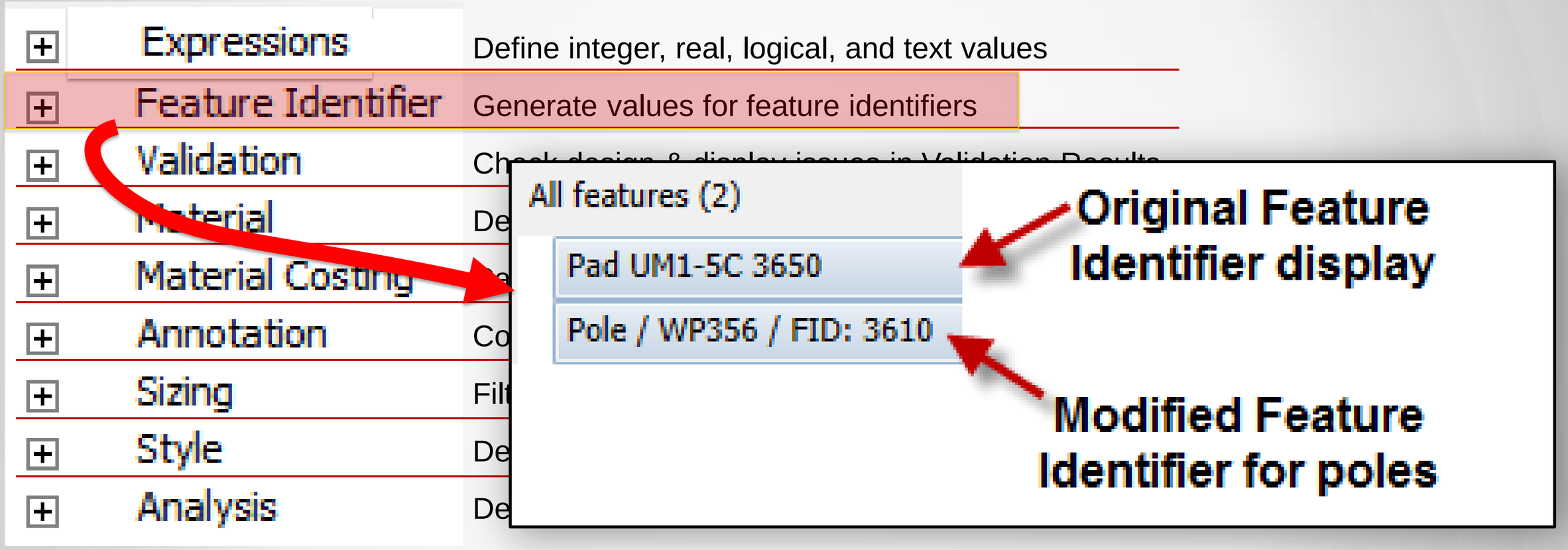


Expressions	Define integer, real, logical, and text values
Feature Identifier	Generate values for feature identifiers
Validation	Check design & display issues in Validation Results
Material	Determine what appears in the Material Editor
Material Costing	Calculate cost of materials in the Material Editor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

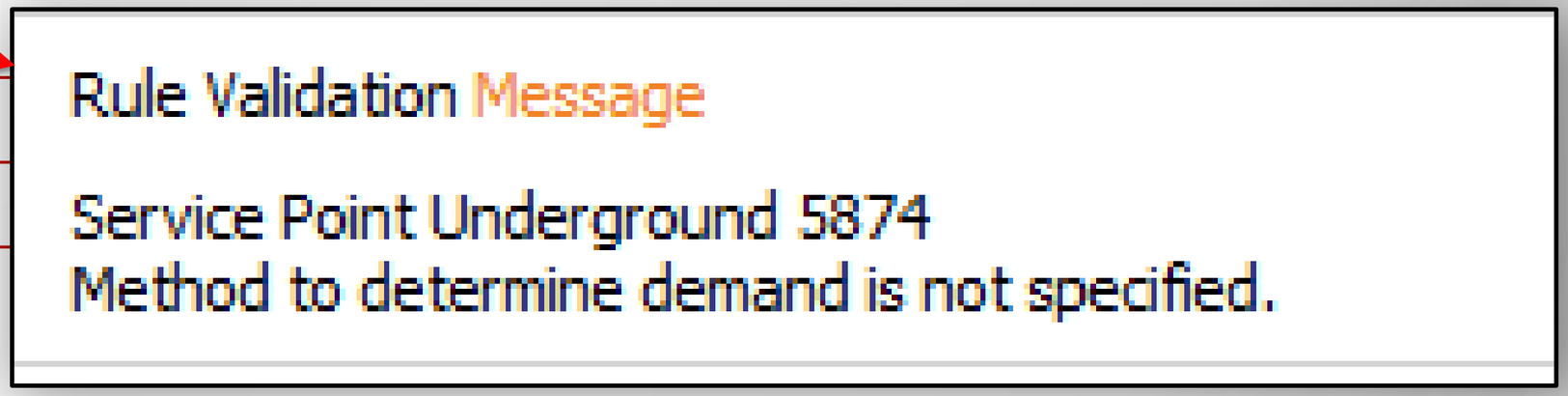


Example: Feature Identifier



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	



Example: Material Rules

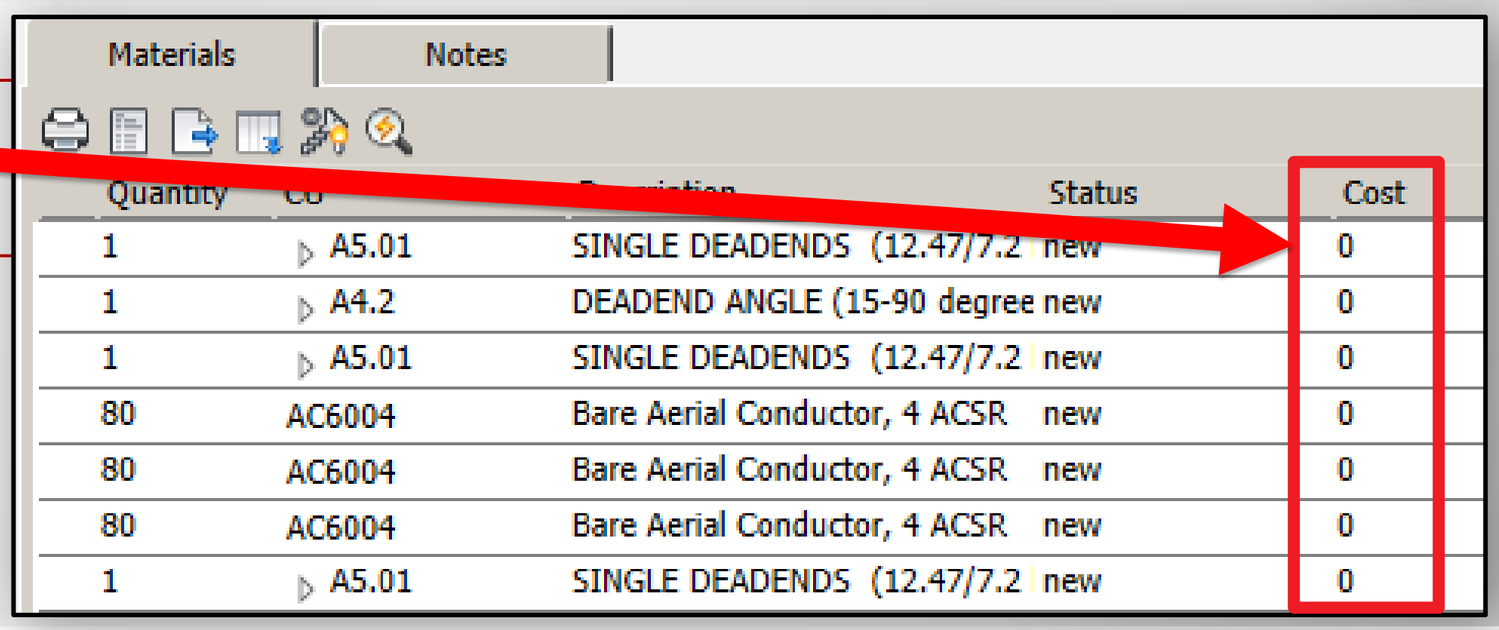
Expressions	Define integer, real, logical, and text values
Feature Identifier	Generate values for feature identifiers
Validation	Check design & display issues in Validation Results
Material	Determine what appears in the Material Editor
Material Costing	Calculate cost of materials in the Material Editor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

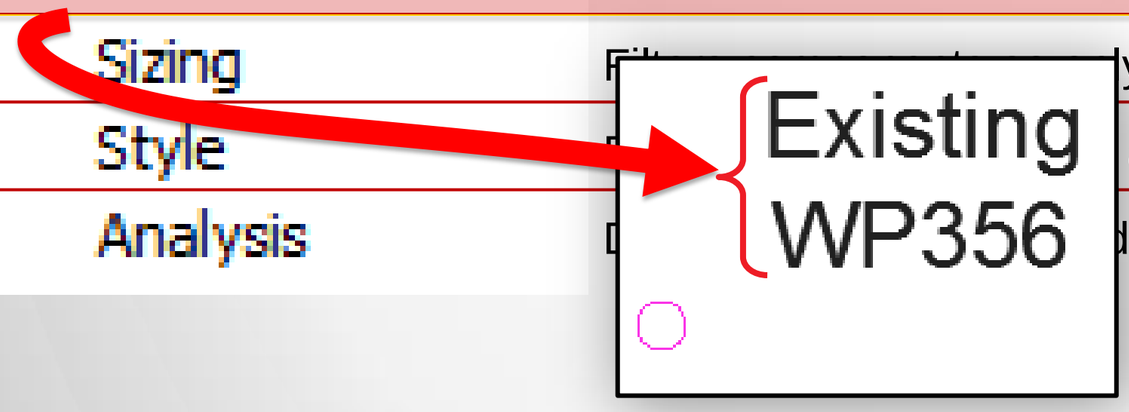
Example: Material Costing Rules

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
Annotation	
Sizing	
Style	
Analysis	



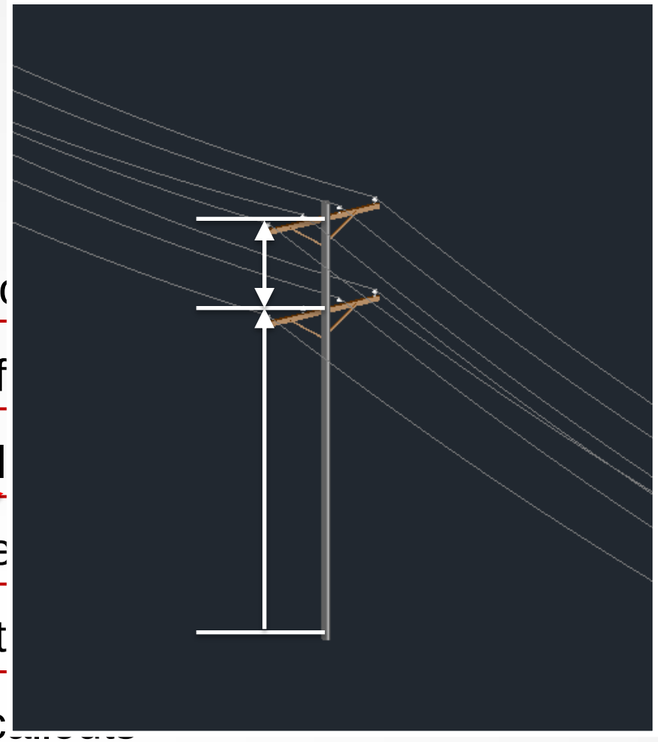
Example: Annotation Rules

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	y valid models are considered
Style	are styled
Analysis	dation equations



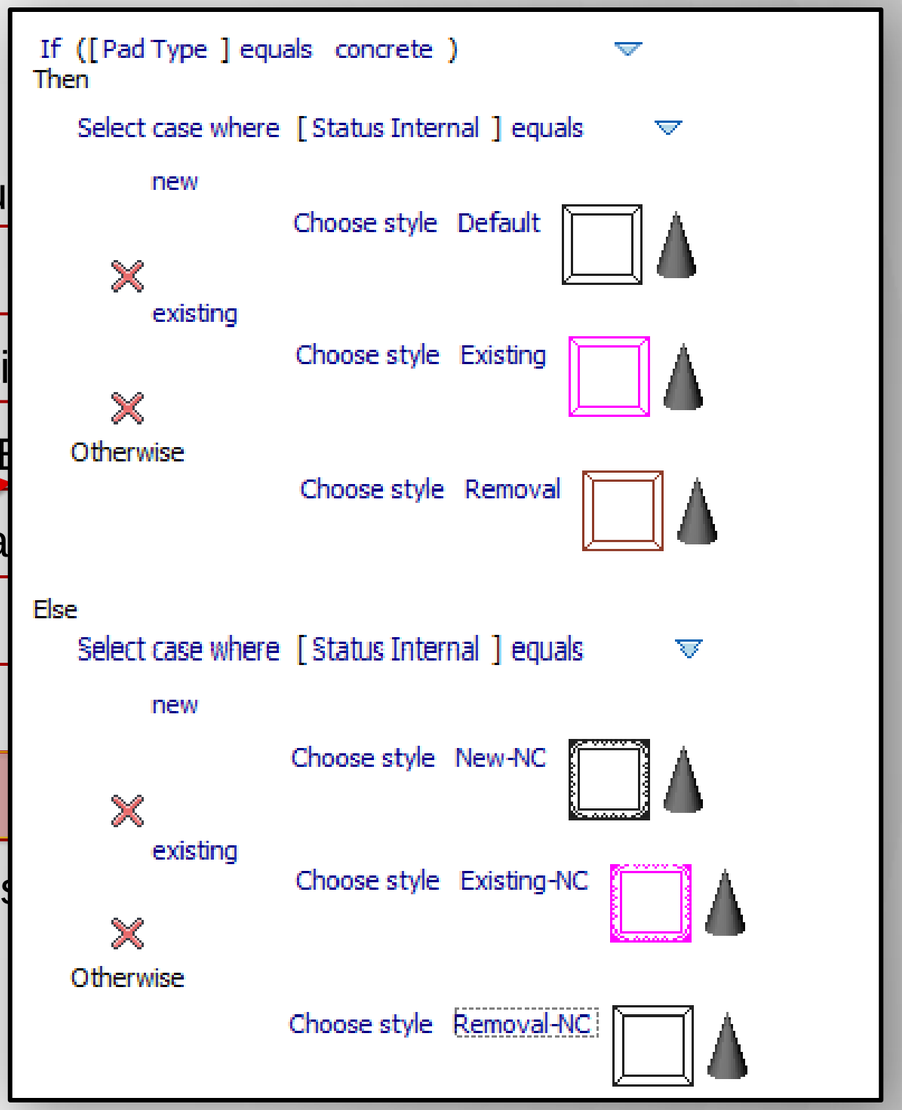
Example: Sizing Rules

Expressions	Define integer, real, l
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



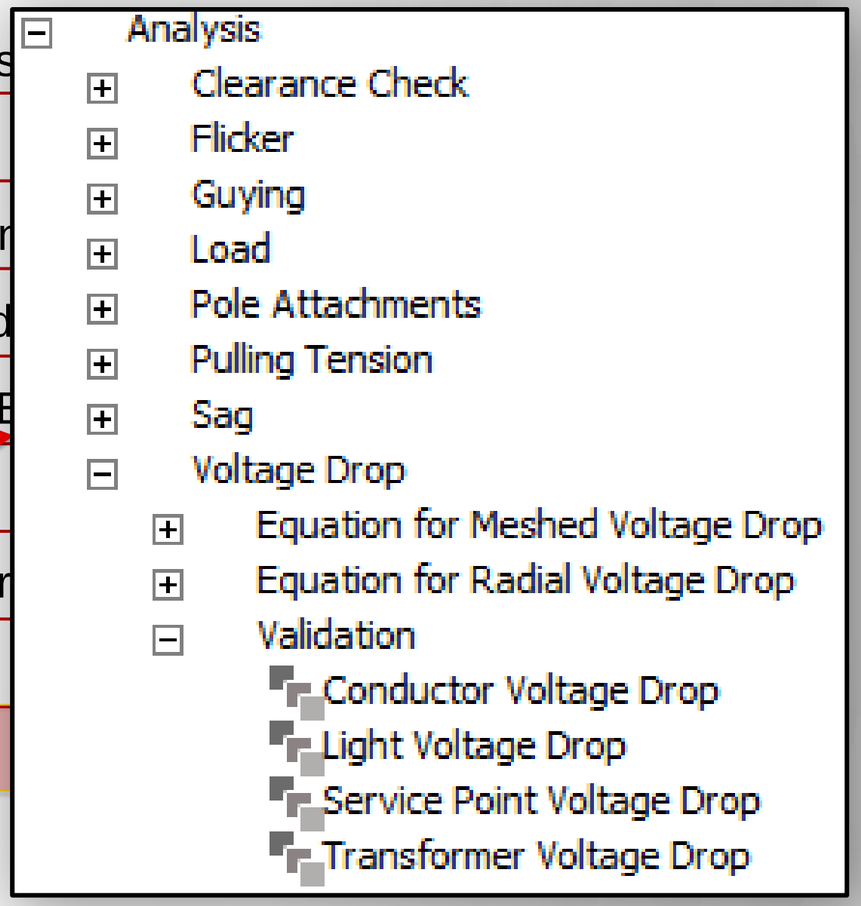
Example: Style Rules

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



Example: Analysis 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 Ed
Material Costing	Calculate cost of materials in the Material E
Annotation	Control location of callouts
Sizing	Filters components so only valid models ar
Style	Defines how components are styled
Analysis	Defines analysis and validation equations



Two types of rules

Expressions (=Functions)

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

Sequential/Action Rules (=Methods)

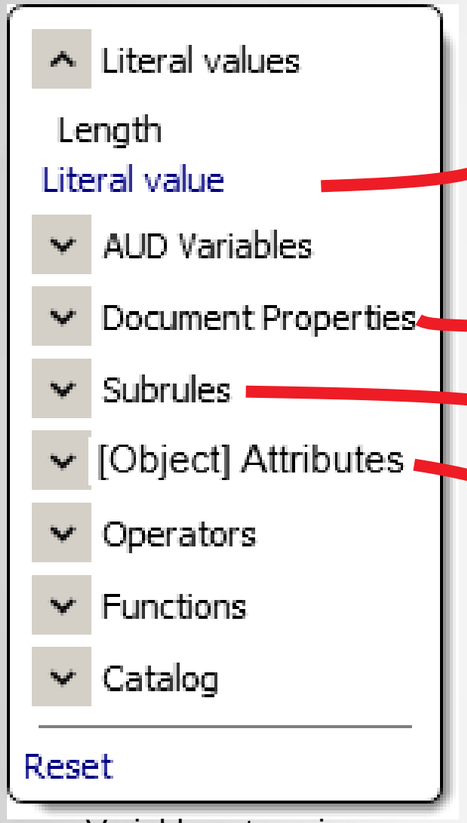
- Sequential/action rules execute a set of instructions
-
```
If ([Status] is not equal to existing )
Then
   Add material Model Name: [Model Name] Quantity: 1 Status: [Status]
```

Basics of Rule Editing

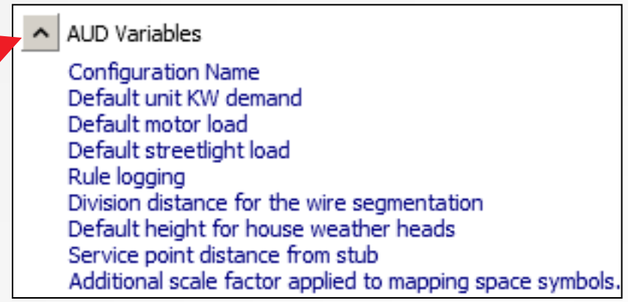
Exercise 1: Basic steps to create/edit a rule

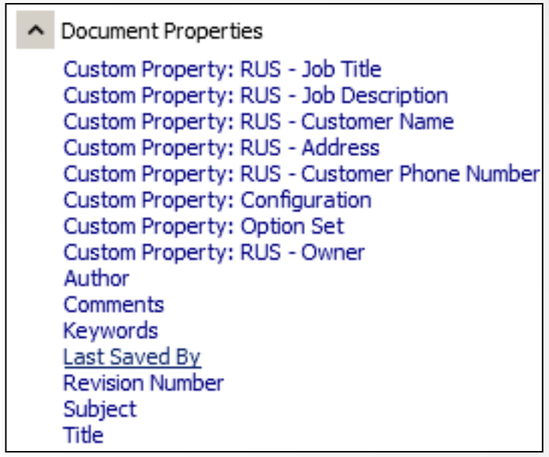
1. Start AutoCAD Utility Design
2. Open the AUD_Rules.dwg file
3. Select the Exercise 1 view – You will see two Vaults
4. Start the Rule Editor
5. Locate and review the style rule for Vaults
6. Change the rule to set a different Style based on Status Internal
7. Note the change in the drawing

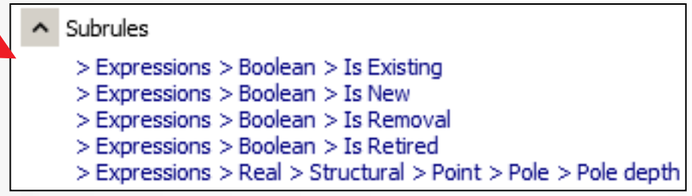
Variables within the rules engine (part 1)

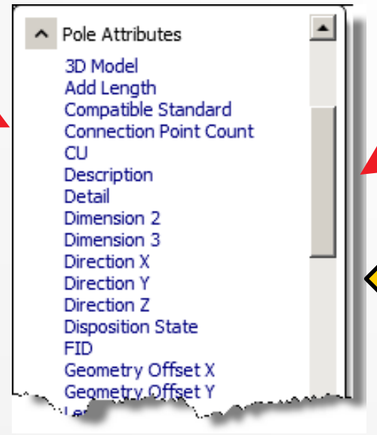


Variable categories





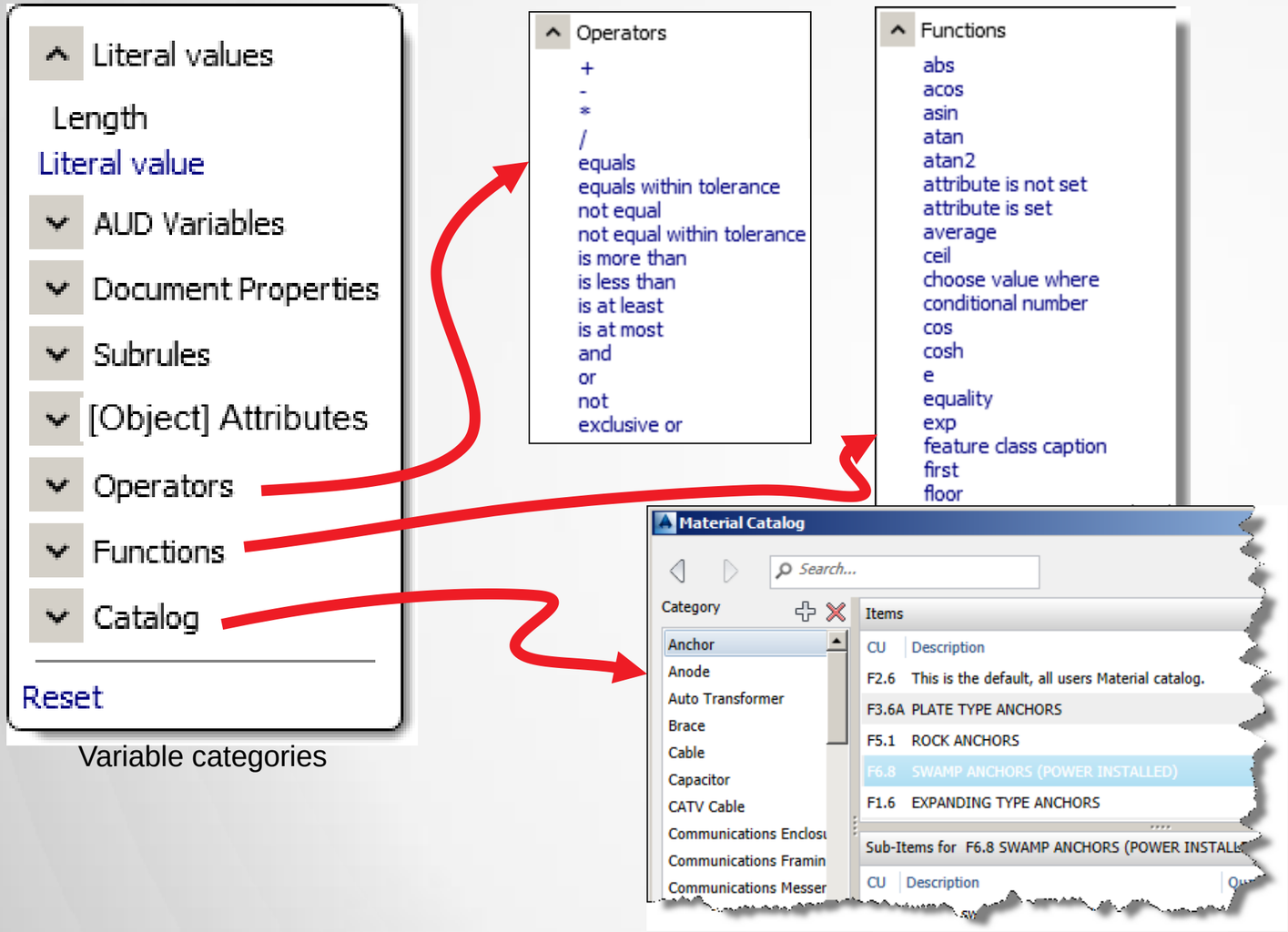




Available variables depend on the object.

Used most often

Variables within the rules engine (part 2)

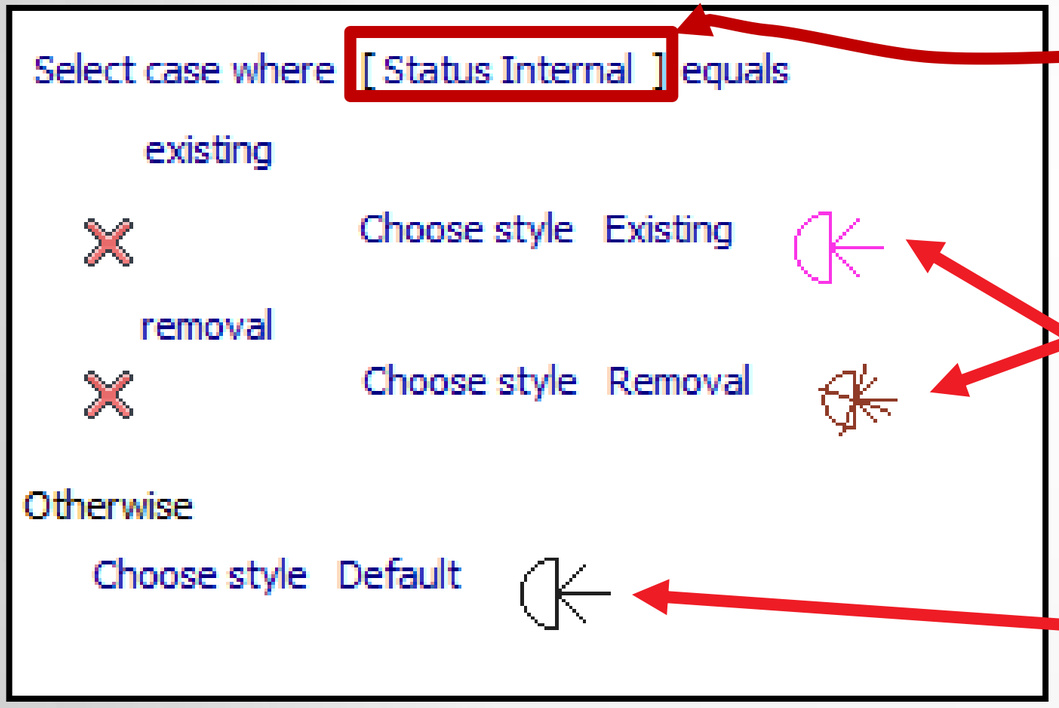


Variable categories

Style Rules

The basic styling rule

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



Case statement using "Status Internal"

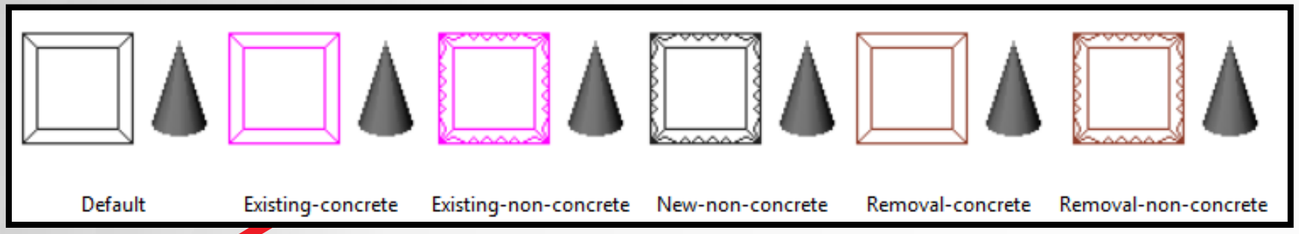
Existing and removal styles explicitly defined

Default style used for "new"

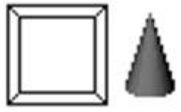
Important stuff!

- ✓ 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

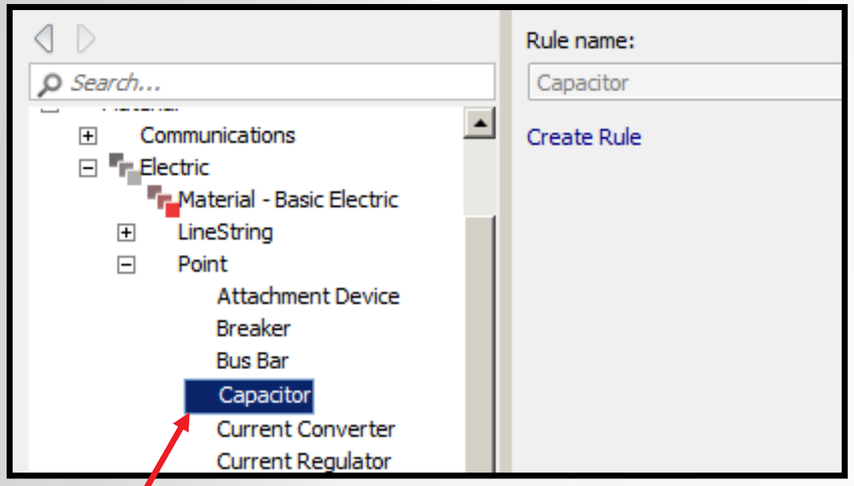
Exercise 2: Creating an advanced styling rule

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

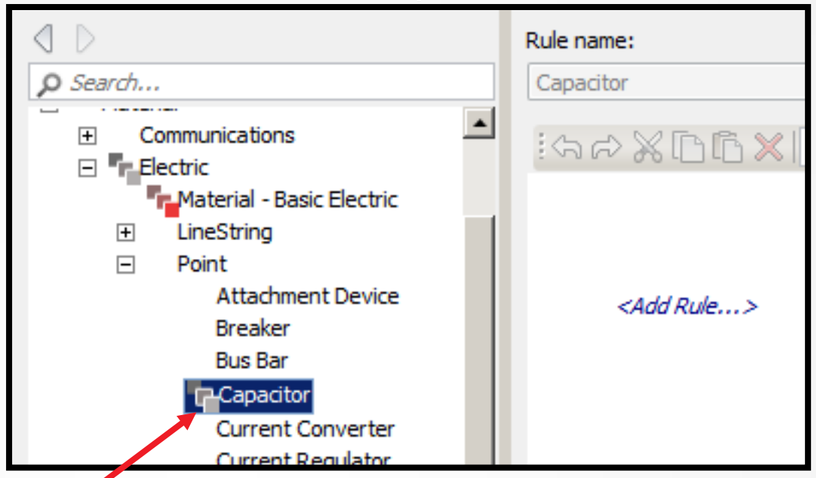
Deleting rules

There is NO capacitor rule here

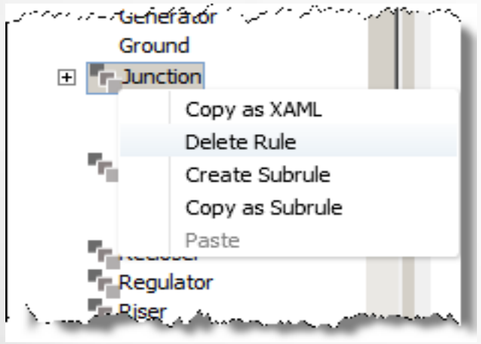


No icon here!

There IS a capacitor rule here



Note the icon

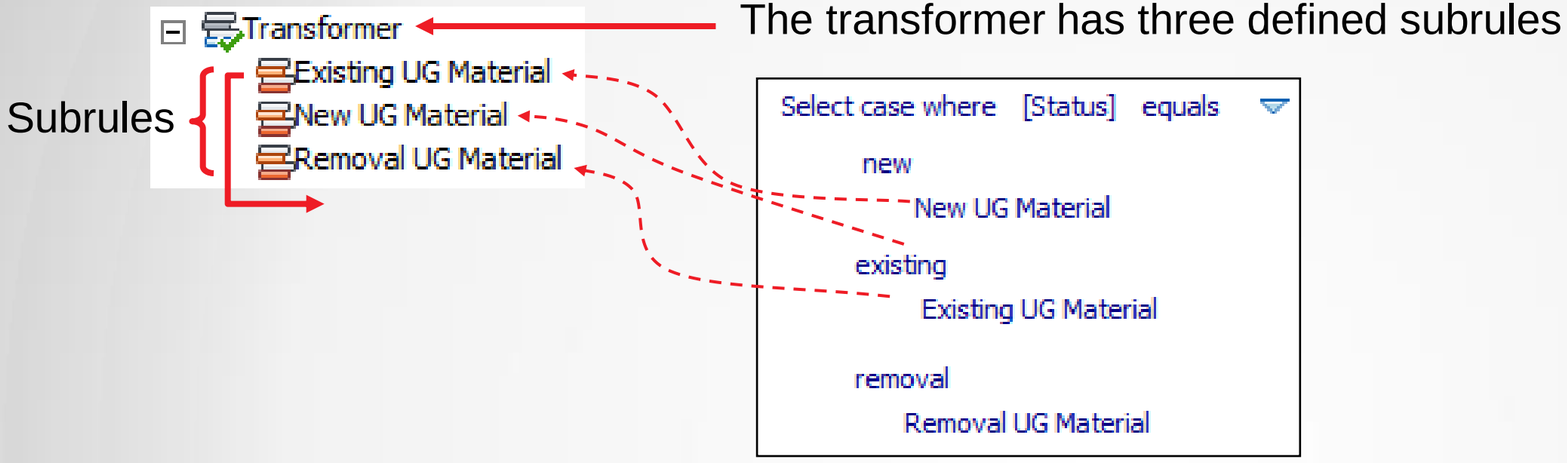


Delete rules by right clicking on the feature class

Important stuff!

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

Subrules



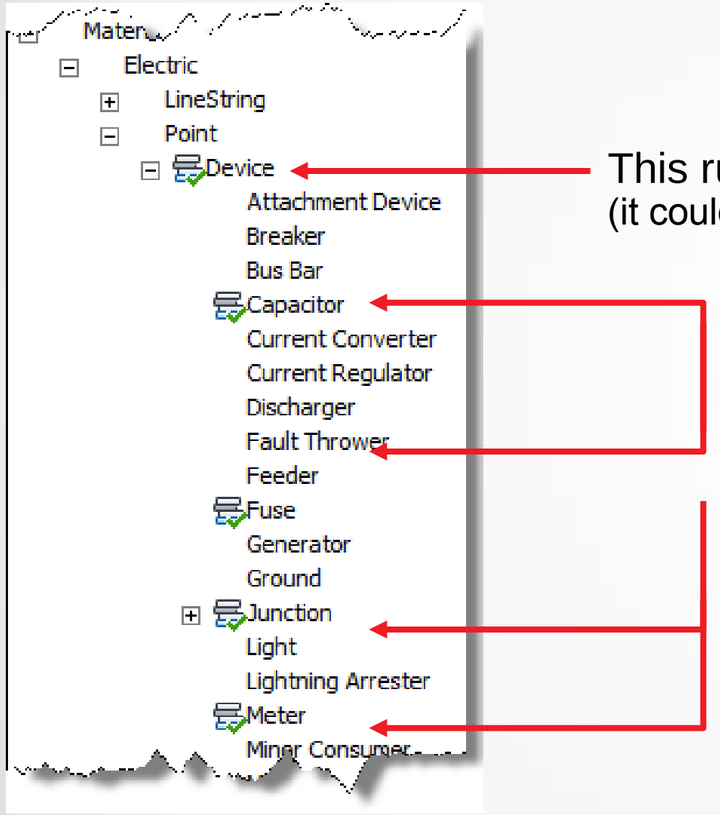
Example code found in subrule “Existing UG Material”

```
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




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

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...>

    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*]

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

Done using the Add Material rule

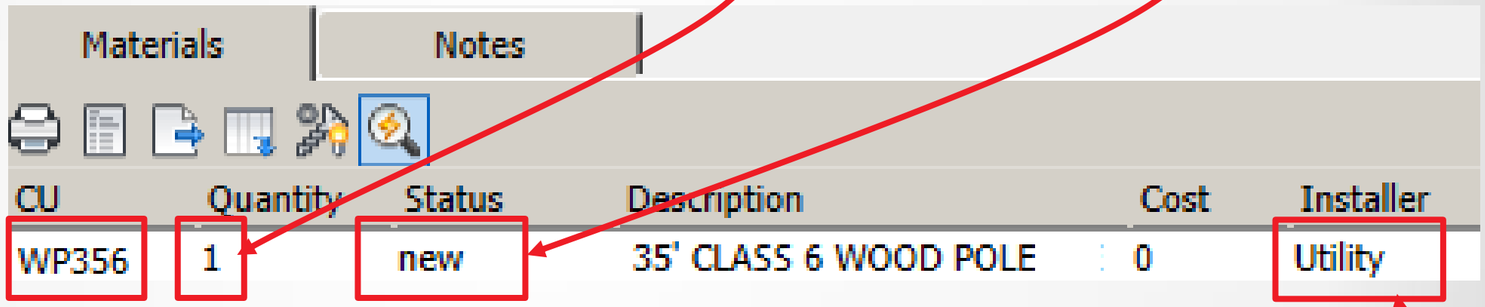
Important stuff!

- ✓ 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]



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

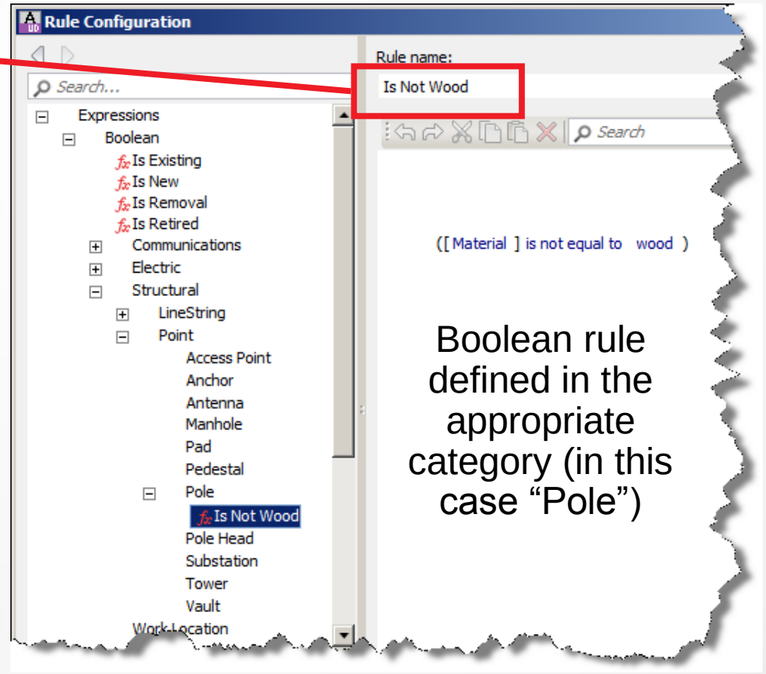
- What are expression rules?
  - Similar to function definitions within a programming language
  - Four types: Boolean, Integer, Real, and Text
  - Define new variables for use in other configuration elements
- Where are expression rules used?
  - To control Industry model attribute visibility
  - To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

Uses for expressions: Controlling visibility

Visibility choices includes RULE!

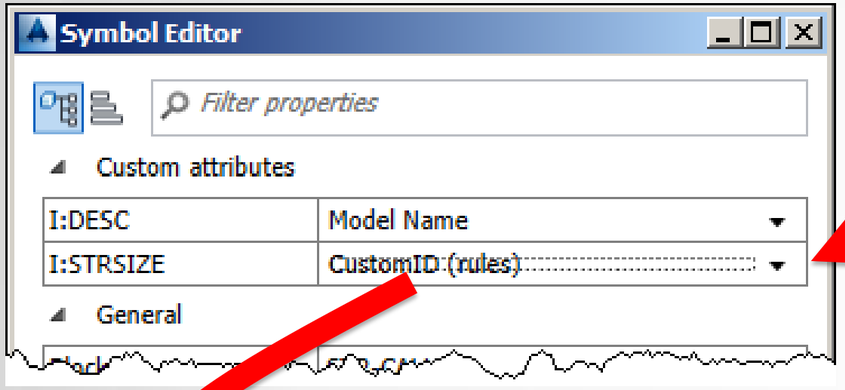
Type	Caption	Unit	Name	Category	De	Is Optional	Visibility
Feature	FID	Number	FID		0	☐	Hide
Precision: 18 Scale: 0							
Model	Length	foot	DIMENSION_1			☑	Show
Model	Pole Type	Domain	ID_POLE_TYPE	Details		☑	Hide
Model	Support Construction	Domain	ID_SUPPORT_CONSTRL			☑	Hide
Model	Maximum Vertical Load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

Type	Caption	Unit	Name	Category	De	Is Optional	Visibility
Custom Feature	Pole shape	Domain	NEW_ATTRIBUTE_0	Details	Ro	☐	Show: Is Not Wood
Domain: Pole Shape							

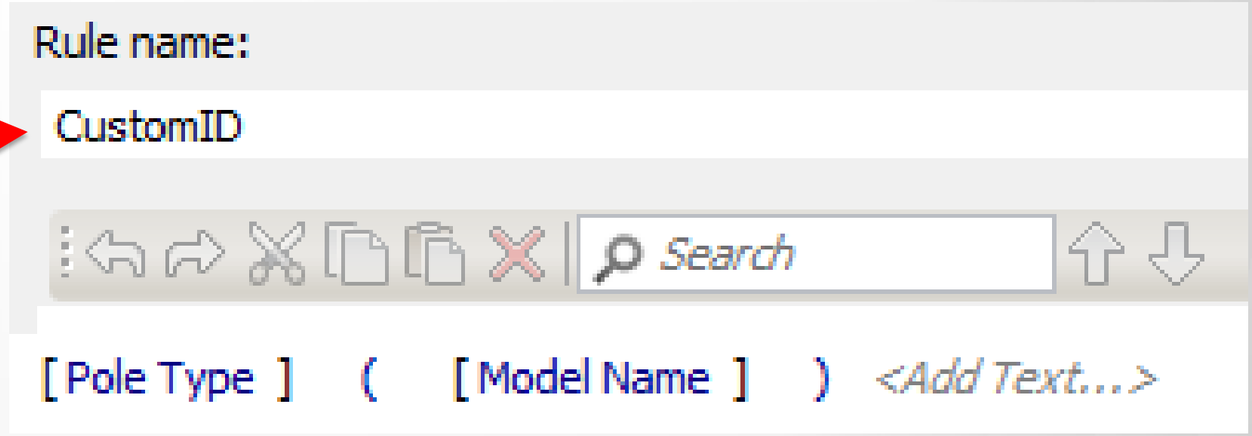


Uses for expressions: Annotation label definitions

Custom attribute definition uses text rule



Rule is defined in Expressions section



Exercise 4: Create an 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

How annotation rules control callouts

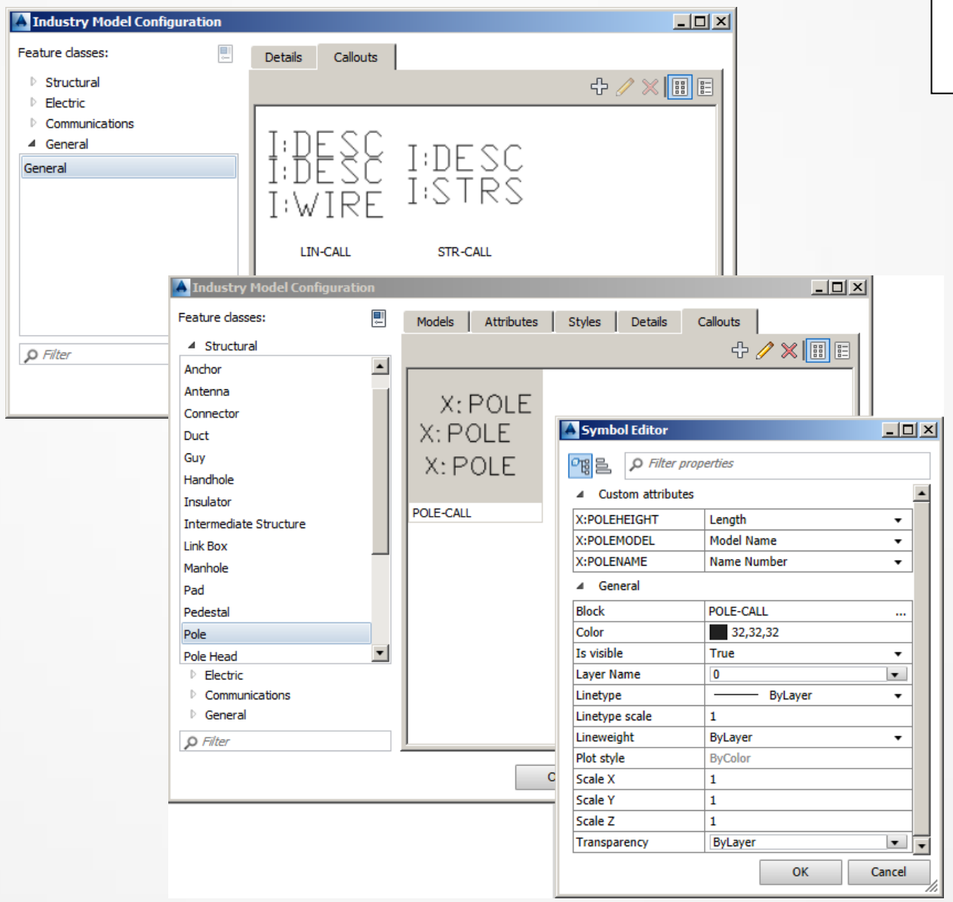
1: Annotation block definitions

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

Block: “STR-CALL”

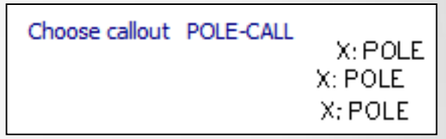
2: Model Definitions (Callout tab)

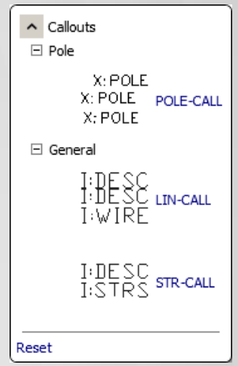
- Determine how attributes are assigned data
- Determine scaling and color



3: Annotation rules

- Determine which callout definitions is used





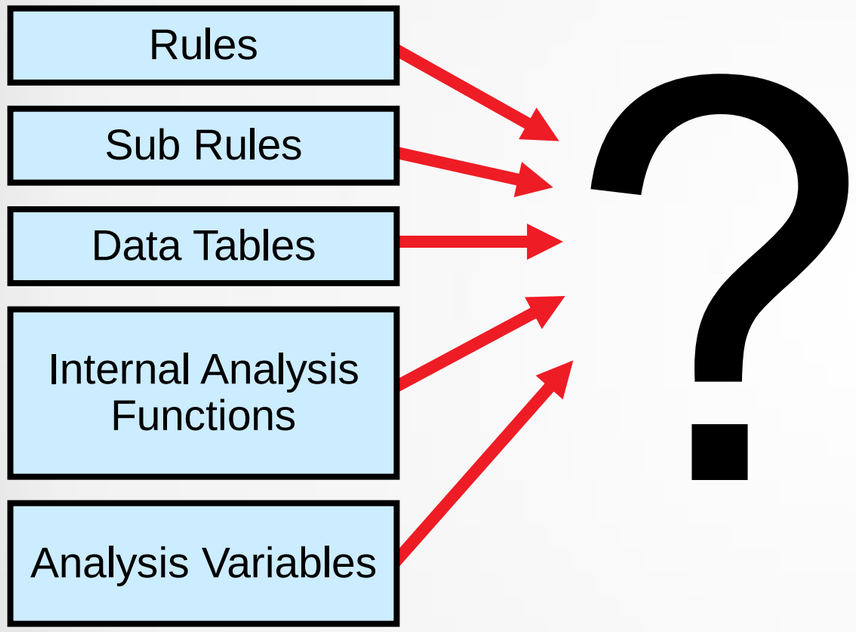
Exercise 5: Configuring annotation rules

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

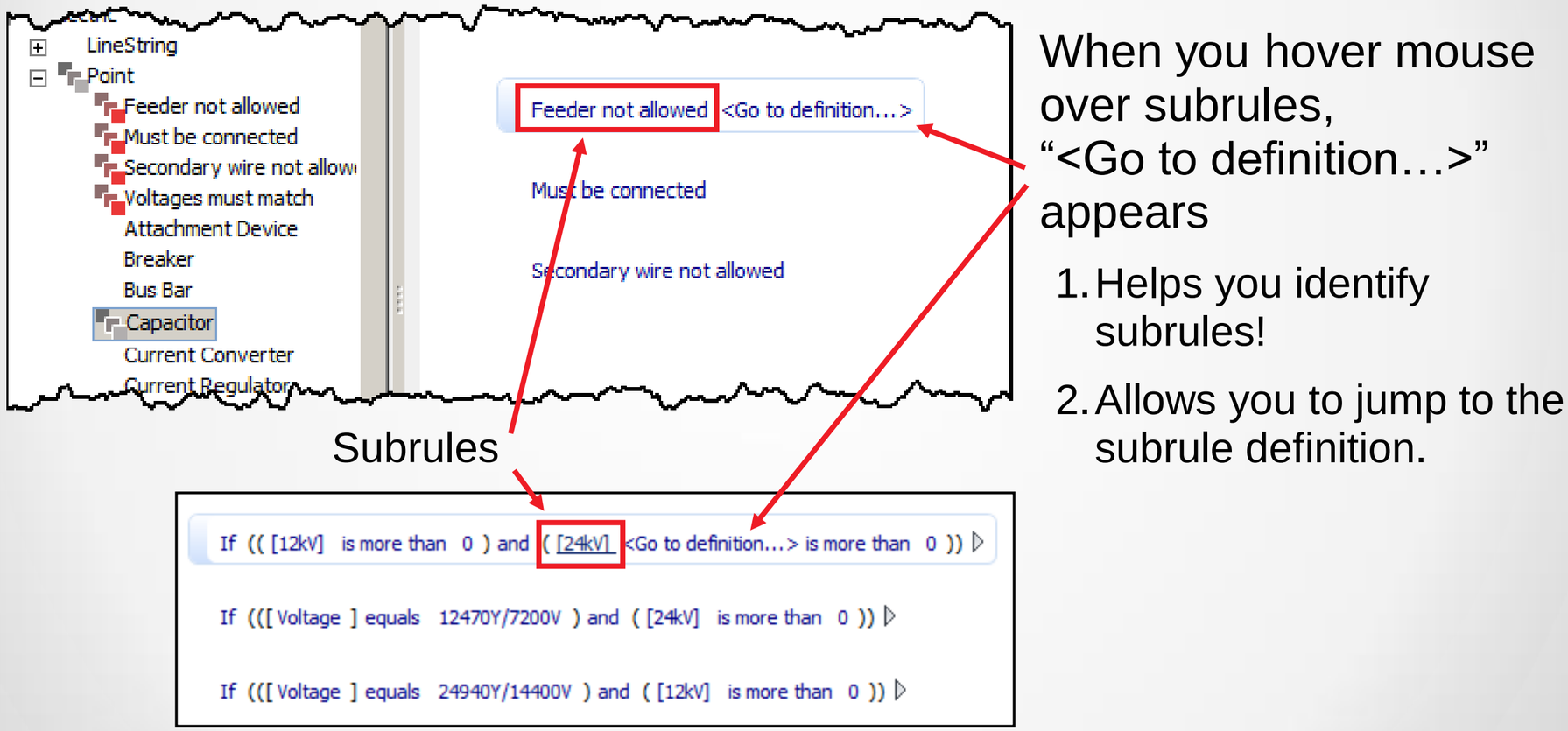
Behavior is driven by a combination of many factors



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

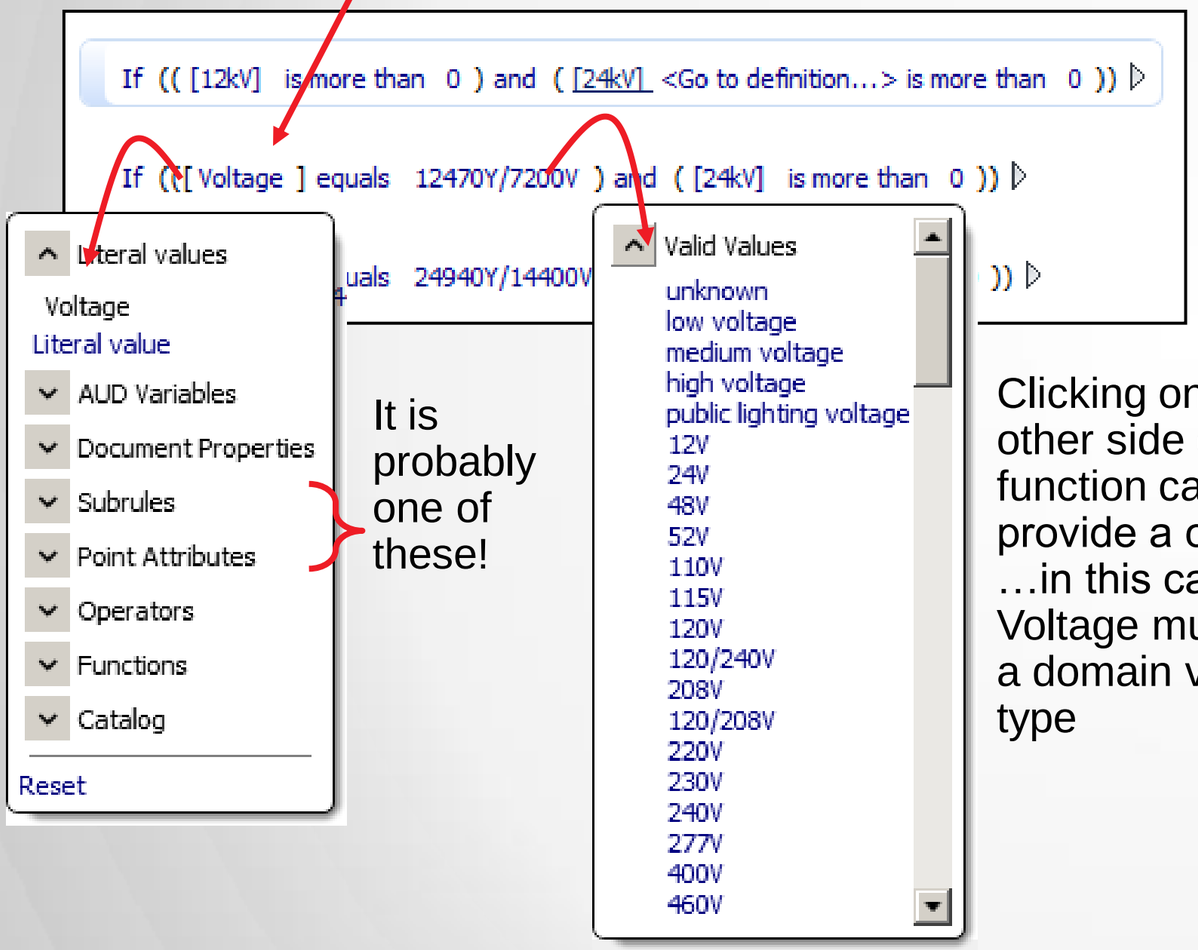


When you hover mouse over subrules, “<Go to definition…>” appears

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

Tip 2: Track down variables

Where is this “Voltage” coming from?



It is probably one of these!

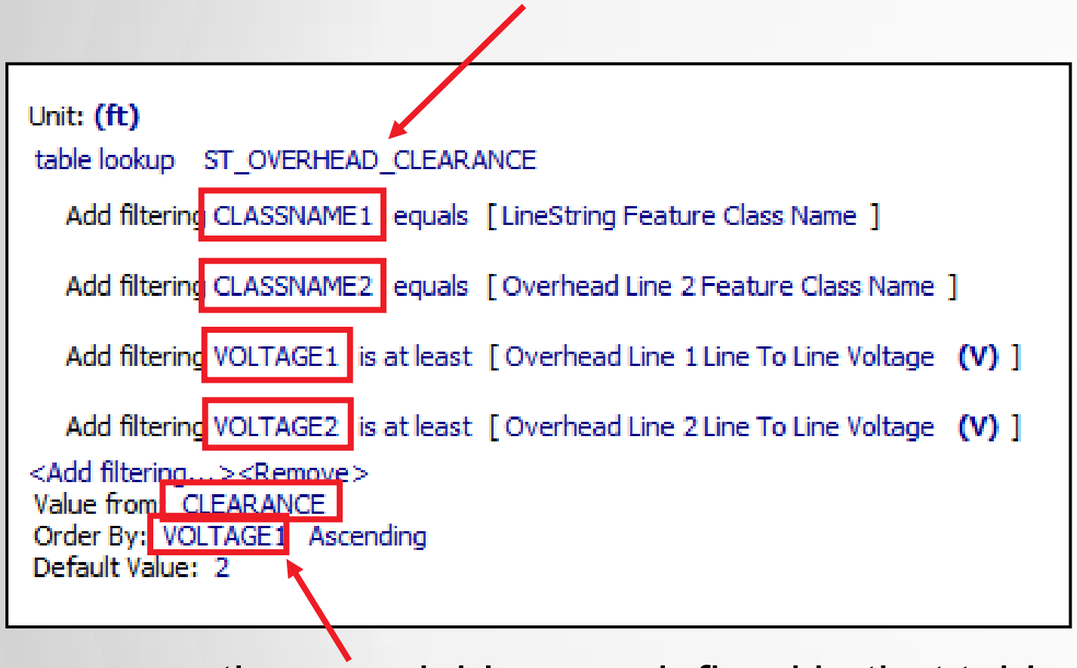
Clicking on the other side of the function can provide a clue…
…in this case Voltage must be a domain variable type

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

Tip 3: Watch out for “table lookup”

Note this is a “table lookup” clause…



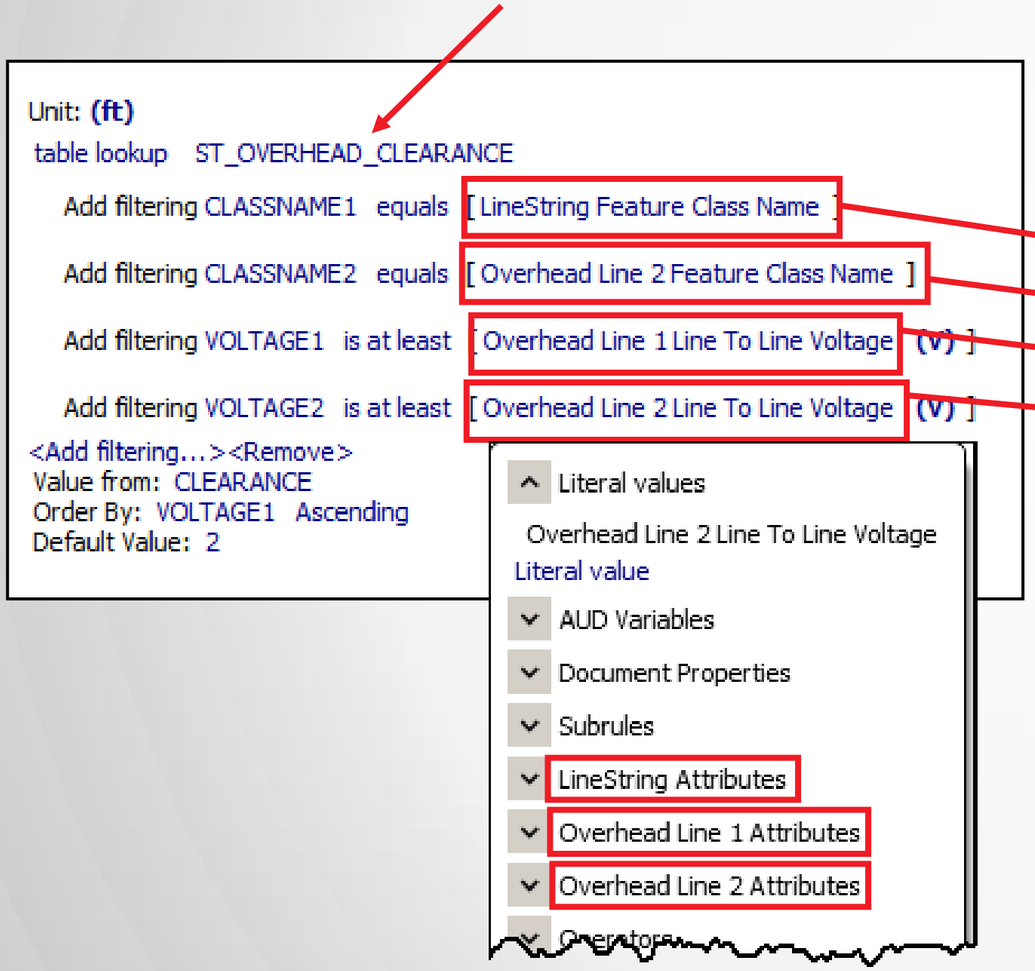
…so these variables are defined in that table.

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.

Tip 4: Watch for prefixes and multiple definitions

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



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

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

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 validation?
- …for material ordering?
- …for automatic sizing?
- …for automatic sizing?
- …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!