Variant Management

ACE23

Aras Variant Management

ACE 2023
Juro Mukai

Welcome to the ACE 2023 Training Session for Aras Variant Management

The objective of this session is to provide a 'practitioners view' of the Variant Management Application through hands-on exercises.

Starting with an interactive tour of the VM Application, we will then extend an existing variable configuration to enable new variant resolutions.

If time allows, we will build a new variable configuration from the ground up.

Let's get started…

Agenda

Part One
- Interactive Tour of the Aras Variant Management Application

Part Two:
- Extending Variability – Ebike conversion for Bicycle structure

Part Three:
- Building a Variable Configuration from Start to Resolution

Part One is a hands-on, click-along exploration of Aras Variant Management with the intent to familiarize with the functional design of the application 'first-hand'.

Part Two involves making some real modifications to the variability of the familiar Bicycle database, during which we will add an EBike conversion and new Features, Options, Rules and Usage to resolve new variants.

Part Three, if time allows, is an assignment for you to do largely on your own with the detailed information (and pre-added Parts to save time). You will need to understand the provided variability example and relate it to what you learned in Parts One and Two.

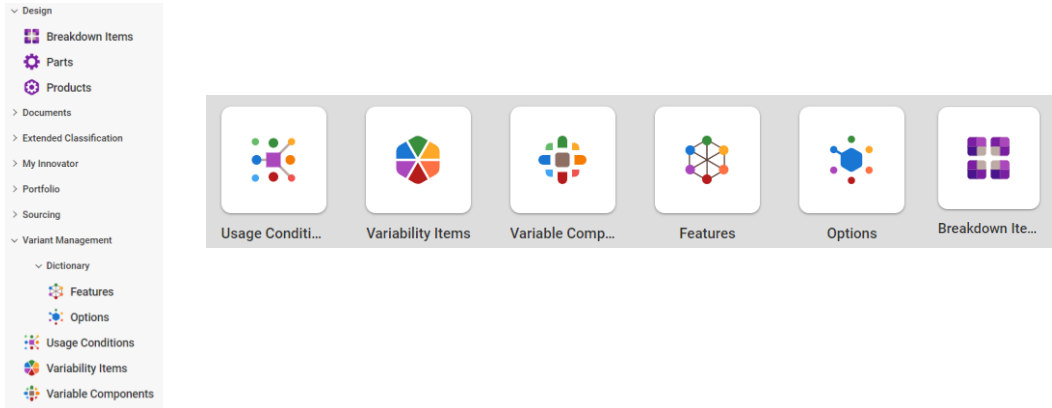# Aras Variant Management –
An Interactive Tour

As mentioned earlier, Part One is a hands-on click-along exploration of Aras Variant Management with the intent to familiarize with the functional design of the application 'first-hand'.
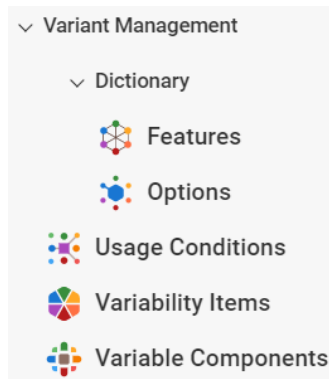
All aboard!

The Aras Variant Management TOC

- Once Variant Management is installed, these additional ItemTypes are available for members of the Variability Management Identity:
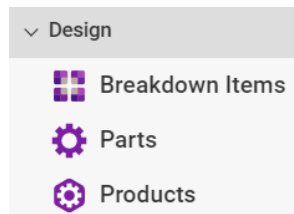
Try it:

- Navigate to Variant Management in the TOC. You will see the following Itemtypes:

- Now navigate to Design. Breakdown Items appear here:

Features and Options

- In Aras VM the meaningful characteristics of a variable configuration are represented as **Features;** and all possible values for them are represented as **Options.**

- **Features** are typically used together to collectively make all of their related **Options** available for use in defining variability.
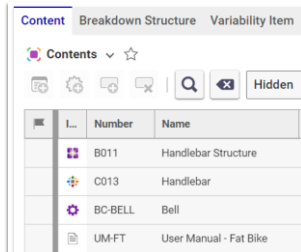
Try it:

- Navigate to *Dictionary->Features* in the TOC

    - Open **F013 Bicycle Type** which has 3 basic Options, then compare with **F005 Frame Size** which has a broader range of values.

    - A hierarchy is implied – some are constrained by others when Options are used to define variability.

    - For example, the Rule (covered in detail later) shown below combines Options from [Bicycle Type], [Bicycle Size] and [Frame Size] Features to restrict the size of X-Large Mountain Bike frames to 23 inches:

IF [Bicycle_Type] = [Mountain Bike] AND [Bicycle_Size] = [X-Large]
THEN [Frame_Size] = [23 inches]

⬛ Breakdown Items

- The **Breakdown Item,** its structure and variability represents a variable configuration in Aras VM. It is analogous to the 150% BOM concept.

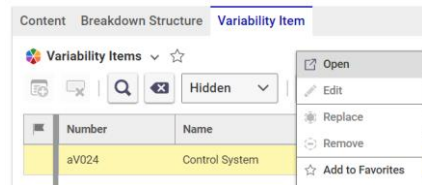- Relationship Tabs found on the Breakdown Item:

- **Content Tab**
- **Breakdown Structure**
- **Variability Item**.

- ▪ **Content** tab
    - ▪ Components applicable to a Breakdown Item, I.e., the 'scope' of a variable configuration. Includes Variable Components, other Breakdown Items (as a structure), or [Parts | Documents | Requirements].
- ▪ **Breakdown Structure** tab
    - ▪ A Tree Grid View of the entire variable configuration (a.k.a. the 150% BOM).
- ▪ **Variability Item** tab
    - ▪ Variability Items contain Rules and relevant Features used to constrain Option Selection during the Resolution of a variant

Try it:

- ▪ From the TOC, open **B012 Bicycle** Breakdown Item for viewing. This is the top-level node in the Bicycle variable configuration.

- ▪ Observe the contents of the three relationship tabs described above.

- ▪ Under **Breakdown Structure**, we see the entire variable configuration of the bicycle as a Tree Grid View

- ▪ Leave ⬛ **B012** Bicycle open for the next exercise…

Breakdown Item Structure - the Variable Configuration

- The Breakdown Item structure contains the permuted variable configuration containing all possible variants under one structure

- We resolve discreet variants from the Breakdown Item by selecting Option combinations
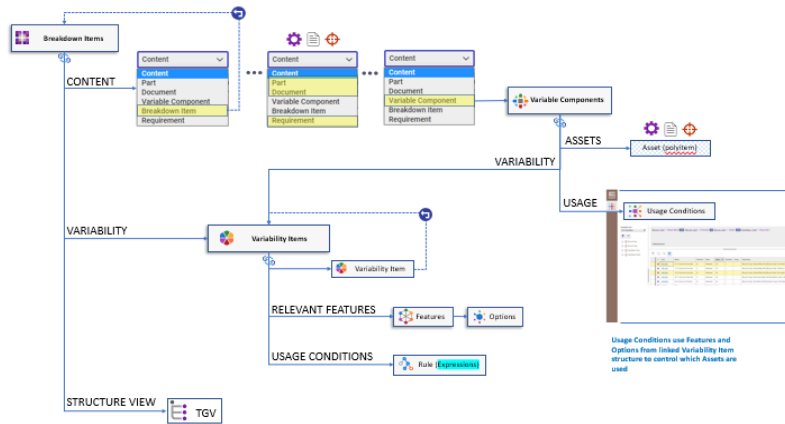
Try it:

- Expand the Breakdown Structure tab and note that it contains other
  Breakdown Items, Variable Components; which in turn contain a number
  of Parts

- This is a view of the entire variable configuration (a.k.a. the 150% BOM)

Variable Components

Variable Components contain all possible items (I.e., Part, Document, or Requirement) that can be resolved for any node in the variable structure.

- For example, **Variable Component** "C013 Handlebar" has 5 possible resolutions

- These specific (resolvable) items are called **Assets**

- **Usage Conditions** use TRUE/FALSE **Expressions** to determine which **Assets** are used in a variant result via

- The Sidebar includes a **Usage Condition Editor,** and the **Resolution** window where Option combinations are selected to resolve into specific variants

---

Try it:

- Open the **Variable Component** 'C013 Handlebar' using the RMB menu:

- Next open the **Usage Condition Editor** from the Sidebar:

I...	Item	Name	Expression
⚙	HB-9456	15.7" Aluminum Drop Bar	[Bicycle Type] = [Road Bike] AND ([Bicycle Size] = [X-Small] OR [Bicycle Si...
⚙	HB-1623	17.3" Aluminum Drop Bar	[Bicycle Type] = [Road Bike] AND ([Bicycle Size] = Medium OR [Bicycle Siz...
⚙	HB-2303	16.5" Aluminum Drop Bar	[Bicycle Type] = [Road Bike] AND [Bicycle Size] = [X-Large] AND [Handleba...
⚙	HB-9556	30.7" Aluminum Riser Bar	[Bicycle Type] = [Mountain Bike] AND [Handlebar Type] = [Riser Bar]
⚙	HB-8948	24.4" Aluminum Flat Bar	[Bicycle Type] = [Fat Bike] AND [Handlebar Type] = [Flat Bar]
⚙	BC-BELL	Bell	

Conditions of Usage

- Click on Usage rows to examine the **Expressions** that resolves a specific **Asset**:

| I...	Item	Name ↑	Revision	State	Expression
⚙	HB-9456	15.7" Aluminum Drop Bar	B	Released	[Bicycle Type] = [Road Bike] AND ([Bicycle Size] = [X-Small] OR [Bicycle Size] = Sm...

[Bicycle_Type] = [Road Bike] **AND** ([Bicycle_Size] = [X-Small] **OR** [Bicycle_Size] = Small) **AND** [Handlebar_Type] = [Drop Bar]

Q: Why does BC-BELL have no Usage Condition Expression?

■■ Resolving a Variant

- Resolution of a Variant from a variable configuration is done from the Breakdown Item using the '**Resolution**' window, accessed from the sidebar menu.

- A graphical interface allows Option selection (as constrained by Rules defined on the related Variability Item(s)
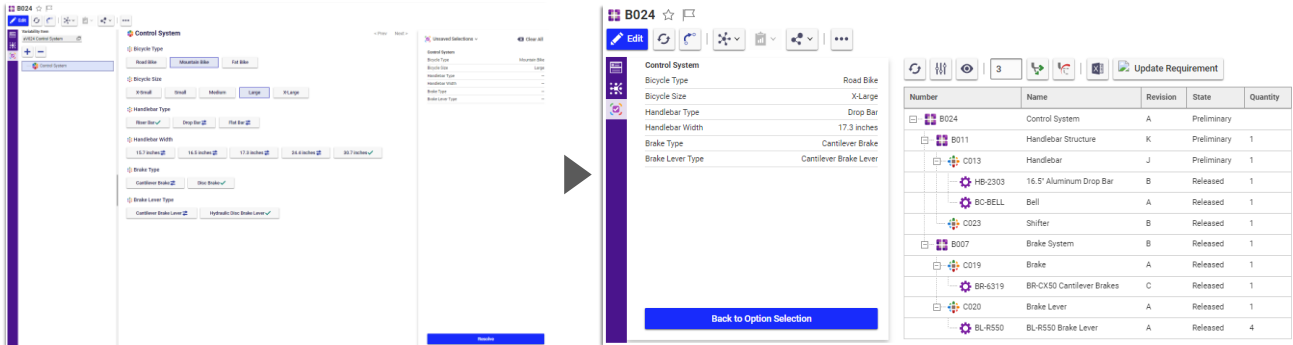

Try it:

- Return to ■■ **B024** and open the 'Resolution' window from the sidebar menu

- Experiment with Option selections and click 'Resolve' to see which specific Parts are resolved for each variant.
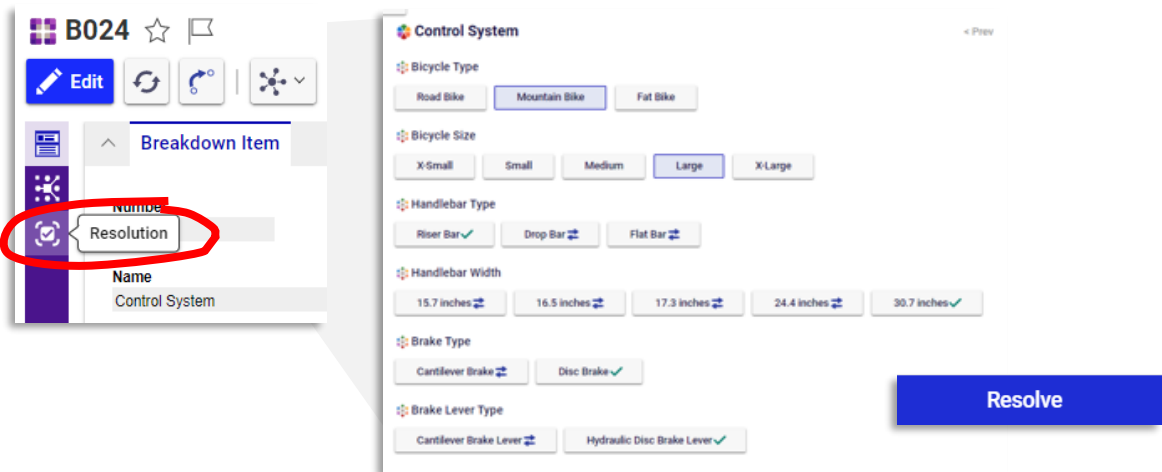
> Usage Conditions on each Variable Component determine which of multiple Assets gets resolved based on Option selections

Checkpoint 01 ✓

- ✓ So far, we've examined **Features** and **Options**

- ✓ Then we reviewed the "**Breakdown Item**" Structure (I.e., the "Variable Configuration", or the "150% BOM")

- ✓ We also explored the **Variable Component** and its **Usage Conditions** that decide which Part (Asset) gets resolved

- ✓ Finally, we selected from the allowable **Option** combinations to resolve a variant ("100% BOM") from the **Resolution** sidebar.
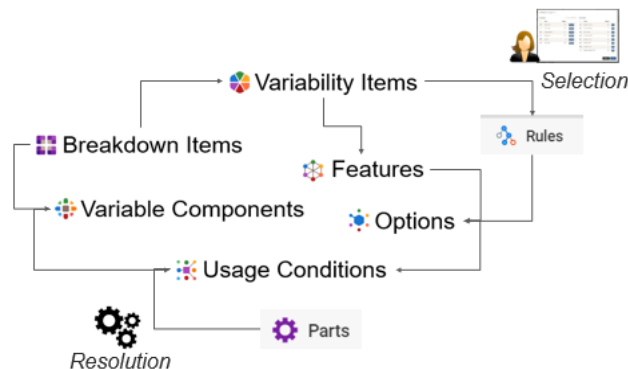
Questions:

1. What represents the characteristics of a product, system, or other variable configuration?

2. Which object contains all possible resolutions for a component in a variant?

3. How does Aras VM determine which Asset is resolved?

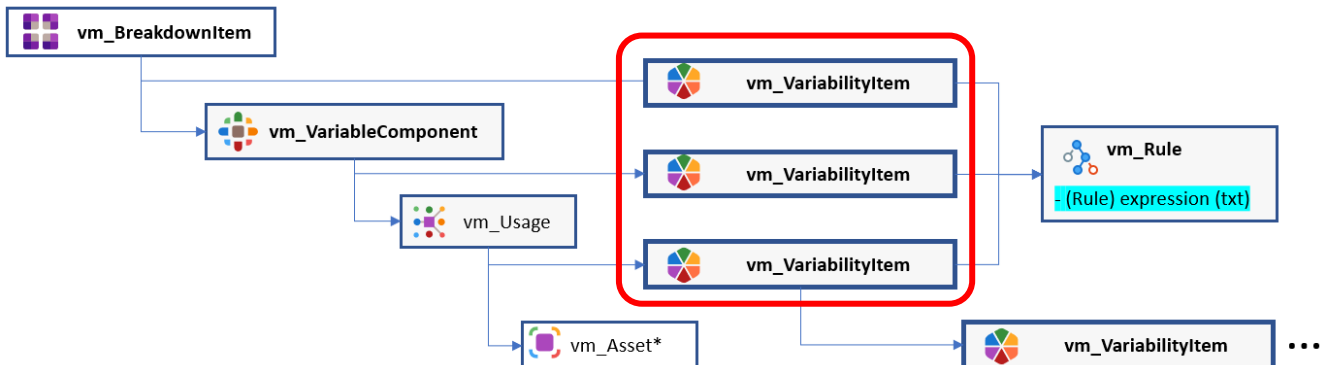4. What Itemtypes can be added to a Breakdown Item structure?

And how did Resolution know which Options were compatible with each other while making our selections?
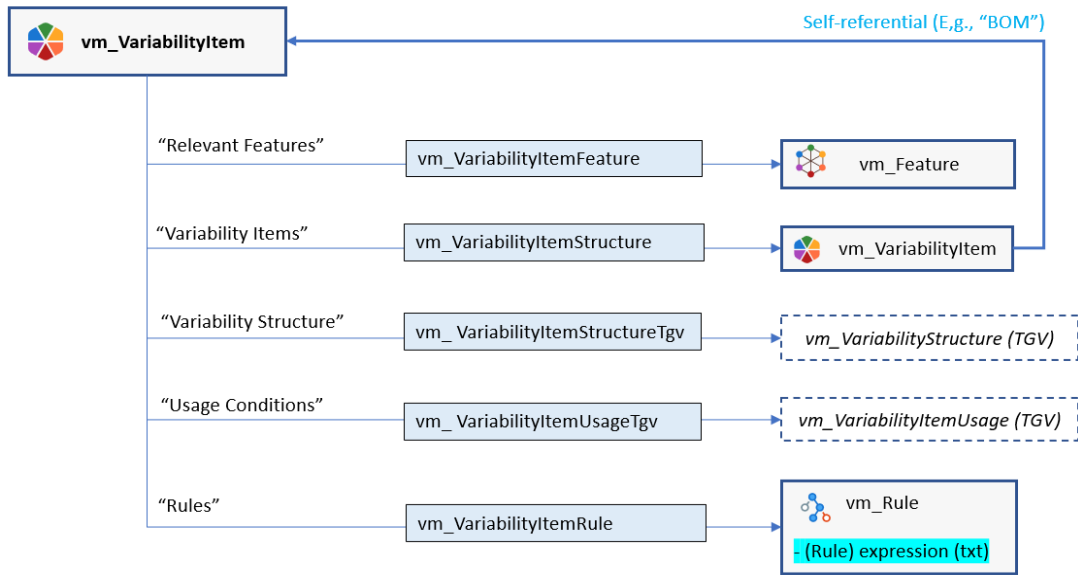
Variability Items
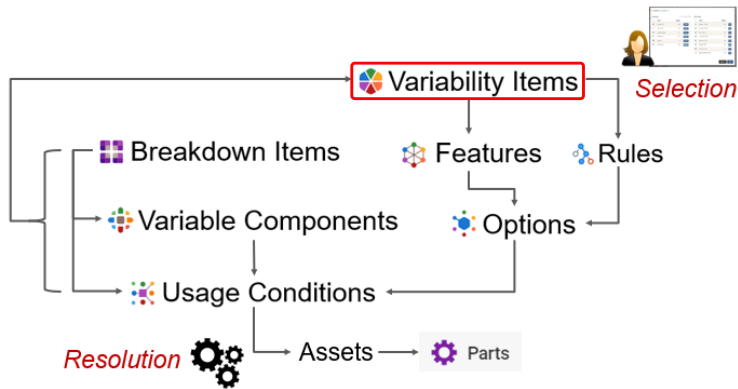
While resolving a Variant from ▦ **B024,** our Option selections were constrained to certain combinations. These constraints were defined as **Rules** on the related ❀ **Variability Item** (per the 'Variability Item' tab on the Breakdown Item).

- **Variability Items** include:
  - **Rules** (conditional expressions that constrain Option combinations)
  - And all Relevant **Features** whose **Options** are used in its Rules

- Variability Items are modular, reusable and versionable. They are referenced by Breakdown Items, Variable Components, Usage Items as well as other Variability Items in a BOM-like structure:
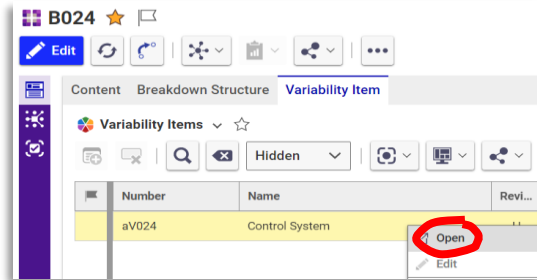
Variability Items

Self-referential (E,g., "BOM")

|		
"Relevant Features"	vm_VariabilityItemFeature	vm_Feature
"Variability Items"	vm_VariabilityItemStructure	vm_VariabilityItem
"Variability Structure"	vm_ VariabilityItemStructureTgv	vm_VariabilityStructure (TGV)
"Usage Conditions"	vm_ VariabilityItemUsageTgv	vm_VariabilityItemUsage (TGV)
"Rules"	vm_VariabilityItemRule	vm_Rule - (Rule) expression (txt)

Try It:

- From the 'Variability Item' tab on **B024** Breakdown Item use the RMB menu to open **aV024** Variability Item


- View the contents of the '**Relevant Features**' and '**Variability Structure**' tabs

- *We will define our own Feature scope(s) and Rules in later exercises*
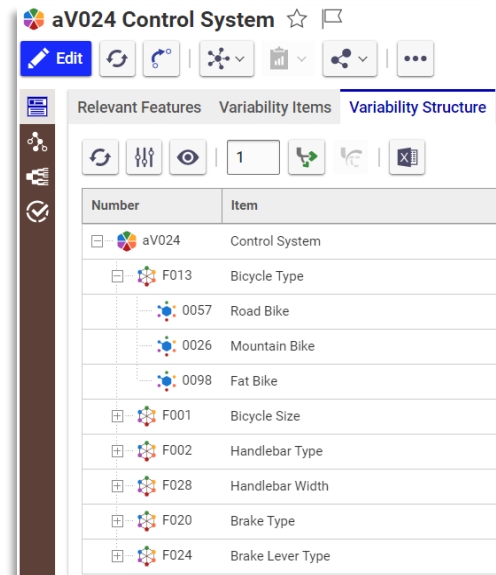
All Options under Relevant Features are available for use as operands in variability expressions (Rules, Usage Expressions)

❇️ Variability Item – ⚛️ Rules

- **Rules** control how Options can be selected and combined to implement variability.
- They are defined within a **Variability Item** and will apply variability to any object referencing it.
- For example: In the case where a Variability Item is used by (related to) a **Breakdown Item** (a.k.a. 150% BOM), the Rules defined within the related Variability Item control the selection of Options to resolve that Breakdown Structure to a specific variant (a.k.a. 100% BOM).
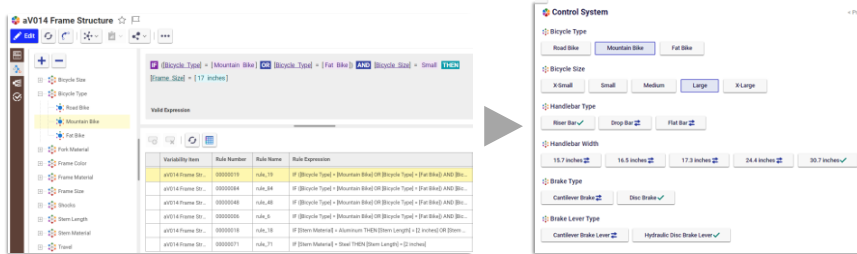
Rules contain expressions using IF/THEN <Feature>=<Option> syntax with support for AND/OR/NOT operators.

Try it:

- Use the 'Rule Editor' sidebar menu on ❇️ **aV024** Variability Item to view the Rules constraining our selections while resolving 🟦 **B024** in an earlier exercise.
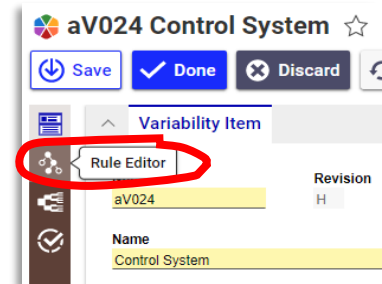
The **Table Editor** provides an intuitive, visual way to define Rules:

Try it:

- Open the Table Editor in the Rules Editor window

- Follow along with your instructor to view various Rule constraints in a table.

  - Review Principal vs. Constrained settings

  - Try 'build table from rules' action

- We will use these tools to create Rules shortly.
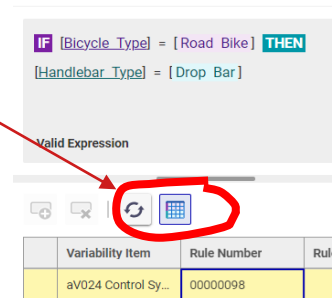
✿ Variability Item – ✅ Validation

- Variability Items allows us to preview the effect of the Rules we've defined using the **Validation** tool, accessed from the sidebar menu.

- We can test our Option selections in various combinations to validate the intended variability as implemented in Rules.
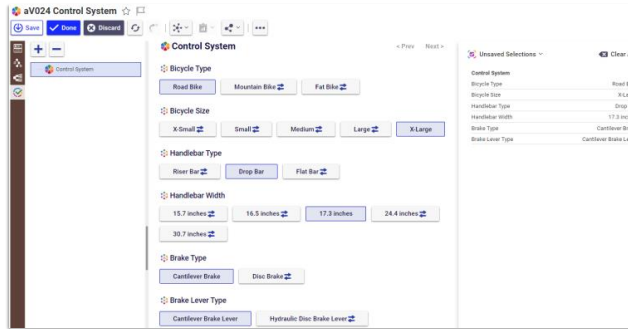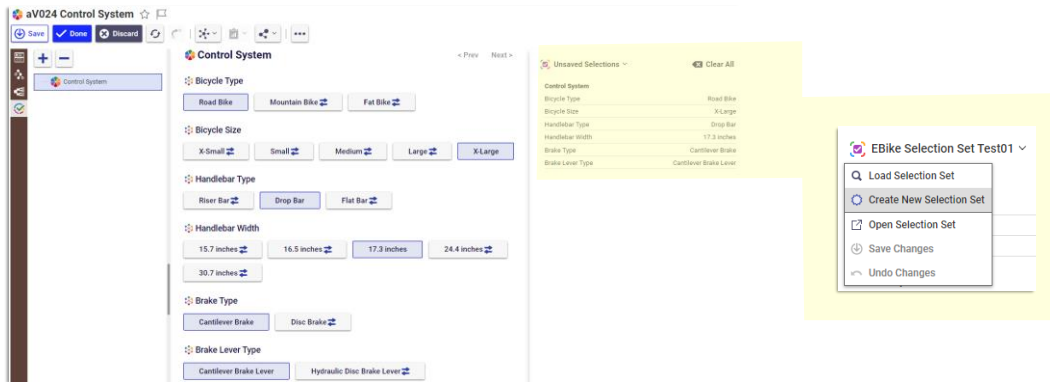
Try it:

- From the 'Validation' sidebar menu on ✿ **aV024** Variability Item open and make selections to preview the variability we encountered when resolving the Breakdown Item structure in ⊞ **B024.**

- Verify Rule behavior by experimentation in the **Validation** window
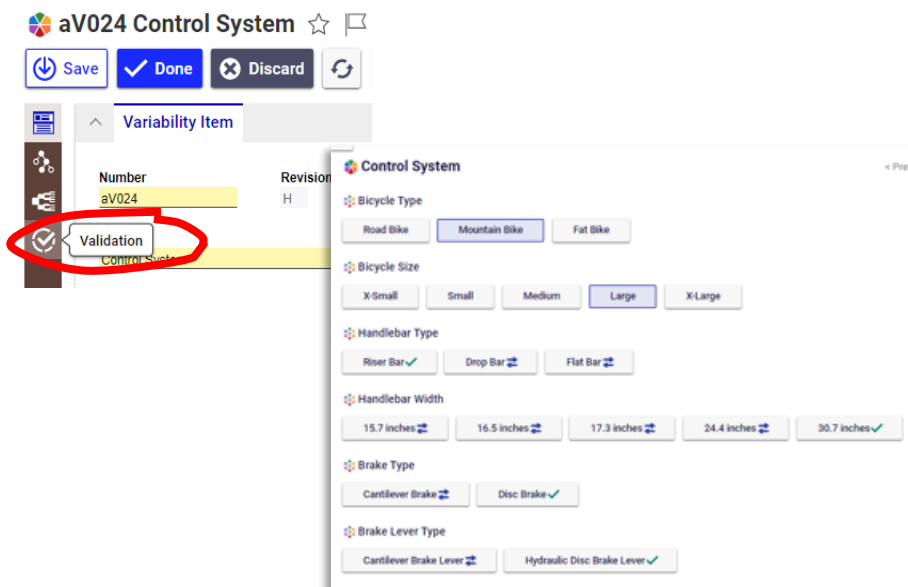
🌸 Variability Item – Saving/Loading a Selection Set

- We can save the Option selections made during Validation to re-load and validate again either in Variability validation –or- to resolve a Breakdown Structure that uses the same variability structure.

Try it:

- From the 'Validation' sidebar menu on 🌸 **aV024** Variability Item make selections to preview variability

- Save your selections from the Summary Pane (far right). Give the selection set a number and an easily recognized name like "Control Sys Config01"

- Now navigate via "Where Used" (or TOC) to the Breakdown Item 🟪 **B024** and open the **Resolution** sidebar window.

- Load the saved Selection Set and Resolve to see the Variant Asset results:

Number	Name	Revision
⊟ 🟪 B024	Control System	A
⊟ 🟪 B011	Handlebar Structure	K
⊟ C013	Handlebar	J
🔧 HB-8948	24.4" Aluminum Flat Bar	B
🔧 BC-BELL	Bell	A
C023	Shifter	B
⊟ 🟪 B007	Brake System	B
⊟ C019	Brake	A
🔧 BR-4069	SLX Hydraulic Disc Brake	B
⊟ C020	Brake Lever	A
🔧 BL-3757	SLX Hydraulic Disc Brake Lever	A

💡 We can reload a saved selection set without reselecting Options

Summary
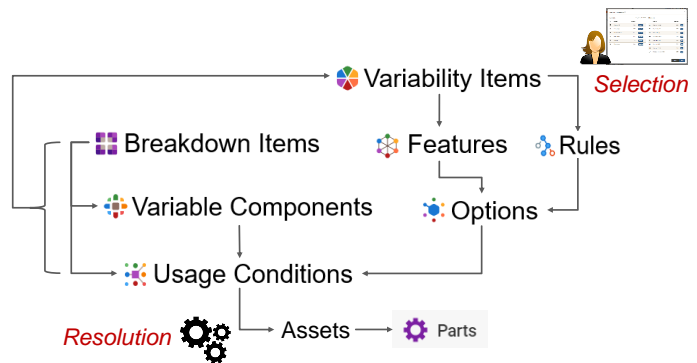
To Summarize:

- Variability Items are modular, reusable objects that combine Relevant Features and Rules to apply variability to referencing items like Breakdown Items, Variable Components, and Usage Conditions.

- Rules constrain the selection of Options in combination with other Options when defining and resolving variability.

- Variability Items provide specialized tools to simplify and enhance the definition of Rules: the Rule Editor & Table Editor, the Variant Matrix, and Validation tool.

Forward Progress!
You should now be able to navigate the Aras Variant Management application with an understanding of its functionality, Object structure.

Quiz:

1. Rules determine which Asset a Variable Component resolves to…
   [TRUE/FALSE]

2. Any Option can be used in a Rule Expression…
   [TRUE/FALSE]

3. Rules are Boolean expressions…
   [TRUE/FALSE]

4. I can save a set of Option selections to re-use on other VM objects later…
   [TRUE/FALSE]

5. I can create and edit Rules in the Variant Matrix …
   [TRUE/FALSE]

Quiz:

1. Rules determine which Asset a Variable Component resolves to…
   [TRUE/FALSE]

2. Any Option can be used in a Rule Expression…
   [TRUE/FALSE]

3. Rules are Boolean expressions…
   [TRUE/FALSE]

4. I can save a set of Option selections to re-use on other VM objects later…
   [TRUE/FALSE]

5. I can create and edit Rules in the Variant Matrix …
   [TRUE/FALSE]

Quiz Answers:

1. *False*, Variable Components determine which Asset (Part/Document/Requirement Item) resolves to a discreet variant via Usage Condition expressions.

2. *True*, provided the Option's parent Feature was added as a relevant Feature to the Variability Item.

3. *False*, Rules are actually Conditional Expressions in the IF/THEN= format with support for AND/NOT. Usage condition expressions are Boolean expressions.

4. *True*, see section 4.2.12.3 in the Variant Management 27 User Guide for details on using the Summary Pane in the Validation window.

5. *False*, Variant Matrix provides a tabular view of valid Option combinations per the Rules configured on a Variability Item – but use the Rules Editor to make any changes

In the next Unit, we'll extend the existing variable configuration through hands-on exercises, and resolve a new variant.
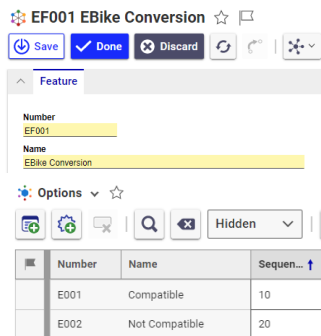
Extending Variability

Now that we've completed the interactive tour of the VM Application, lets make some modifications to the Bicycle variable configuration.

With the popularity of Electric Bikes (EBikes) we will need to add a conversion for the Fat Bike to 'go electric'.

Follow along with the instructor to add Features and Options, extend variability, and ultimately resolve new discreet variants with EBike conversions in the 100% BOM.

Adding an EBike Conversion Feature

- We will first create a new Feature that can be applied to the Bicycle variable configuration
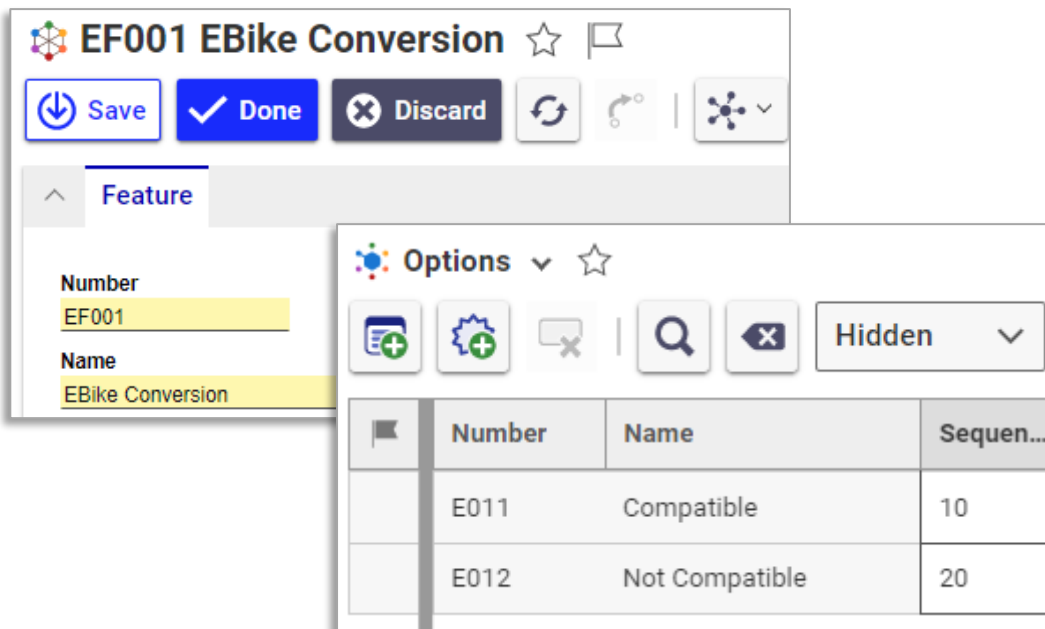- Then we will use this Feature to configure a variant that enables conversion to EBike

Try it:

- Navigate to *Dictionary->Features* in the TOC
    - Create a new Feature '**EF001 EBike Conversion**'
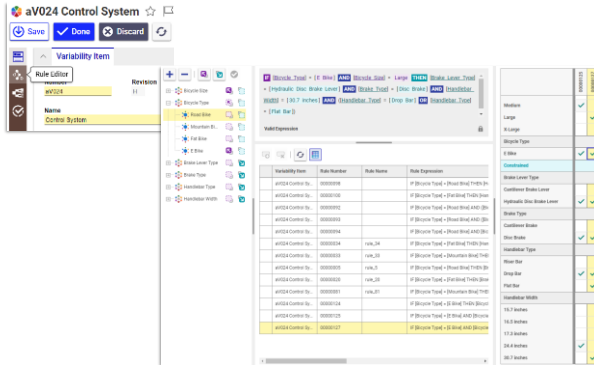   - Add Options as shown below, and save the Feature:

Add Rules to Constrain EBike Conversions

- Next, we will add a new Rule to allow EBike conversions for a specific Bicycle Type (Fat Bikes)

Try it:

Work along with the Instructor to add Rules to constrain the EBike conversions:

1) Locate the ⚙ **aV012 Bicycle** Variability Item from the TOC (Or navigate to Variability Item tab from Bicycle Breakdown Item). Open it for Editing.

2) Add the EF001 EBike Conversion Feature to the 'Relevant Features' tab

3) Follow along to configure these 2 Rules in the **Rule Editor** sidebar window and Validate once added:

First rule:

IF [Bicycle_Type] = [Fat Bike] THEN [EBike_Conversion] = Compatible
AND ([Bicycle_Size] = Medium OR [Bicycle_Size] = Large) ← parens

Second rule:

IF [Bicycle_Type] = [Road Bike] OR [Bicycle_Type] = [Mountain Bike]
THEN [EBike_Conversion] = [Not Compatible]

Validate rules:

⚙ Bicycle Type

Road Bike ⇄   Mountain Bike ⇄   Fat Bike

⚙ Bicycle Size

X-Small ⇄   Small ⇄   Medium   Large   X-Large ⇄

⚙ EBike Conversion

Compatible   Not Compatible ⇄

Add Additional Features and Options for E-Bikes

- Next, we will extend Bicycle variability by adding new Rules for EBikes using existing Features and Options that apply to the new Bicycle Type

  - Motor
    - Hub Drive
    - Gear Drive
  - Controller
  - Throttle
    - Thumb
    - Twist Grip
  - Battery Pack
    - 36v 14Ah
    - 52v 20Ah

Try it:

Again, work along with the Instructor to create new Features and Options:

1) Feature "EF002 EBike Motor"
   - Option "E002 Hub Drive"
   - Option "E003 Gear Drive

2) Feature "EF003 EBike Controller"
   - Option "E004 Controller"

3) Feature "EF004 EBike Throttle"
   - Option "E005 Thumb"
   - Option "E006 Twist Grip"

4) Feature "EF005 EBike Battery Pack"
   - Option "E008 36v 14Ah"
   - Option "E009 52v 20Ah"

We will use these new Features and Options to extend variability for the new Bicycle Type.

> In the next Unit, we'll extend the existing variable configuration through hands-on exercises, and resolve a new variant.

Configure E-Bike Variability Items 🎨

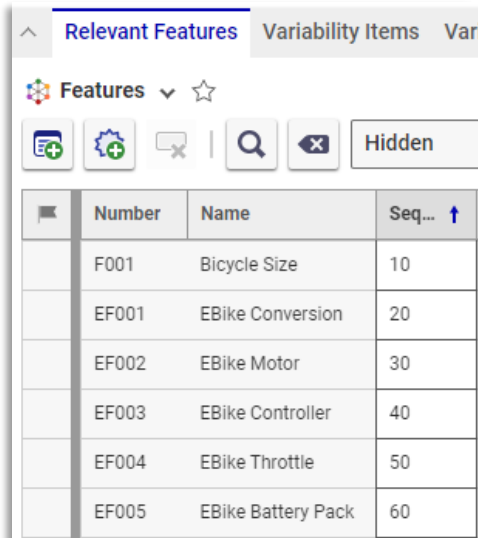**Define new Rules using the EBike Features and Options**

We will now apply the new Options we've created in Rules to define variability of EBikes. Steps include:

- Create a New Variability Item
- Add Relevant Features
- Define Rules
- Validate the new Rule set
- Apply the EBike Variability Item to the parent Variability (Bicycle)
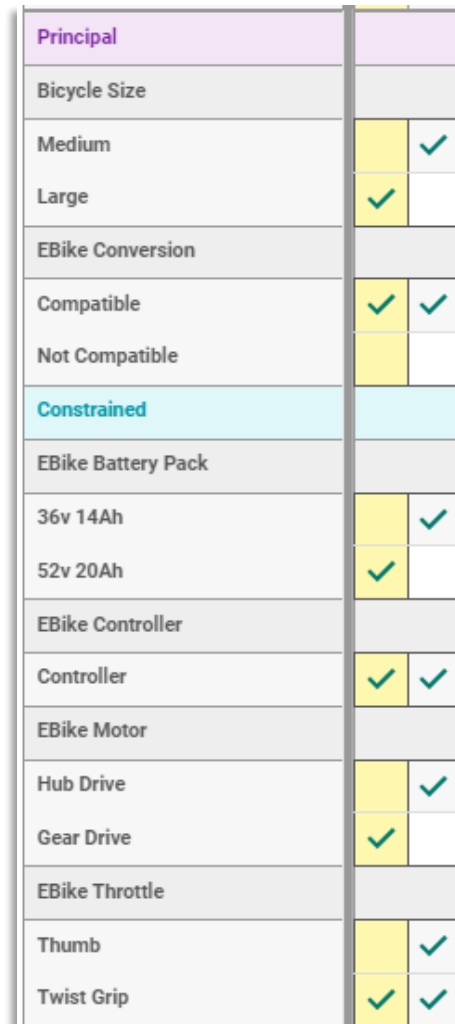- Validate

---

Try it:

1) Create a new Variability Item named;
   "eV001 EBike Variability"

2) Add Relevant Features as shown below:

Number	Name	Seq... ↑
F001	Bicycle Size	10
EF001	EBike Conversion	20
EF002	EBike Motor	30
EF003	EBike Controller	40
EF004	EBike Throttle	50
EF005	EBike Battery Pack	60

Relevant Features | Variability Items | Var
Features | Hidden

3) Next, work with your instructor to configure Rules per the **Table Rule Editor** screenshot shown at right:

4) Validate your Rules

Principal		
Bicycle Size		
Medium		✓
Large	✓	
EBike Conversion		
Compatible	✓	✓
Not Compatible	✓	
Constrained		
EBike Battery Pack		
36v 14Ah		✓
52v 20Ah	✓	
EBike Controller		
Controller	✓	✓
EBike Motor		
Hub Drive		✓
Gear Drive	✓	
EBike Throttle		
Thumb		✓
Twist Grip	✓	✓

Include EBike Variability on Bicycle

With EBIke Variability now applied and validated, let's incorporate it into the top-level Bicycle Variability Item

Try it:

1) With **"aV012 Bicycle"** Variability Item in edit mode, add **"eV001 EBike Variability"** to its Variability Items tab.

2) Save, and examine Rules. You should see that the Rules and Feature scope from **eV001 EBike Variability** Item have been incorporated in **aV012 Bicycle**.

3) Open Validation from the sidebar menu of **"aV012 Bicycle"** Variability Item and verify the Rule logic in the context of the Bicycle parent
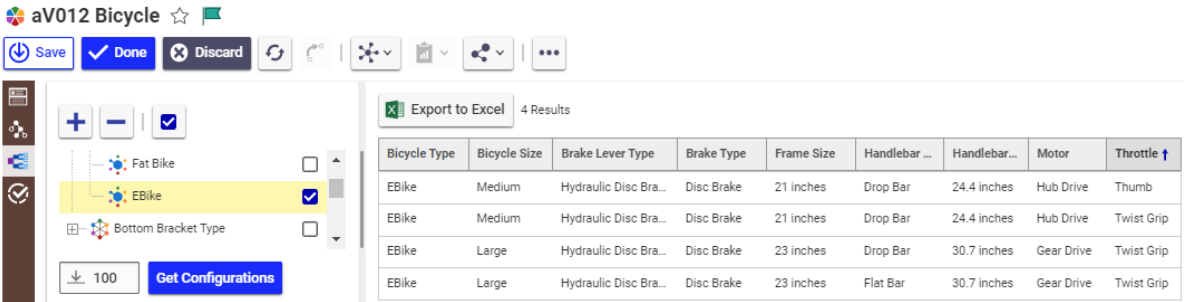
Variability Item – Variant Matrix

Another useful variability tool is the Variant Matrix view, accessible from the Variability Item Sidebar. It allows you to selectively generate a grid (matrix) of valid Option combinations in the right-side pane for a given selection of Feaure/Options in the left-side pane:

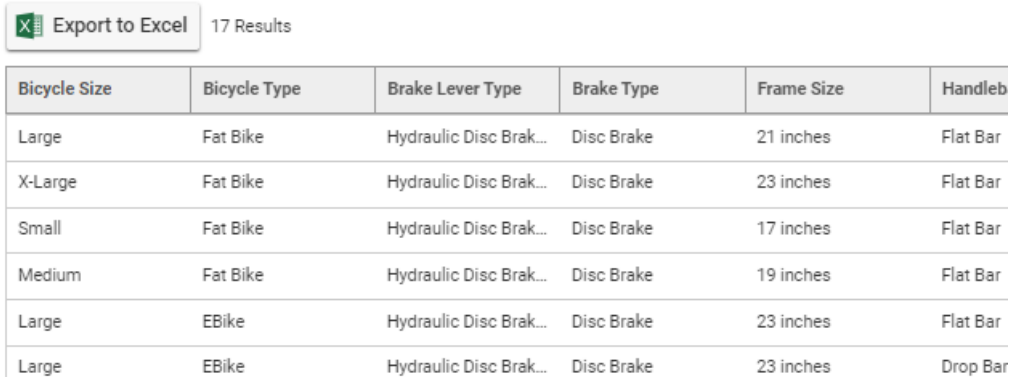## Try it:

1) From the Variability Item Sidebar menu, open the **Variant Matrix** window.

2) Make Feature/Option selections as desired, then hit **Get Configurations**

💡 You can optionally export the permuted matrix to Excel.

Breakdown Structure to Resolve EBike Variants
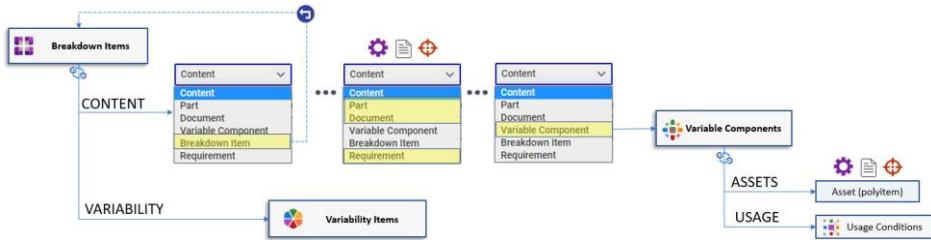
We have now extended Bicycle Variability to include an EBike conversion with corresponding Features (Motor, Throttle, Battery, etc.) and their Option choices. We now need to define the Breakdown Item structure that will resolve into actual Part items in the variant BOM.

Content	Breakdown Structure	Variability Item		

Number	Name	Revision	Sequence
▦ EB001	EBike Component Structure	A	
⬡ EC002	EBike Motor	A	10
⚙ EP-001	Bafang Gear Hub Motor	A	10
⚙ EP-002	DNYSYS Gear Motor	A	20
⬡ EC003	EBike Controller	A	20
⚙ EP-007	EBike System Controller	A	10
⬡ EC004	EBike Throttle	A	30
⚙ EP-003	DPX Thumb Throttle	A	10
⚙ EP-004	TNT Twist Grip Throttle	A	20
⚙ EP-004	TNT Twist Grip Throttle	A	30
⬡ EC005	EBike Battery Pack	A	40
⚙ EP-005	Standard EBike Battery	A	10
⚙ EP-006	Extended Range Battery	A	20

⚏ Breakdown Structure Updates for EBikes
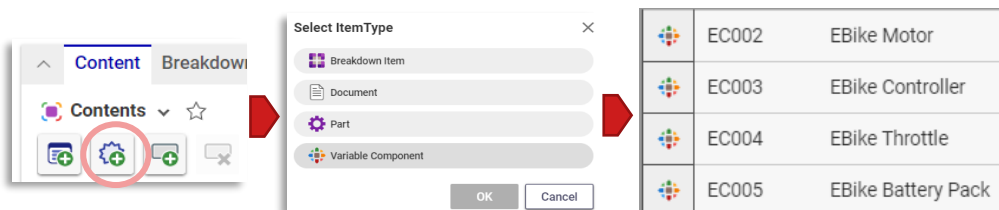
- In order to resolve discreet variant configurations for EBikes we need to define a Breakdown Item structure to add to the parent Bicycle structure.

- Recall that Breakdown Items use ⊕**Variable Components** to contain multiple possible Assets (Parts, Documents, Requirements, etc.) for each Node in the Breakdown Structure.

- We will now add Variable Components to our new Breakdown Item, then add Part Assets to the Component nodes to complete the 150% BOM

Try it:

- Create a new Breakdown Item with the number **EB001** named '**EBike Components'**. In the 'Content' tab, use the 'New Content' button to add four new Variable Components:

⊕	EC002	EBike Motor
⊕	EC003	EBike Controller
⊕	EC004	EBike Throttle
⊕	EC005	EBike Battery Pack

- Part Assets have been provided to save time. Add them to the Variable Components accordingly:

| EBike Motor |
| EBike Controller |
| EBike Throttle |
| EBike Battery Pack |

| ⚙ | EP-001 | Bafang Gear Hub Motor |
| ⚙ | EP-002 | DNYSYS Gear Motor |

| ⚙ | EP-007 | EBike System Controller |

| ⚙ | EP-003 | DPX Thumb Throttle |
| ⚙ | EP-004 | TNT Twist Grip Throttle |

| ⚙ | EP-005 | Standard EBike Battery |
| ⚙ | EP-006 | Extended Range Battery |
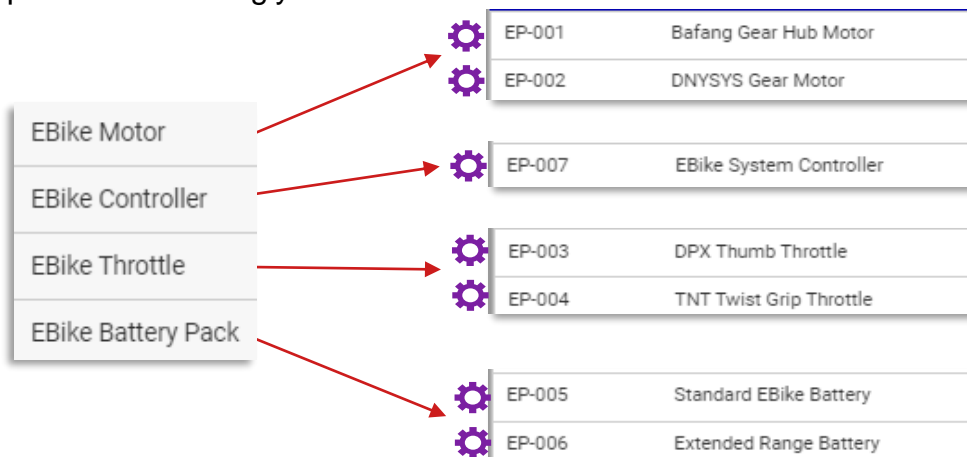
Add Usage Conditions for EBike Assets

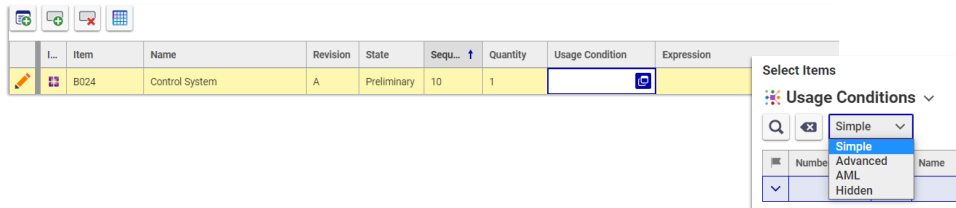|			
EB001	EBike Component Struc...		
EC002	EBike Motor		
EP-001	Bafang Gear Hub Motor	[Bicycle Size] = Medium AND [EBike Motor] = [Hub Drive]	
EP-002	DNSYS Gear Motor	[EBike Motor] = [Gear Drive] AND [Bicycle Size] = Large	
EC003	EBike Controller		
EP-007	EBike System Controller		
EC004	EBike Throttle		
EP-003	DPX Thumb Throttle	[EBike Conversion] = Compatible AND [Bicycle Size] = Medium AND [EBike Throttle] = Thumb	
EP-004	TNT Twist Grip Throttle	[Bicycle Size] = Large AND [EBike Conversion] = Compatible AND [EBike Throttle] = [Twist Grip]	
EP-004	TNT Twist Grip Throttle	[Bicycle Size] = Medium AND [EBike Conversion] = Compatible AND [EBike Throttle] = [Twist Grip]	
EC005	EBike Battery Pack		
EP-005	Standard EBike Battery	[EBike Conversion] = Compatible AND [Bicycle Size] = Medium	
EP-006	Extended Range Battery	[EBike Conversion] = Compatible AND [Bicycle Size] = Large	

aras

Try it:

1) Open the Usage Condition Editor for each Variable Component.

2) Select **eV001 EBike Variability** here:

   Variability Item
   eV001 EBike Variability

3) Select the Asset and enter the text from the right-column below into the '**Expression**' column. (The Instructor will show you how to use the Table expression editor as well)

4) Test Resolution of **EB001** EBike Breakdown Item
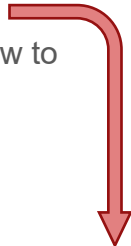
Part	Name	Expression
EP-001	Bafang Gear Hub Motor	[Bicycle Size] = Medium AND [EBike Motor] = [Hub Drive]
EP-002	DNSYS Gear Motor	[EBike Motor] = [Gear Drive] AND [Bicycle Size] = Large
EP-003	DPX Thumb Throttle	[EBike Conversion] = Compatible AND [Bicycle Size] = Medium AND [EBike Throttle] = Thumb
EP-004	TNT Twist Grip Throttle	[Bicycle Size] = Large AND [EBike Conversion] = Compatible AND [EBike Throttle] = [Twist Grip]
EP-005	Standard EBike Battery	[EBike Conversion] = Compatible AND [Bicycle Size] = Medium
EP-006	Extended Range Battery	[EBike Conversion] = Compatible AND [Bicycle Size] = Large
EP-007	EBike System Controller	-

Reusable Usage Conditions

- Rather than hardcoding a Usage Expression into a specific Variable Component's Usage Sidebar pane, we can apply the same Usage Expression for multiple Variable Components -> Asset and/or Breakdown Item -> Component Usage(s)

- Steps:
  - Create an independent **Usage Condition** Item instance.
  - A column for selection of existing Usage Conditions objects is provided on the Usage Condition Window grid (next to Expressions, which are hard-coded)
  - Use search criteria to find and re-use the modular Usage Condition objects

> We have not used this new feature in these exercises - but it is well documented in section **5.2** in the **Aras Variant Management 27 User Guide**

Add EBike Breakdown Item to Bicycle Parent

Now that EBike Variability is defined, we need to add it to the parent Bicycle structure to allow resolution of variants with the new Options.

Steps:

- Add **EB001** EBike Component Structure to Content Tab on **B012 Bicycle**
- Set Usage Conditions (Sidebar menu) on **EB001** to limit Usage of EBike Components when:

  [Bicycle_Type] = [Fat Bike] AND [EBike_Conversion] = Compatible

This instructs the Resolution engine to omit EB001 EBike Components when the variant has not been converted to electric.

Try it:

1) Edit **B012 Bicycle** Breakdown Item and add **EB001 EBike Components** to the Contents Tab

2) From the **Usage Conditions** Sidebar menu, open and assign the Expression below:

[Bicycle_Type] = [Fat Bike] **AND** [EBike_Conversion] = Compatible

Valid Expression

I...	Item	Name	Expression ↓
▦	EB001	EBike Component Structure	[Bicycle Type] = [Fat Bike] AND [EBike Conversion] = Compatible

This assures that EBike Components will only be included in resolution results when a bicycle is converted.

💡 Now you can resolve discreet variants from the parent Bicycle Breakdown Structure with support for EBike Conversions!

Resolve EBike Variants

From the top-level B012 Bicycle Breakdown Structure, open the Resolution Sidebar window and try configuring variants with/without EBike conversions.

- If you saved a Selection Set in the past during validation etc, you can re-load it here.
- Experiment with various Selection combinations.

Kudos! This concludes the instructor-led portion of this training. In the next section you will build a variable configuration from start to finish on your own with the Instructor's assistance as needed.