

Configuring Effectivity Management

Effectivity Management of Part BOM structures is available in the Product Engineering (PE) application. This enables expressing Effectivity conditions of Parts within the BOM and resolving BOM structures against these conditions. As an Administrator, you can configure and customize some aspects of Effectivity Management in the PE application, such as:

- Creating additional Effectivity Variables and Effectivity Scopes.
- Customizing the Effectivity Scope Builder Method for custom business logic.
- Configuring Query Definitions for structure resolution logic.
- Configuring Tree Grid Views to display Item and Relationship properties.
- Configuring Permissions that determine who can manage Effectivity information.

Important

The **effs_** name prefix is used for the ItemTypes that belong to Effectivity Management.

Overview of the Effectivity Services Data Model

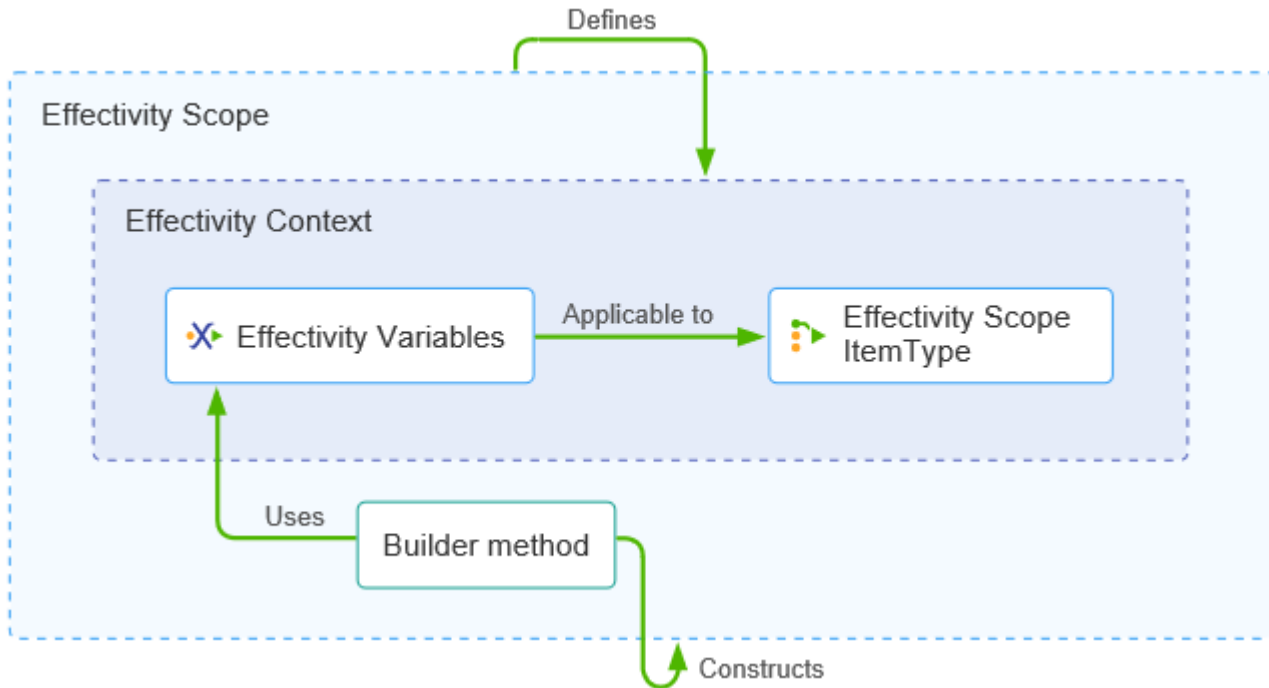
This section provides a high-level overview of the Effectivity Services based data model concept as implemented in the PE application. For more information on Effectivity Services, refer to the *Aras Innovator – Effectivity Services Programmer’s Guide*.

An Effectivity Scope defines a context for Resolution through Effectivity Criteria, which should be applied to particular effective items. An Effectivity Scope consists of the following items:

- Effectivity Variables, which are variables influencing Effectivity decisions.
- Effectivity Scope ItemType, which determines Items to be resolved by Effectivity.

A Builder Method uses Effectivity Variables to construct a Scope object in the runtime, used by the Effectivity Resolution Engine to perform Resolution of effective items.





Overview of Effectivity Services, TGV, and QD Interactions

The user interacts with the Tree Grid View (TGV) to provide the Effectivity Criteria for structure resolution.

TGV requests necessary data from the associated Query Definition (QD). The Effectivity Criteria is passed to this QD via QD parameters.

QD requests Effectivity Expression items with provided Effectivity Criteria.

The Effectivity Resolution Engine resolves Effectivity Expression items against provided Effectivity Criteria and returns the result.

The results are displayed to the user using the TGV.



Managing Effectivity Variables

Many variables influence Effectivity decisions. They depend on the industry, product, and business processes among others. They are used in:

- Effectivity Expressions to set Effectivity conditions within a structure.
- Effectivity Criteria for structure resolution.

Some of the most commonly used Effectivity Variables are provided out of the box:

- Model
- Unit
- Date

The **Aras Part BOM Scope** Effectivity Scope uses these Effectivity Variables.

Based on business needs, additional Effectivity Variables can be defined, as described in the following section.

To access the Effectivity Variables, select **Administration Effectivity Services Effectivity Variables** from the TOC.

Creating Effectivity Variables

Use the standard Item creation procedure to create a new Effectivity Variable. The following fields are required:

- **Name**
- **Type**: The out-of-the-box Builder Method supports the following Effectivity Variable types:
 - **Date**
 - **Integer**
 - **Item**
 - **List**
 - **String**
- If the specified **Type** is **List** or **Item**, provide an accompanying value in the **List** or **ItemType** field.

Important

When creating an Effectivity Variable with type "Item", the ItemType must be added as a Poly Source of the ScopeCachedependency ItemType.



A List used for an Effectivity Variable must have both a Value and a Label for each entry. The Builder Method uses both Value and Label. The Label is presented to a user when working with effectivity.

Editing Effectivity Variables

Important

Before changing any of the required fields of an Effectivity Variable, make sure that the Effectivity Variable is not used in any Effectivity Expressions.

Use the standard Item editing procedure to edit an Effectivity Variable.

Deleting Effectivity Variables

Important

Before deleting an Effectivity Variable, make sure that the Effectivity Variable is not used in any Effectivity Expressions.

Use the following procedure to delete an Effectivity Variable:

1. Remove it from its associated Effectivity Scopes.
2. Use the standard Item deletion procedure to delete the Effectivity Variable.

Managing Model Items for the Model Effectivity Variable

The **Model** Effectivity Variable uses Items of the **Model (effs_model)** ItemType. There are no Model Items out of the box. If your business uses the Model as an Effectivity Variable, you need to create Model Items.

To access the Model Items, go to **Administration Effectivity Services Models** in the TOC.

Creating a Model Item

Use the standard Item creation procedure to create a new Model Item. The **Name** field is required and unique.

Important



If a Model has a Label, then the Label is presented to a user when working with effectivity. Otherwise, the Name is presented.

Editing a Model Item

Important

Before changing the Name or Label of a Model Item, make sure that the Model Item is not used in any Effectivity Expressions.

Use the standard Item editing procedure to edit a Model Item.

Deleting a Model Item

Important

Before deleting a Model Item, make sure that it is not used in any Effectivity Expressions.

Use the standard Item deleting procedure to delete a Model Item.

Managing Effectivity Scopes

An **Effectivity Scope** is an Item of the **effs_scope** ItemType that defines the context for setting Effectivity conditions and resolving structures against given Effectivity Criteria.

To access the Effectivity Scopes, select **Administration Effectivity Services Effectivity Scope** from the TOC.

The PE application provides the **Aras Part BOM Scope** Effectivity Scope. The **effs_scopeObjectBuilderMethod** Builder Method constructs an object of this Scope.

The **Aras Part BOM Scope** Effectivity Scope uses the **Part BOM** ItemType and the **Model**, **Unit**, and **Date** Effectivity Variables. You can either update this Scope or create a new one to specify a set of Effectivity Variables that are relevant to your business.



Aras Part BOM ...

Aras Part BOM Scope

Edit

Effectivity Scope

Name

Aras Part BOM Scope

Builder Method

[effs_scopeObjectBuilderMethod](#)

Effectivity Scope ItemType

Effectivity Variables

Effectivity Variables

Hidden

Name	Type	List [...]	ItemType [...]
Unit	Integer		
Date	Date		
Model	Item		effs_model

Builder Method

A Builder Method is an Item of the **Method** ItemType, which constructs the Scope object using business data and logic.

In the PE application, **effs_scopeObjectBuilderMethod** is the default Builder Method that works with the Effectivity Scope ItemType and Effectivity Variables.



Methods ▼ ☆						
Search Clear Simple ▼ Current ▼ ▼						
	Name	Method T...	Core	Ver	execution_allowed_to [...]	Comments
▼	effs_scopeObjectBuilderMethod	▼			...	
	effs_scopeObjectBuilderMethod	CSharp	☐	1	Administrators	

Managing Effectivity Variables in Effectivity Scopes

There are no restrictions on the number or types of Effectivity Variables in an Effectivity Scope. You can either update the **Aras Part BOM Scope** Effectivity Scope or create a new Effectivity Scope to specify a set of Effectivity Variables that are relevant to your business.

Attaching Effectivity Variables to Effectivity Scopes

Use the standard procedure for attaching Items on the **Effectivity Variables** accordion tab of *an existing Effectivity Scope* Item view.

^ Effectivity Scope ItemType Effectivity Variables ▼ ☆ Hidden ▼ ▼ ▼ ▼				
	Name	Type	List [...]	ItemType [...]
	Unit	Integer		
	Date	Date		
	Model	Item		effs_model

Removing Effectivity Variables from Effectivity Scopes

Important

Do not remove an Effectivity Variable from an Effectivity Scope if it is used in an Effectivity Expression for the given Scope.

Use the standard procedure for removing attached Items on the **Effectivity Variables** accordion tab of *an Effectivity Scope* Item view.

Effectivity Scope ItemType

Effectivity Variables

Effectivity Variables

+

+

×

🔍

✖

Hidden

⌵

🔗

🖥

🔗

	Name	Type	List [...]	ItemType [...]
	Unit	Integer		
	Date	Date		
	Model	Item		effs_model

Creating Effectivity Scopes

Use the standard Item creation procedure to create a new Effectivity Scope. The following fields are required:

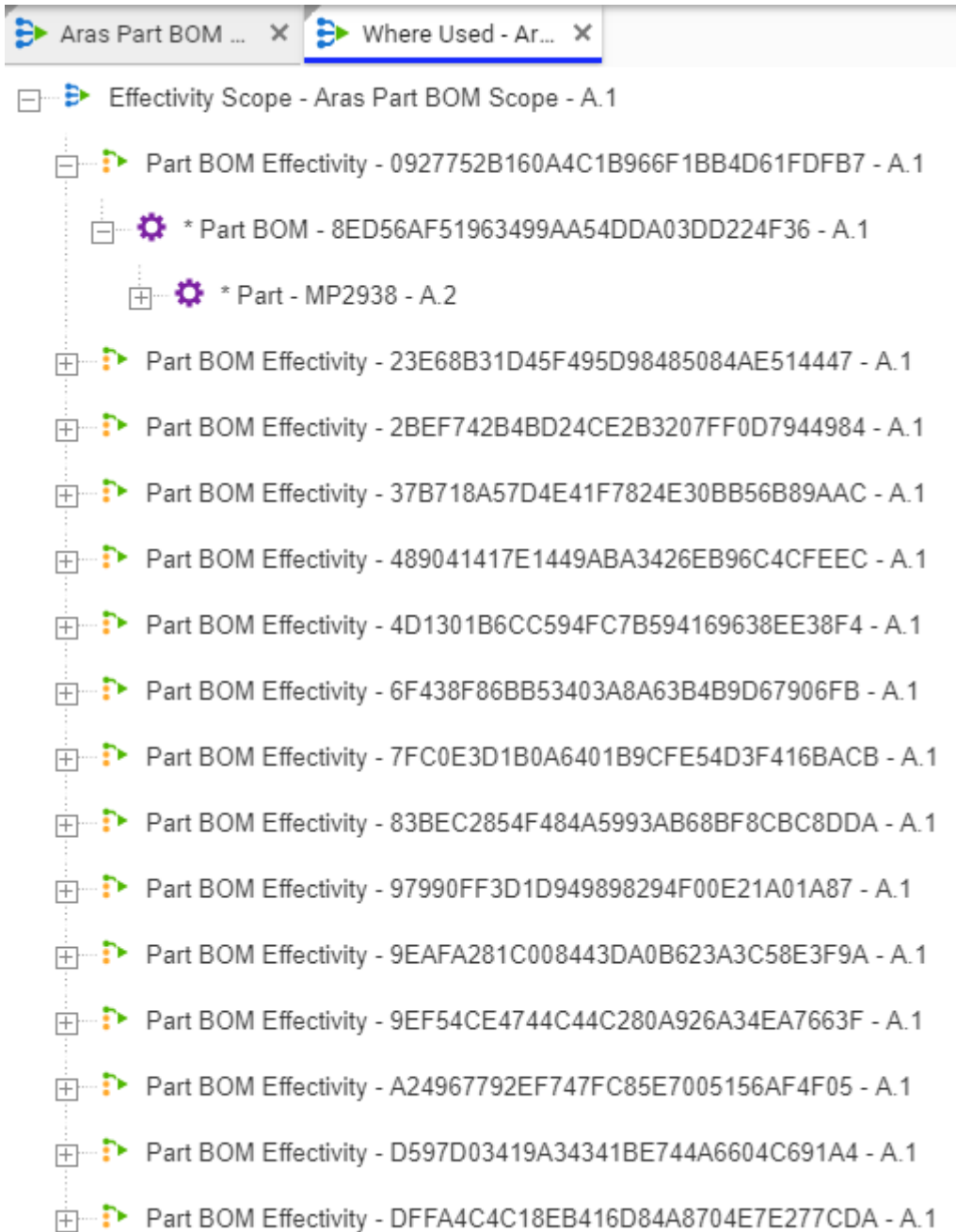
- **Name**
- **Builder Method** - Use the **effs_scopeObjectBuilderMethodmethod** unless you have implemented a different Method that uses custom business logic.

Specify Effectivity Scope ItemTypes and Effectivity Variables on the corresponding accordion tabs.

Finding Where Effectivity Scope is Used

To find out where an Effectivity Scope is used, you can use the **Where Used** browser.





Understanding Query Definitions for Effectivity Resolution

This section provides an overview of the Query Definitions (QD) configuration for resolving the BOM structures against Effectivity Criteria.

- Definition of a QD containing Effectivity filtering logic.



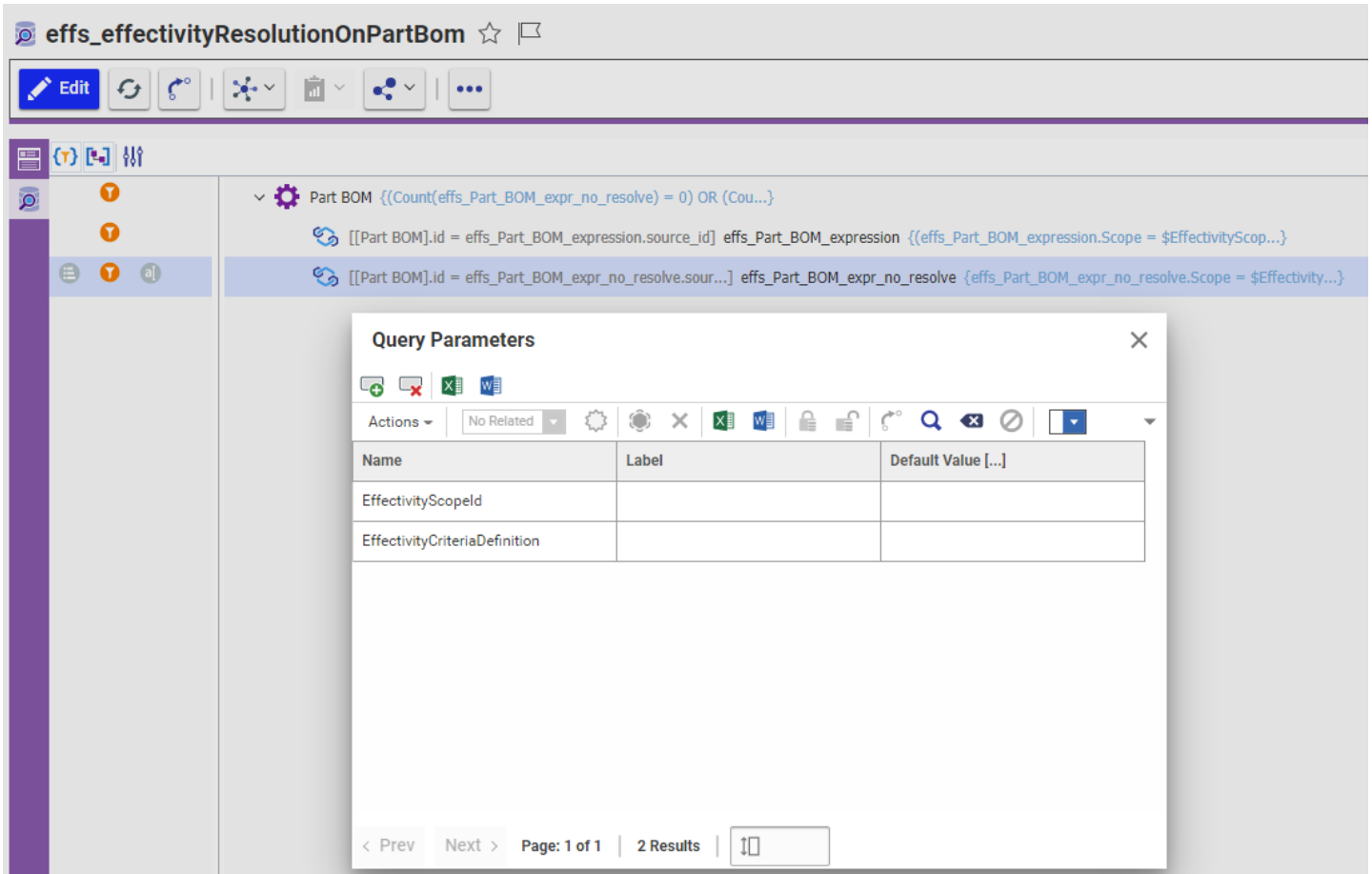
- Merging this QD into another QD to create a single executable QD for a related Tree Grid View.

In the Product Engineering (PE) application, Part BOM Relationships are resolved against Effectivity Criteria on the **BOM Structure** tab of a Part Item. The resolution uses Effectivity Scope and Effectivity Variable values set in the **Effectivity Criteria Filter** dialog. The resolved structure is displayed using the **PE_BomStructure** Tree Grid View (TGV) configuration.

You can configure the **PE_BomStructure** TGV to add, remove, or reorder columns to display Part and Part BOM properties. Adding or removing columns requires changing the QD configuration to return the given properties. You can use the standard TGV and QD procedures for this configuration.

The QD containing Effectivity filtering logic is **effs_effectivityResolutionOnPartBom**. It retrieves assigned Effectivity Expressions and evaluates them against the given Effectivity Criteria in the Resolution Engine. The QD filters out Items that are evaluated as not valid for the given Effectivity Criteria.





The **effs_effectivityResolutionOnPartBom** QD passes data to the Effectivity Resolution Engine using the following Query Parameters:

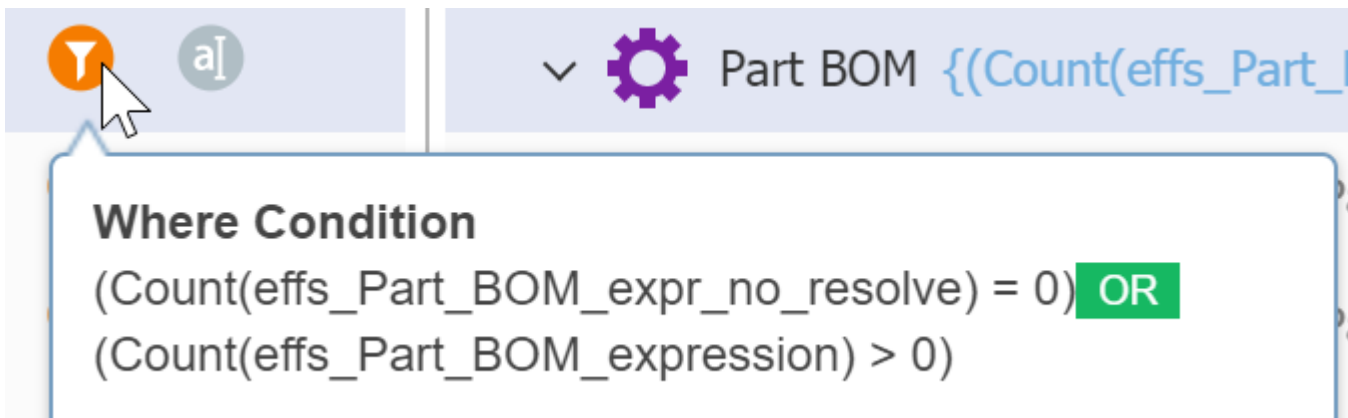
- **EffectivityScopeld** - the ID of the Effectivity Scope.
- **EffectivityCriteriaDefinition** - the Effectivity Criteria Definition.

The Query Structure of the **effs_effectivityResolutionOnPartBom** QD is as follows:

- **Part BOM is the Effective ItemType** of the PE application Effectivity Scopes.
Where Condition:
 $(\text{Count}(\text{effs_Part_BOM_expr_no_resolve}) = 0) \text{ OR } (\text{Count}(\text{effs_Part_BOM_expression}) > 0)$

Important

You can change the **Where Condition** to use AND logic instead of OR. The logic to use depends on how multiple Effectivity Expressions on a given single Part BOM Item should be evaluated. For details, refer to section *Configuring Logic to Evaluate Effectivity Expressions for Effectivity Resolution*.



Where Condition

(Count(effs_Part_BOM_expr_no_resolve) = 0) **OR**
 (Count(effs_Part_BOM_expression) > 0)

Important

It is recommended to use only the "Count()" aggregate function for Effectivity relationships to prevent performance degradation.

- **effs_Part_BOM_expression** is a RelationshipType with the source Part BOM ItemType.

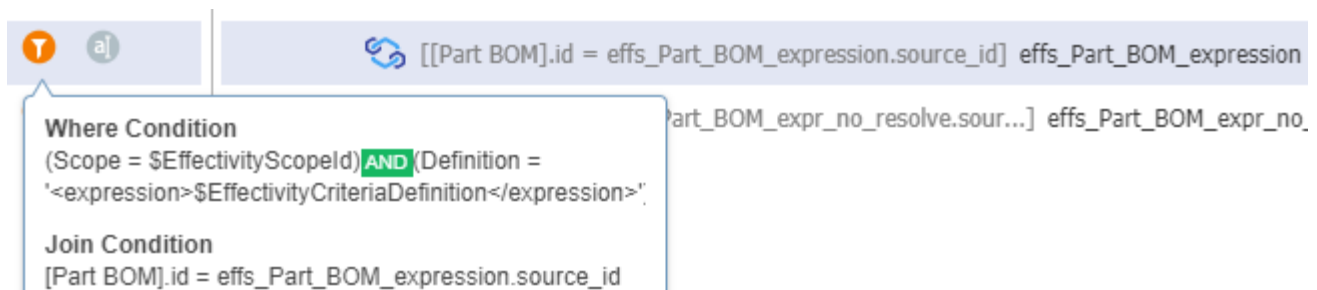
Where Condition:

(Scope = \$EffectivityScopeId) AND (Definition = '\$EffectivityCriteriaDefinition')

The Effectivity Scope ID value is set from the **Scope** cell, and the Effectivity Criteria Definition value is set from the **Variables** and **Values** cells of the **Effectivity Criteria Filter** dialog.

Join Condition:

[Part BOM].id = effs_Part_BOM_expression.source_id



Where Condition

(Scope = \$EffectivityScopeId) **AND** (Definition = '<expression>\$EffectivityCriteriaDefinition</expression>')

Join Condition

[Part BOM].id = effs_Part_BOM_expression.source_id



- **effs_Part_BOM_expr_no_resolve** is used to check if the Part BOM item has any effectivity relationship items for the specified scope.

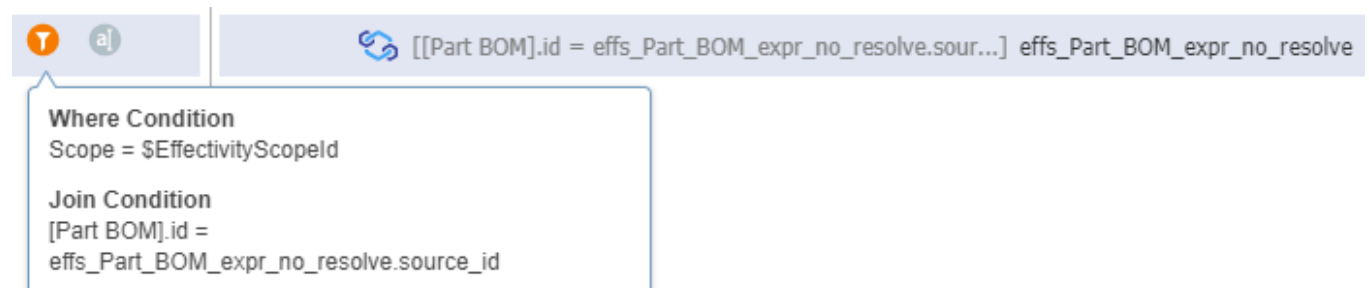
Where Condition:

`Scope = $EffectivityScopeId`

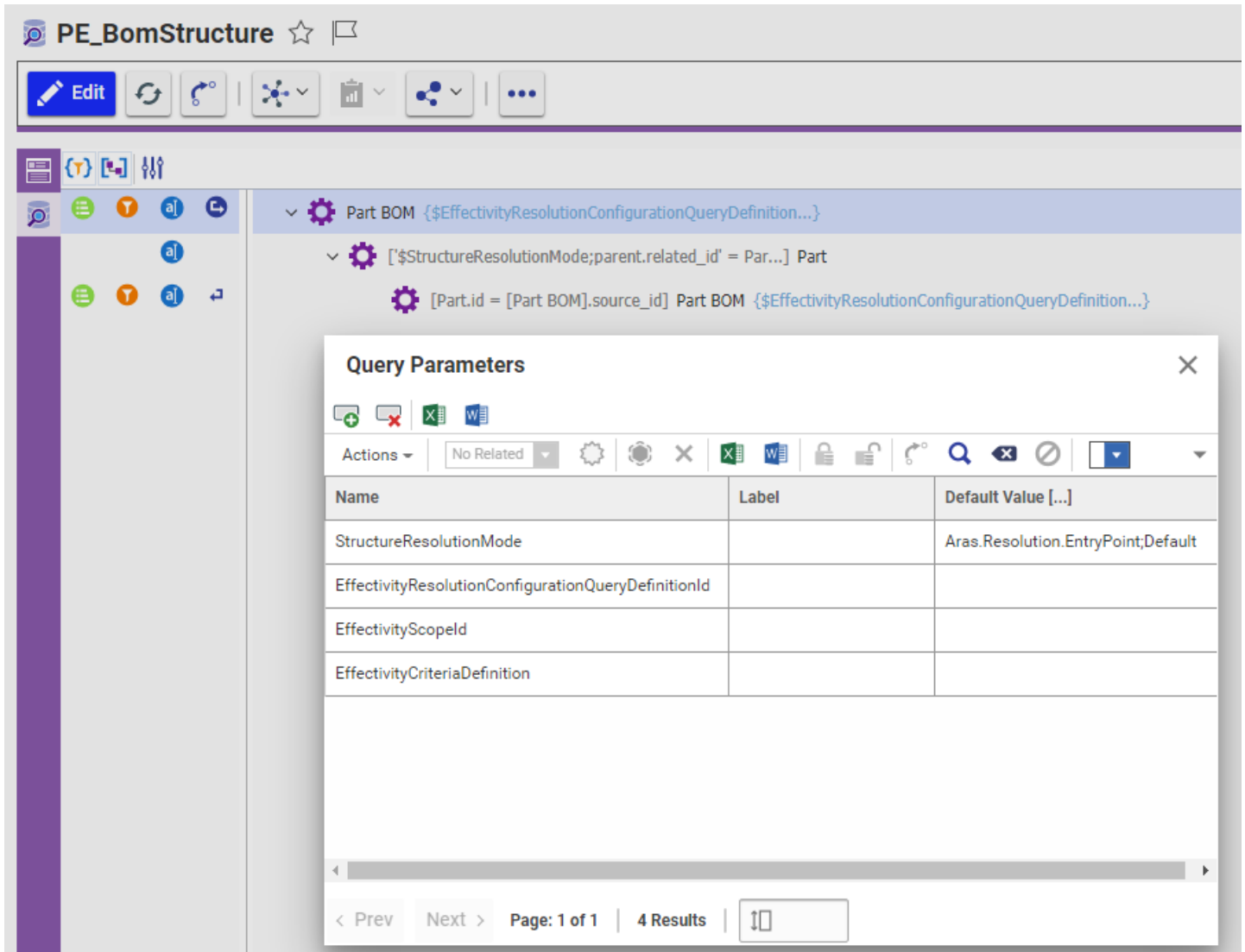
The Effectivity Scope ID value is set from the **Scope** cell.

Join Condition:

`[Part BOM].id = effs_Part_BOM_expr_no_resolve.source_id`



The **effs_effectivityResolutionOnPartBom** QD is merged into the **PE_BomStructure** QD to produce a single executable QD, as described later in this section.

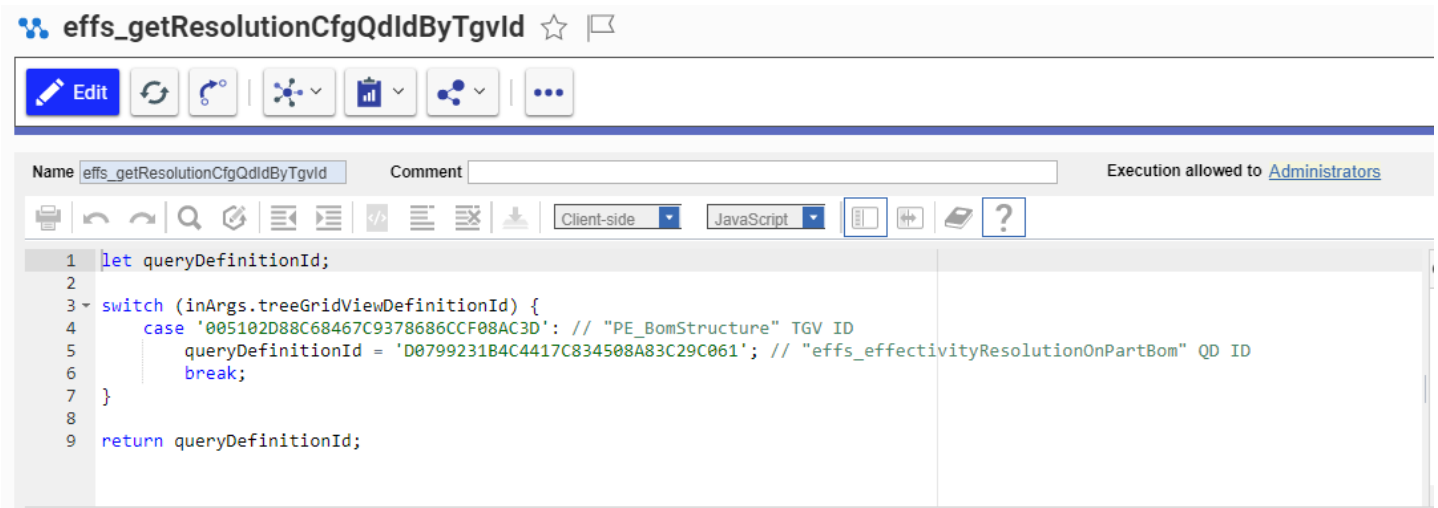


The **PE_BomStructure** QD has the following Query Parameters:

- **EffectivityResolutionMode** — structure resolution mode.
Aras.Resolution.EntryPoint;Default is the default value.
- **EffectivityResolutionConfigurationQueryDefinitionId** — ID of the QD containing Effectivity filtering logic to enable Effectivity resolution.
- **EffectivityScopeId** — ID of the Effectivity Scope Item passed to the Effectivity Resolution Engine.
- **EffectivityCriteriaDefinition** — the Effectivity Criteria Definition passed to the Effectivity Resolution Engine.



When Effectivity Resolution is triggered from the **Effectivity Criteria Filter** dialog (see section *Understanding TGV Definition for Effectivity Resolution*), the **EffectivityResolutionConfigurationQueryDefinitionId** parameter value is retrieved from the **effs_getResolutionCfgQdIdByTgvId** Method.



The method defines 1-to-1 associations between Tree Grid Views and Query Definitions containing Effectivity Resolution logic to be merged for Effectivity Resolution, when triggered from the **Effectivity Criteria Filter** dialog.

The Query structure of the **PE_BomStructure** QD is recursive:

- **Part BOM is the** root ItemType of the **BOM Structure** accordion tab.

Order:

1. Sequence (Ascending)

Where Condition:

\$EffectivityResolutionConfigurationQueryDefinitionId =
\$EffectivityResolutionConfigurationQueryDefinitionId

Important

This condition is used as an anchor to detect the place in the QD where the **effectivityResolutionOnPartBom** QD is merged. This condition should not be removed otherwise Effectivity Resolution will not work.



Properties:

Quantity, Sequence, Reference Designator, Expression, id

The image displays three sequential screenshots of a software interface, likely a query builder or data management tool, showing different views of a 'Part BOM' entity.

- Top Screenshot:** Shows the 'Order' tab. A dropdown menu is open, displaying '1. Sequence' and '(Ascending)'. The background shows a table with columns for 'Part BOM' and a query definition.
- Middle Screenshot:** Shows the 'Where Condition' tab. A text input field contains the condition: `$EffectivityResolutionConfigurationQueryDefinitionId = $EffectivityResolutionConfigurationQueryDefinitionId`. The background shows the same table structure.
- Bottom Screenshot:** Shows the 'Properties' tab. A list of properties is displayed: Quantity, Sequence, Reference Designator, Expression, and id. The background shows the same table structure.

- **Part**

Join Condition:

'\$StructureResolutionMode;parent.related_id' = Part.id

Properties:



Part Number, Revision, State, locked_by_id, Name, id

Where Condition
Join Condition
'\$StructureResolutionMode;parent.related_id' = Part.id

Properties

- Part Number
- Revision
- State
- locked_by_id
- Name
- id

- **Part BOM** is extracted recursively – all its values are the same as for the top **Part BOM**, and the following **Join Condition** is added:

Part.id = [Part BOM].source_id

Order
1. Sequence (Ascending)

Where Condition
\$EffectivityResolutionConfigurationQueryDefinitionId = \$EffectivityResolutionConfigurationQueryDefinitionId

Join Condition
Part.id = [Part BOM].source_id

Properties

- Quantity
- Sequence
- Reference Designator
- Expression
- id

To merge the QD for Effectivity Resolution, the **PE_BomStructure** QD has the **OnBeforeExecute** event set for the **effs_mergeQueryDefinition** Method.

Name: PE_BomStructure Context Item Type: Part BOM

Description:

qry_QueryItem qry_QueryReference qry_QueryParameter **qry_QueryDefinitionEvent**

Methods

Name ▲	Method Type	Ver	execution_allowed_to [...]	Comments	event
qry_ResolveStructureEntryPoint	CSharp	1	Administrators		OnBeforeExecute
effs_mergeQueryDefinition	CSharp	1	Administrators		OnBeforeExecute

By default, the QD Item view does not display tabs. To view and to add events to QDs, make tabs visible by setting the **Default Structure View** property of the **qry_QueryDefinition** ItemType to **Tabs On**.

qry_QueryDefinition

Save Done Discard

ItemType

Name: qry_QueryDefinition History Template:

Singular Label: Query Definition Plural Label: Query Definitions

Show Parameters Tab: When Populated

Class Structure

Default Structure View: Tabs Off, **Tabs On**, Tabs Off, Tabs Max, Tabs Min

Understanding TGV Definition for Effectivity Resolution

This section provides an overview of how Tree Grid View (TGV) is configured to work with a corresponding Query Definition (QD) to display the resolved structure for user-provided Effectivity Criteria. In the Product Engineering (PE) application, the Part BOM Relationships are resolved against Effectivity Criteria on the **BOM Structure** tab of a Part Item. This tab uses a TGV to display data retrieved from a QD as a tree grid. You can configure the **PE_BomStructure** TGV to add, remove, or reorder columns to display Part and Part BOM properties. Adding or removing columns requires changing the QD configuration to return the given properties. You can use the standard TGV and QD procedures for this configuration. The **PE_BomStructure** TGV configuration has the following UI elements: the **Set Effectivity Criteria** button, the **View Effectivity** context menu item, and the **Manage Effectivity Expressions** split pane.

Part Number	Set Effectivity Criteria	State	S...	Q...	Name	Effectivity	Clai...	Refer
Child Part A	A	Prelimi...	1	1	Child Part A			
Child Part A.1	A	Prelimi...	1	1	Child Part A.1	Model = R...		
Child Part A.2	A	Prelimi...	2	1	Child Part A.2			
Child Part B	A	Prelimi...	2	1	Child Part B			

Effectivity Scope	Effectivity Expression
Aras Part BOM Scope	Model = Replicator

You can configure the **PE_BomStructure** TGV to add, remove, or reorder columns to display Part and Part BOM properties. Adding or removing columns requires changing the QD configuration to return the given properties. Use the standard TGV and QD configuration procedures.

Part Number	Revision	State	Sequence	Quantity	Claimed By	Name	Effectivity	Reference Designator	
Part BOM --- Part	{Part.item_number}	{Part.major_rev}	{Part.state}	{Part BOM.sort_order}	{Part BOM.quantity}	{Part.locked_by_id/@keyed_name}	{Part.name}	{Part BOM.eff_expression_string_notation}	{Part BOM.reference_designator}
Part BOM	{Part.item_number}	{Part.major_rev}	{Part.state}	{Part BOM.sort_order}	{Part BOM.quantity}	{Part.locked_by_id/@keyed_name}	{Part.name}	{Part BOM.eff_expression_string_notation}	{Part BOM.reference_designator}

©2026 Aras Corporation All Copyrights Reserved



PE_BomStructure Presentation Configuration

Edit

^ Presentation Configuration

Color

Name

PE_BomStructure Preser

^ Command Bar Section

cui_PresentConfigWinSection

CommandBarSection

Hidden

	Classification	Name	Location [...]	sort_order	For Identity [...]
	Data Model	effs_tgvToolbarCustomization	TGV Toolbar	128	World
	Data Model	effs_tgvContextMenuCustomization	TGV ContextMenu	256	World

The **effs_tgvToolbarCustomization** command bar section adds the new button to the default TGV toolbar. This button is **effs_tgvToolbarSetEffectivityCriteriaButton** labeled as **Set Effectivity Criteria**.



effs_tgvToolbarCustomization

Edit

^

Command Bar Section

Name

effs_tgvToolbarCustomization

Location

TGV Toolbar

Classification

Data Model

Label

Description

^

Command Bar Item

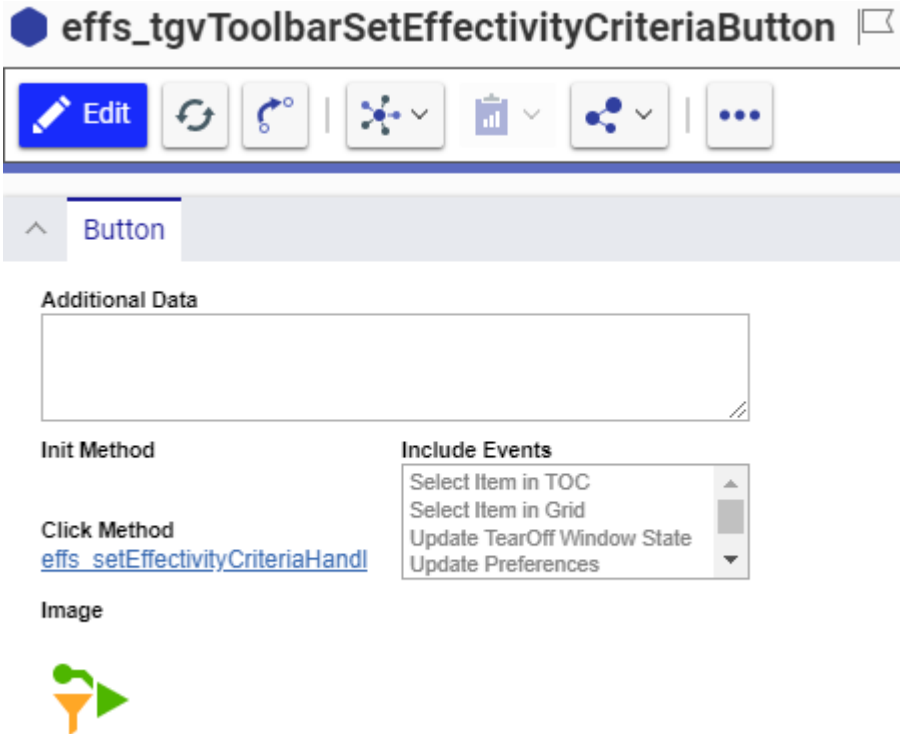
Command Bar Section Dependency

CommandBarItem

Hidden

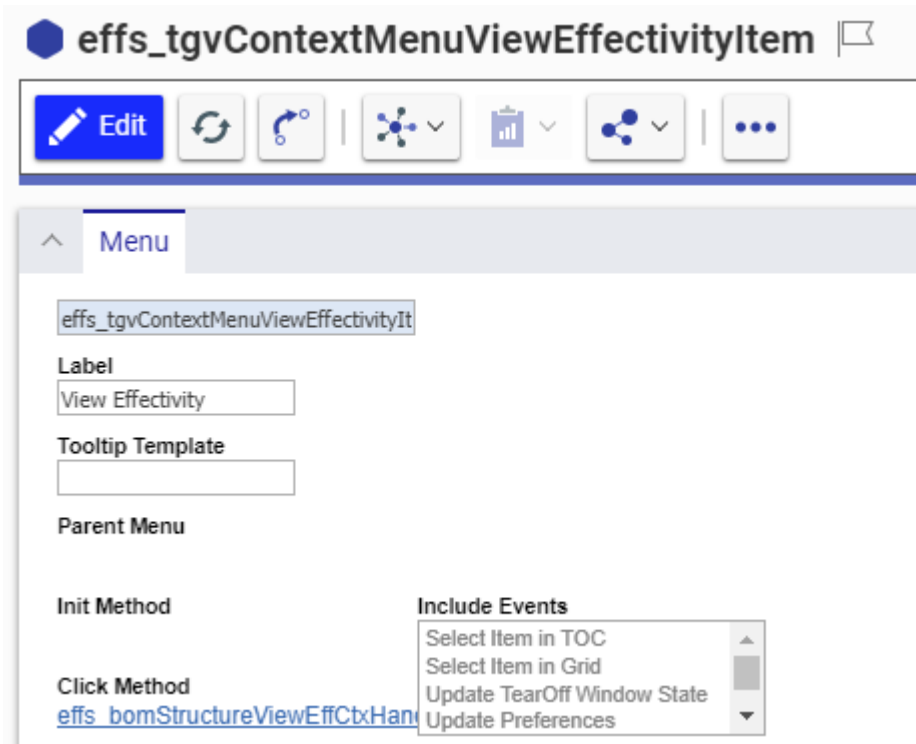
	ItemType	Name	Sort Order	Action	For Identity
	Button	effs_tgvToolbarSetEffectivityCriteriaButton	320	Add	World

Set Effectivity Criteria triggers the **effs_setEffectivityCriteriaHandl** Method to show the **Effectivity Criteria Filter** dialog for setting the Effectivity Criteria, for which a BOM structure will be resolved.



The **effs_tgvContextMenuCustomization** command bar section adds a new menu item to the default TGV context menu. This menu item is **effs_tgvContextMenuViewEffectivityItem** labeled as **View Effectivity**.





Enabling Manual Resize for Custom Dialogs

Custom dialogs can be configured to support manual resizing by setting the `enableManualResize`` option.

Example use case Set the `enableManualResize`` option when creating a custom dialog:

```
``typescript
import { Dialog } from 'core';
const dialog = Dialog.show('iframe', {
  content: 'myCustomDialog.html',
  title: 'Custom Dialog',
  dialogWidth: 600,
  dialogHeight: 400,
  enableManualResize: true
});
``
```

See usage examples in `Modules/cui/plugins/dialogs.ts``.

Custom Dialog Types



When adding a custom dialog type to `dialogTypes` in `Dialog.ts`, include the `enableManualResize` property:

```
``typescript
const dialogTypes = {
  MyCustomDialog: {
    title: 'mycustomdialog.title',
    content: 'myCustomDialog.html',
    classList: 'aras-dialog_my-custom',
    enableManualResize: true
  }
};
```

See examples in `Modules/core/Dialog.ts` (SearchDialog, ImageBrowser definitions).

Controlling the Visibility of Effectivity

In Aras Innovator, Effectivity Management on Part BOM structures is available out of the box. If your business is not going to use this feature, you can hide its associated UI elements. This section explains how to hide and redisplay the discussed UI elements. These elements are as follows:

- The **Effectivity Services** subcategory under **Administration** in **Contents**:
 - **Effectivity Scope**
 - **Effectivity Variables**
 - **Models**
- The **Effectivity** tab in the **Part BOM** Relationship Item view.
- The **BOM Structure** tab in the **Part** Item view:
 - The **Set Effectivity Criteria** button in the **BOM Structure** toolbar
 - The **View Effectivity** menu item in the context menu of a Part
 - The **Effectivity** column

Important

Hiding Effectivity-related UI elements does not delete any existing Effectivity Items. If you redisplay the UI elements, the Effectivity Items will be available.

Visibility of the Effectivity Services TOC Subcategory



The **Effectivity Services** subcategory of the **Administration** category in the **Table of Contents** (TOC) contains the following ItemTypes:

- **effs_scope** (Effectivity Scope)
- **effs_variable** (Effectivity Variable)
- **effs_model** (Model)

The visibility of the **Effectivity Services** category depends on the visibility of its contained ItemTypes:

- Visible when at least one of the ItemTypes is visible.
- Hidden when all the ItemTypes are hidden.

You can hide and redisplay the **Effectivity Services** ItemTypes in any sequence. Use the standard TOC configuration.

When redisplaying an **Effectivity Services** ItemType, use the **Effectivity Management** Identity and **Effectivity Services** Category.

effs_scope x

effs_scope ☆

Edit

ItemType

PropertiesRelationshipTypesViewsServer EventsActionsLife CyclesWorkflowsTOC Access

Identities ☆

Hidden

Name	Category ↑
Effectivity Management	Effectivity Services

Visibility of the Effectivity Tab in Part BOM Item View

©2026 Aras Corporation All Copyrights Reserved

27

By default, the **Part BOM** Relationship Item view includes the **Effectivity** accordion tab.

Parts x

Parent Part x

C14F7A09EEF5... x

C14F7A09EEF54AFA90A69D1096F1D242

Edit

Part BOM

Quantity

1

Part Number

Child Part A

Reference Designator

Instances

Substitutes

Effectivity

Part BOM Effectivity

Hidden

Scope [...]

Expression [...]

Aras Part BOM Scope

Unit = 100

The visibility of the **Effectivity** accordion tab is controlled by the **Hide In All** check box in the **Source** group of the **effs_Part_BOM_expression** RelationshipType Item view.



RelationshipTyp... x effs_Part_BOM_... x

effs_Part_BOM_expression

Save Done Discard Refresh

RelationshipType

Name
effs_Part_BOM_expression

Label
Effectivity

Description

Source
Source ItemType
Part BOM ...

☐ Hide In All

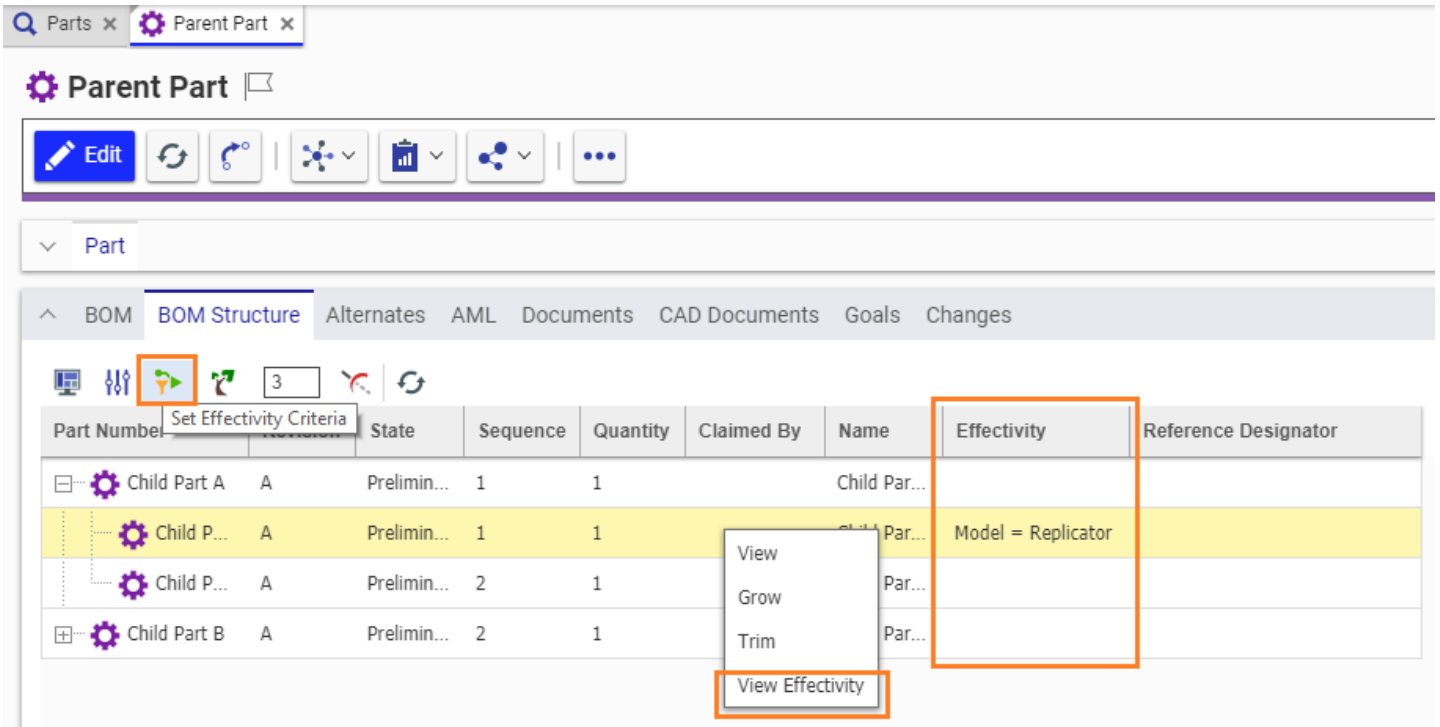
Paste Defaults
☐ Copy Permissions

To hide this tab, select the **Hide In All** check box. To display this tab, clear the check box.

Visibility of the Effectivity-Related Elements in Part Item View

By default, the **BOM Structure** accordion tab of the **Part** Item view has the following Effectivity Management UI elements:

- The **Set Effectivity Criteria** button in the **BOM Structure** toolbar.
- The **View Effectivity** menu item in the context menu of a Part.
- The **Effectivity** column.



This section describes configuration specifications for the **PE_BomStructure** QD and TGV, which shows these UI elements as well as calls Effectivity filtering.

If you need to hide the discussed elements, remove the described configuration from the TGV and then from the QD. If you need to redisplay the discussed elements after you have removed them, configure the QD and then the TGV according to the given specifications. Relog into Aras Innovator to confirm the change.

PE_BomStructure TGV Configuration for Visibility of Effectivity-Related Elements

To show the Effectivity-related UI elements on the **BOM Structure** accordion tab of the **Part** Item view, the **PE_BomStructure** TGV is configured as specified in this section. The **PE_BomStructure** TGV uses the **PE_BomStructure Presentation Configuration**.

Tree Grid Views x

PE_BomStructure x

File

Edit

Views

Actions

Reports

Tools

Help

+

↓

✖

↺

🖨

📊

📄

🔒

🔓

↶

↷

🔗

?

✓

📄

📄

Name

PE_BomStructure

Query Definition

PE_BomStructure

Context Item Type

Part BOM

Description

Max Visible Children On Expand

100

Max Grow Levels

3

Linked Toolbar/Context Menu

PE_BomStructure Presentation Configuration

The **PE_BomStructure Presentation Configuration** is configured to show the **Set Effectivity Criteria** toolbar button and the **View Effectivity** context menu item to the **World** Identity:

- **effs_tgvToolbarCustomization** located on **TGV_Toolbar** with sort order **128**.
- **effs_tgvContextMenuCustomization** located on **TGV_ContextMenu** with sort order **256**.

Tree Grid Views x
PE_BomStructure x
PE_BomStructu... x

PE_BomStructure Presentation Configuration

Edit
Refresh
Undo
Share
Add
Remove
More

Presentation Configuration

Command Bar Section
cui_PresentConfigWinSection

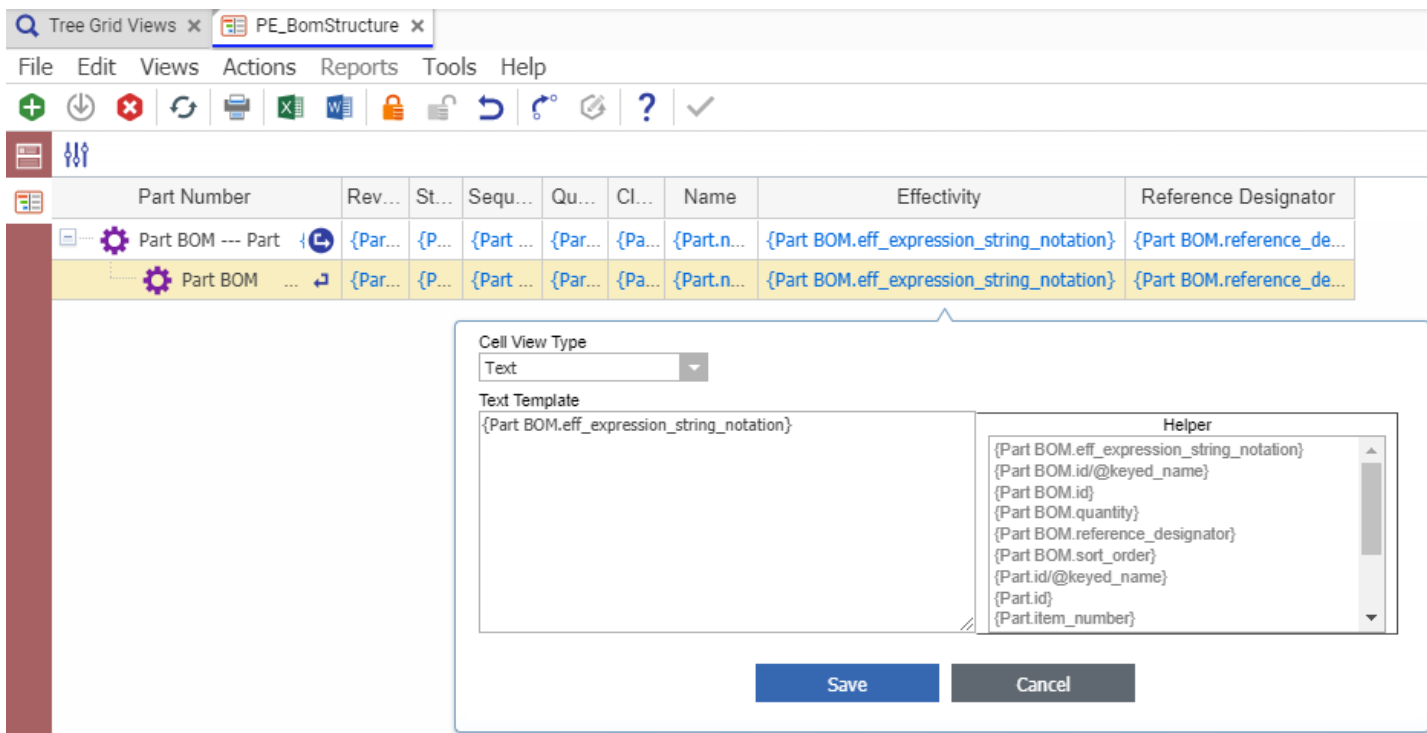
CommandBarSection

Add
Remove
Search
Close
Hidden
Fullscreen
Zoom
Share

	Classification	Name	Location [...]	sort_order	For Identity [...]
	Data Model	effs_tgvToolbarCustomization	TGV Toolbar	128	World
	Data Model	effs_tgvContextMenuCustomization	TGV ContextMenu	256	World

To display the **Effectivity** column on the **BOM Structure** accordion tab of the **Part** Item View, the **PE_BomStructure** TGV has the **Effectivity** column between the **Name** and **Reference Designator** columns specified to show data of the **Text** type according to the **{Part BOM.eff_expression_string_notation}** text template.





To hide the Effectivity-related UI elements on the **BOM Structure** tab of the **Part** Item View, remove **effs_tgvToolbarCustomization** and **effs_tgvContextMenuCustomization** from the **PE_BomStructure Presentation Configuration** as well as the **Effectivity** column in the TGV editor, and then reconfigure the **PE_BomStructure** QD. To redisplay these elements after they were hidden, reconfigure the **PE_BomStructure** QD, and then add the command bar sections and Effectivity column as specified.

PE_BomStructure QD Configuration for Visibility of Effectivity-Related Elements

To show the Effectivity-related UI elements on the **BOM Structure** accordion tab of the **Part** Item view, the **PE_BomStructure** QD is configured as follows:

- The **qry_QueryDefinitionEvent** relationship tab has an **OnBeforeExecute** event using the **effs_mergeQueryDefinition** Method.

Important

For more information about this Method and how to display and hide tabs in the **QD** Item view, refer to section *Understanding Query Definitions for Effectivity Resolution*.



ItemTypes x qry_QueryDefini... x Query Definitions x PE_BomStructure x

File Edit Views Actions Reports Tools Help

+ ↻ ⛔ ↺ 🖨️ 📄 🔒 🔓 ↶ ↷ 🔗 ? ✓

Name PE_BomStructure **Context Item Type** [Part BOM](#)

Description

qry_QueryItem qry_QueryReference qry_QueryParameter [qry_QueryDefinitionEvent](#)

Methods ☆

🔍 ⛔ Hidden 🔍 🖨️ 🔗

	Name ▲	Method Type	Ver	execution_allo...	Comments	event
	qry_ResolveStructureEntryPo...	CSharp	1	Administrators		OnBeforeExecute
	effs_mergeQueryDefinition	CSharp	1	Administrators		OnBeforeExecute

- The top **Part BOM** item in the QD editor has the **Expression** and **id** properties.

ItemTypes x qry_QueryDefini... x Query Definitions x PE_BomStructure x

File Edit Views Actions Reports Tools Help

+ ↻ ⛔ ↺ 🖨️ 📄 🔒 🔓 ↶ ↷ 🔗 ? ✓

🔍 ⛔ 🔗

🔍 ⛔ 🔗

▼ ⚙️ Part BOM {\$EffectivityResolutionConfigurationQueryDefinition...}

parent.related_id' = Par...] Part

Properties

- Quantity
- Sequence
- Reference Designator
- Expression
- id



When hiding the Effectivity-related UI elements on the **BOM Structure** tab of the **Part** Item View, remove the **effs_mergeQueryDefinition** Method and the **Expression** and **id** properties after reconfiguring the **PE_BomStructure** TGV. When redisplaying these elements after they were hidden, add these Method and properties back before reconfiguring the **PE_BomStructure** TGV.

Configuring Logic to Evaluate Effectivity Expressions for Effectivity Resolution

A Part BOM Item can have several Effectivity Expressions.

Parent Part 

▼ Part

^ BOM **BOM Structure** Alternates AML Documents CAD Documents Goals Changes

Part Number	R...	State	Se...	Qu...	Claimed...	Name	Effectivity
 Child Part A	A	Prelimi...	1	1		Child Part A	(Model = Replicator), (Unit = 100)
 Child Part B	A	Prelimi...	2	1		Child Part B	

 **BOM Effectivity: Child Part A Child Part A**

Effectivity Scope	Effectivity Expression ↑
Aras Part BOM Scope	Model = Replicator
Aras Part BOM Scope	Unit = 100

By default, multiple Effectivity Expressions on a Part BOM Relationship Item are evaluated using the OR conditional operator for the Effectivity Resolution: A given Part BOM is effective against Effectivity Criteria if *at least one* of its Effectivity Expressions evaluates to true for the selected Effectivity Scope. It is possible to change the evaluation logic to use the AND conditional operator: A given Part BOM is effective against Effectivity Criteria if *all* of its Effectivity Expressions evaluate to true for the selected Effectivity Scope.

The evaluation logic is configured in the **Where Condition** of the top **Part BOM** item in the **effs_effectivityResolutionOnPartBom** QD.

This section describes the configuration specifications of this Query Definition for both evaluation logic.

Important

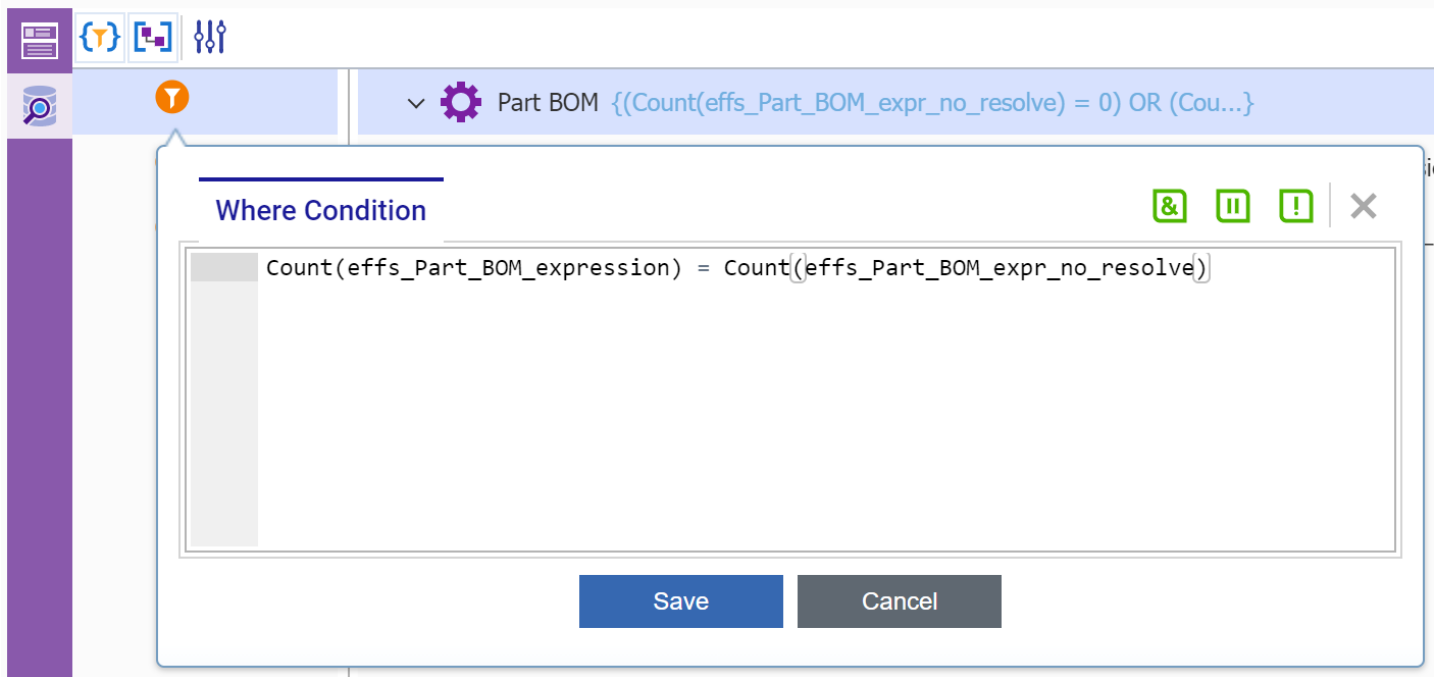
It is recommended to use only the "Count()" aggregate function for Effectivity relationships to prevent performance degradation.



Specification of the AND Evaluation Logic

For the AND evaluation logic, the **Where Condition** of the top **Part BOM** item of the **effs_effectivityResolutionOnPartBom** Query Definition is configured as follows:

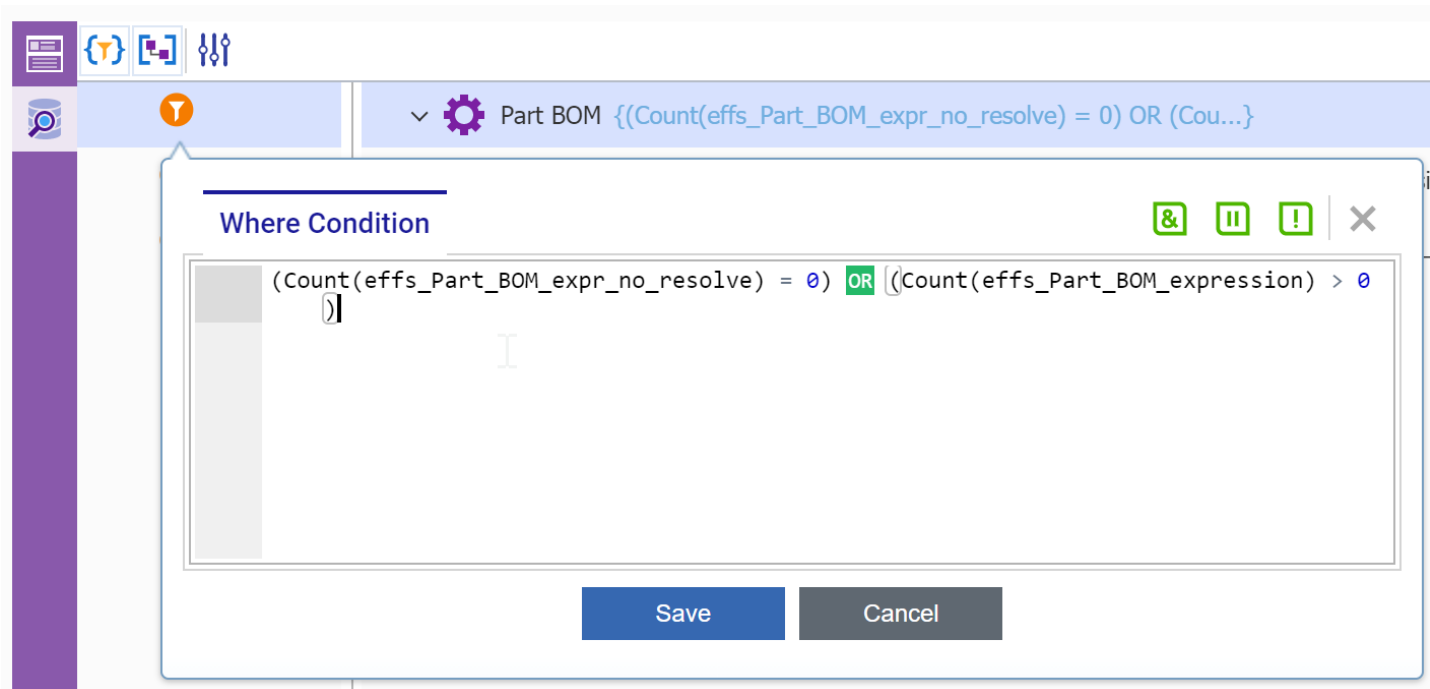
```
Count(effs_Part_BOM_expression) = Count(effs_Part_BOM_expr_no_resolve)
```



Specification of the OR (Default) Evaluation Logic

For the OR (default) evaluation logic, the **Where Condition** of the top **Part BOM** item of the **effs_effectivityResolutionOnPartBom** Query Definition is configured as follows:

```
(Count(effs_Part_BOM_expr_no_resolve) = 0) OR  
(Count(effs_Part_BOM_expression) > 0)
```



Customizing Effectivity String Notations

While an Effectivity Expression is saved as XML, a user-friendly Notation of this Expression is displayed to users. It is possible to change the construction of this Notation. This section describes how the Notation can be customized. Effectivity on a **Part BOM** Relationship is managed using its related **effs_Part_BOM_expression** Items. A **Part BOM** Relationship can have zero, one, or more Effectivities. The **string_notation** text property on the **effs_Part_BOM_expression** ItemType stores the user-friendly Notation for a single Effectivity Expression. The **eff_expression_string_notation** federated property on the **Part BOM** ItemType dynamically constructs the Effectivity text from its related Effectivities. The **string_notation** property from the **effs_Part_BOM_expression** ItemType appears in:

- The **Manage Effectivity Expressions** split pane on the **BOM Structure** tab of the **Part** Item view when the **View Effectivity** context menu item is accessed.
- The **Effectivity** tab of the **Part BOM** Relationship Item view.

The **eff_expression_string_notation** property on the **Part BOM** ItemType is displayed in:

- The **BOM Structure** tab of the **Part** Item view in the **Effectivity** column.

Customizing Effectivity String Notation on Effectivity Expression

Upon its creation and update, an Effectivity Expression gets a new Effectivity String Notation. This behavior is the job of the **effs_expStringNotationConversion** server-side Method executed on the **onBeforeAdd** and **onBeforeUpdate** events of the **effs_Part_BOM_expression** Items. This Method converts an Effectivity Condition to an Effectivity String Notation and stores the output to the **string_notation** Item property. Customization of the Effectivity String Notations requires modification of the discussed Method as follows:

1. Define a new class to represent a new Effectivity String Notation converter and to provide a String Notation value for a given Effectivity Expression object representation. This class should get objects of Effectivity Variables and Effectivity Expression, convert them into Effectivity String Notation, and return this Notation. The class must implement the **Aras.Server.Core.Configurator.IStringNotationConverter** interface, as included in the Appendix.

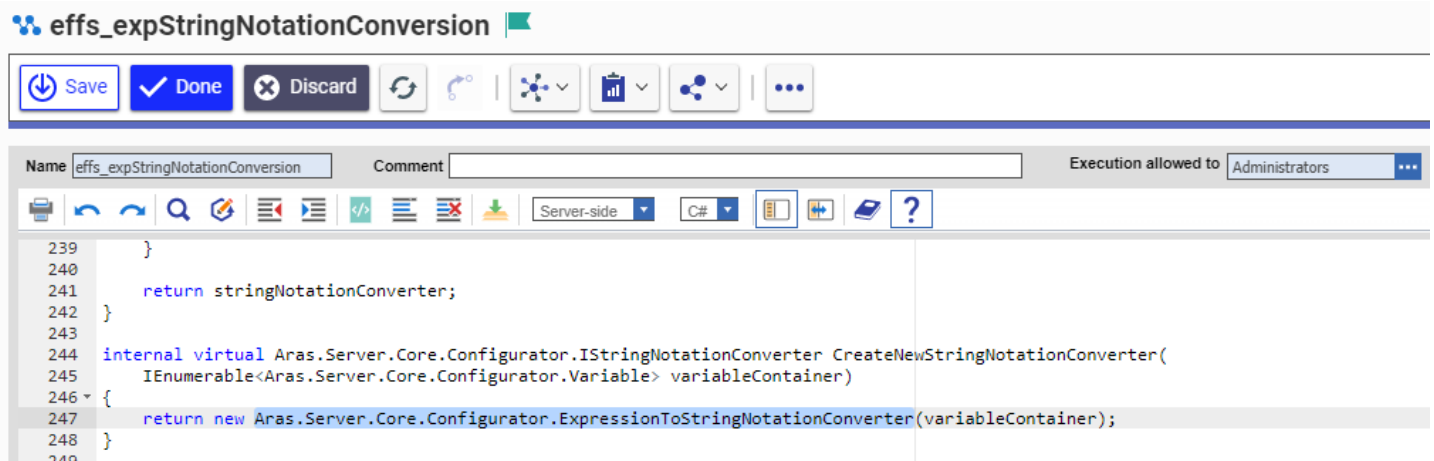
Important

A new class for a custom Effectivity String Notation converter is required because the *Aras.Server.Core.Configurator.ExpressionToStringNotationConverter* class with the default Effectivity String Notation converter is compiled. A customization example is provided in the Appendix.

2. In the **CreateNewStringNotationConverter** local method definition, replace the **Aras.Server.Core.Configurator.ExpressionToStringNotationConverter** class with your custom class in the return statement. Do not change the **variableContainer** argument.

```
internal virtual Aras.Server.Core.Configurator.IStringNotationConverter CreateNewStringNotationConverter(
IEnumerable<Aras.Server.Core.Configurator.Variable> variableContainer)
{
return new Aras.Server.Core.Configurator.ExpressionToStringNotationConverter(
variableContainer);
}
```





Customizing Unified Effectivity String Notation

The **Part BOM** ItemType is the **Effectivity Scope ItemType** of the **Aras Part BOM Scope**. An ItemType used as the **Effectivity Scope ItemType** can have a unified Effectivity String Notation joined from its related single Effectivity Expressions. To provide this feature, the **Part BOM** ItemType uses the:

- **eff_expression_string_notation** federated property.
- **effs_getExpressionStringNotation** server-side Method executed on the **onAfterGet** event.

Customization of the unified Effectivity String Notations requires modification of this Method, where the **JoinExpressionStringNotation** local method definition is changed according to the desired uniting template.

Important

A customization example is provided in the Appendix.

The local method produces a single value for the **eff_expression_string_notation** property. It combines related Effectivity Expressions using a conjunction.

The following is the default implementation of **JoinExpressionStringNotation** where parentheses enclose individual Effectivity String Notations and commas join these Notations.



```
internal virtual string JoinExpressionStringNotation(IEnumerable<string> expressionStringNotations)
{
    if (!string.IsNullOrEmpty(expressionStringNotations.ElementAtOrDefault(1)))
    {
        expressionStringNotations = expressionStringNotations.Select(
            expressionStringNotation => "(" + expressionStringNotation + ")");
    }
    return string.Join(", ", expressionStringNotations);
}
```

effs_getExpressionStringNotation