# Aras Innovator 12

## Tree Grid View Administrator Guide

***Document #: 12.0.02019052001***

***Last Modified:*** *5/12/2020*

# Copyright Information



Aras Corporation
100 Brickstone Square
Suite 100
Andover, MA 01810

**Phone:** 978-806-9400
**Fax:** 978-794-9826

**E-mail:** Support@aras.com

**Website:** https://www.aras.com

# Table of Contents

# Send Us Your Comments

Aras Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for future revisions.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where and what level of detail?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, indicate the document title, and the chapter, section, and page number (if available).

You can send comments to us in the following ways:

**Email:**
Support@aras.com
Subject: Aras Innovator Documentation

Or,

**Postal service:**
Aras Corporation
100 Brickstone Square
Suite 100
Andover, MA 01810
Attention: Aras Innovator Documentation

Or,

**FAX:**
978-794-9826
Attn: Aras Innovator Documentation

If you would like a reply, provide your name, email address, address, and telephone number.

If you have usage issues with the software, visit https://www.aras.com/support/

# Document Conventions

The following table highlights the document conventions used in the document:

Table 1: Document Conventions

| Convention | Description |
|---|---|
| **Bold** | This shows the names of menu items, dialog boxes, dialog box elements, and commands.<br>Example: Click **OK**. |
| `Code` | Code examples appear in `courier` font. It may represent text you type or data you read. |
| `Yellow highlight` | Code highlighted in yellow draws attention to the code that is being indicated in the content. |
| `Yellow highlight with red text` | Red text highlighted in yellow indicates the code parameter that needs to be changed or replaced. |
| *Italics* | Reference to other documents. |
| **Note:** | Notes contain additional useful information. |
| **Warning** | Warnings contain important information. Pay special attention to information highlighted this way. |
| Successive menu choices | Successive menu choices may appear with a greater than sign (-->) between the items that you will select consecutively.<br>Example: Navigate to **File** --> **Save** --> **OK**. |

# 1 Overview

The Tree Grid View application provides a means to build a visual data structure for end users. The data structure can provide information on where a given item fits in the context of other items. It offers a visual layout of the data as a Relationship tab in item view. The application supports sorting on selected columns at every level in the grid that displays hierarchical data.

The Tree Grid Views are grids defined by the administrators. This application takes advantage of the Query Builder application to submit a query to get the necessary data. It then uses the data to populate the grid created by the administrator.

This guide describes the procedures to create a Relationship Tab on the Part ItemType showing a grid like the one depicted here:

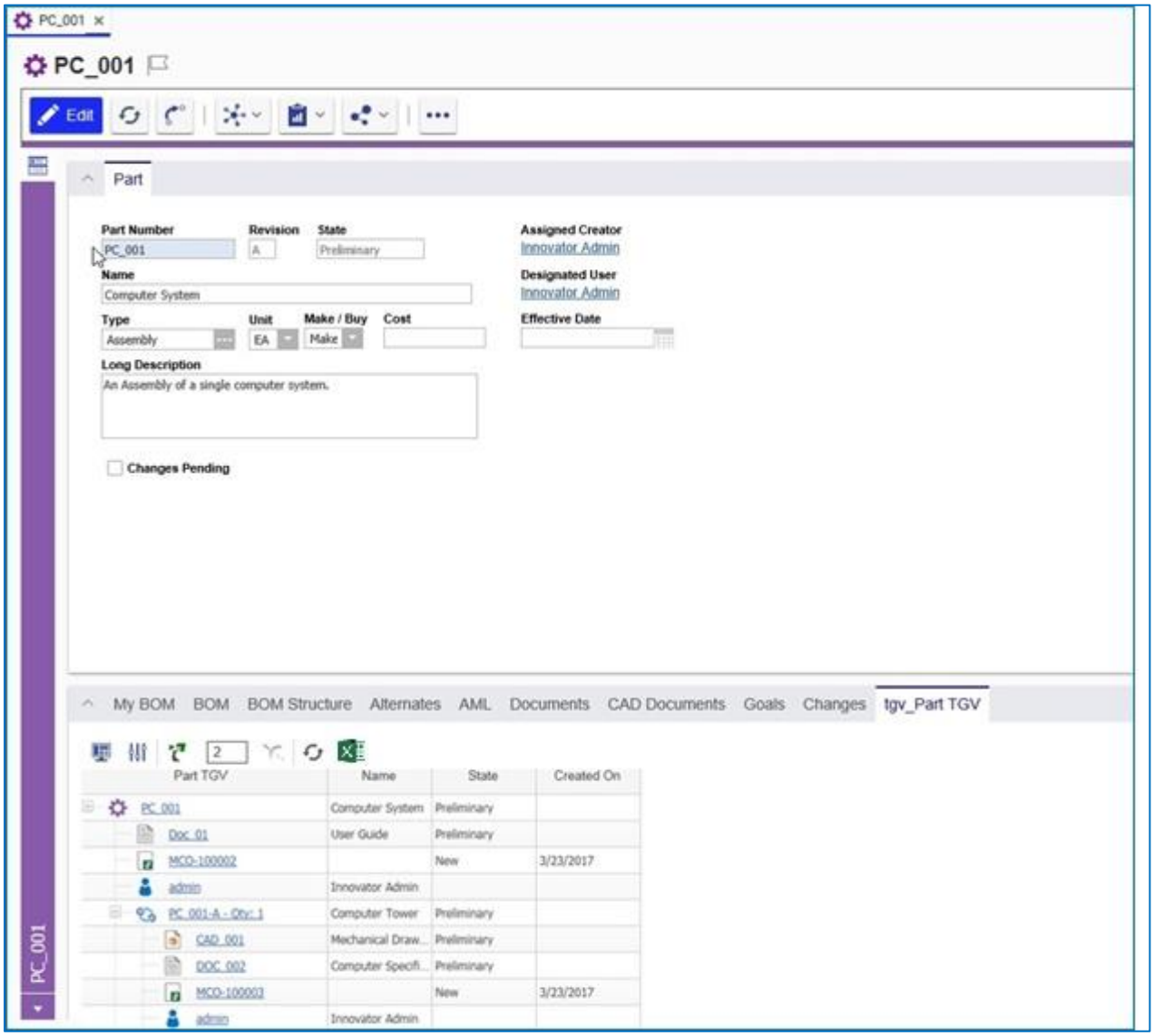


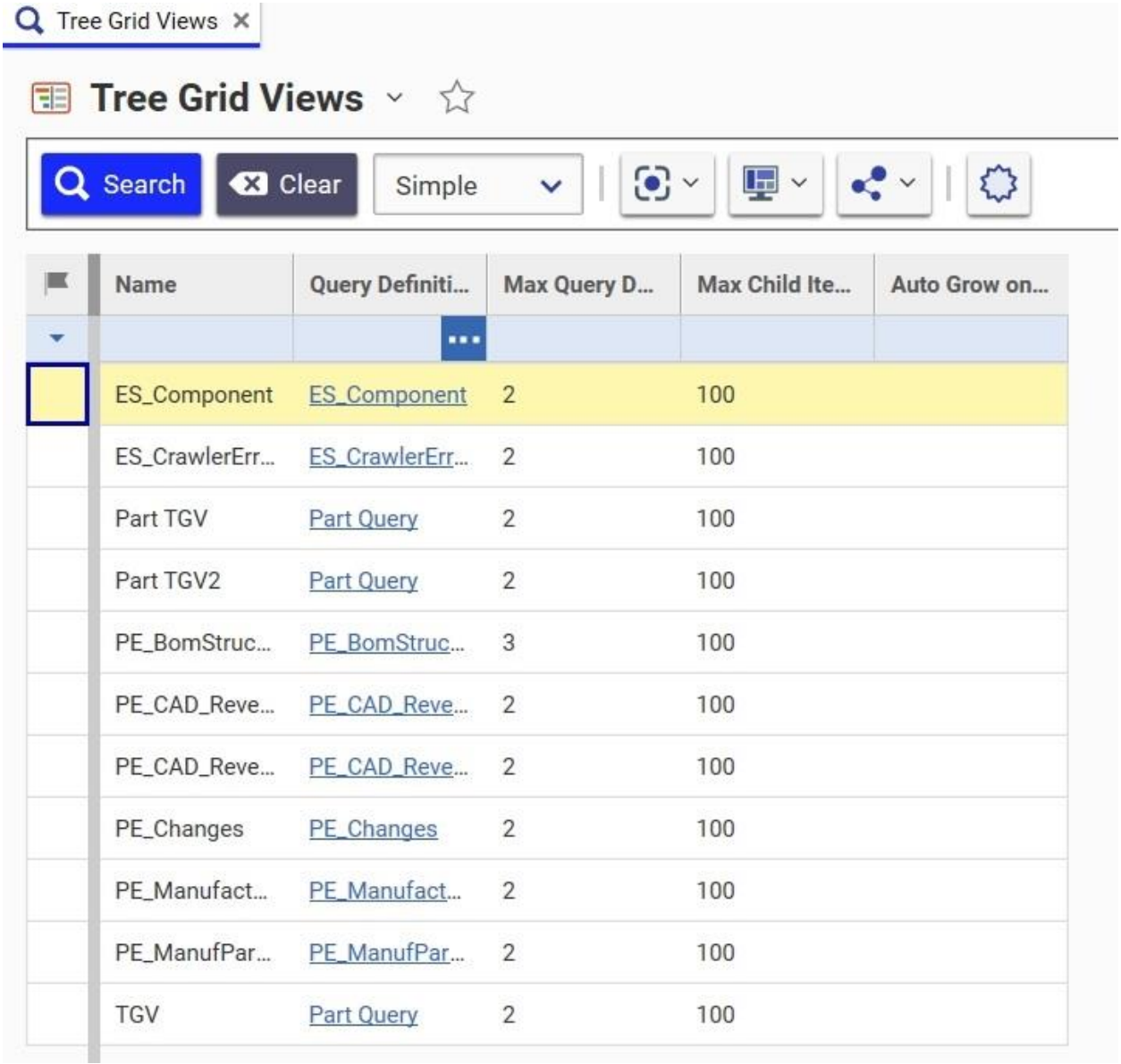

Figure 1.

## 1.1 Support for xProperties and PolyItems

Extended Properties (XProperties) enable you to add text, create rules, and define variables as Item properties. You need to assign xProperties to a specific ItemType. Once you assign them, they can be used by items associated with that ItemType. You need to define xProperties for an Item before you can use them.

# 2 Creating Tree Grid Views

The Tree Grids are defined by the Tree Grid View Items, found under **Administration\Configuration\Tree Grid Views** in the TOC.




| Name | Query Definiti... | Max Query D... | Max Child Ite... | Auto Grow on... |
|---|---|---|---|---|
| ES_Component | ES_Component | 2 | 100 | |
| ES_CrawlerErr... | ES_CrawlerErr... | 2 | 100 | |
| Part TGV | Part Query | 2 | 100 | |
| Part TGV2 | Part Query | 2 | 100 | |
| PE_BomStruc... | PE_BomStruc... | 3 | 100 | |
| PE_CAD_Reve... | PE_CAD_Reve... | 2 | 100 | |
| PE_CAD_Reve... | PE_CAD_Reve... | 2 | 100 | |
| PE_Changes | PE_Changes | 2 | 100 | |
| PE_Manufact... | PE_Manufact... | 2 | 100 | |
| PE_ManufPar... | PE_ManufPar... | 2 | 100 | |
| TGV | Part Query | 2 | 100 | |

Figure 2.

Each Tree Grid View Item is associated with a **Query Definition**, which is based on a **Context Item Type**. Once selected, you can build a grid for data display. The following are the basic steps for creating a Tree Grid View:

1. Build a Query Definition.
2. Create a Tree Grid View for viewing the data.

3. Map the data from the Query Definition to the grid.

The following sections describe how to build a Tree Grid View for the Part ItemType.

## 2.1 Creating a Query Definition

Before creating the Tree Grid View, you must first create the Query Definition. For information on how to create a Query Definition, refer to the *Query Builder Guide*. Specifically, Section 2 walks you through creating a sample Query Definition. This *Tree Grid View Administrator Guide* takes that sample Query Definition and uses it in the following procedure to build a sample Tree Grid View.

## 2.2 Building the table

Once the Query Definition is ready, build the Tree Grid View using the following procedure:

1. Create a new Tree Grid View item and specify a unique **Name** and select an existing **Query Definition** to be used.

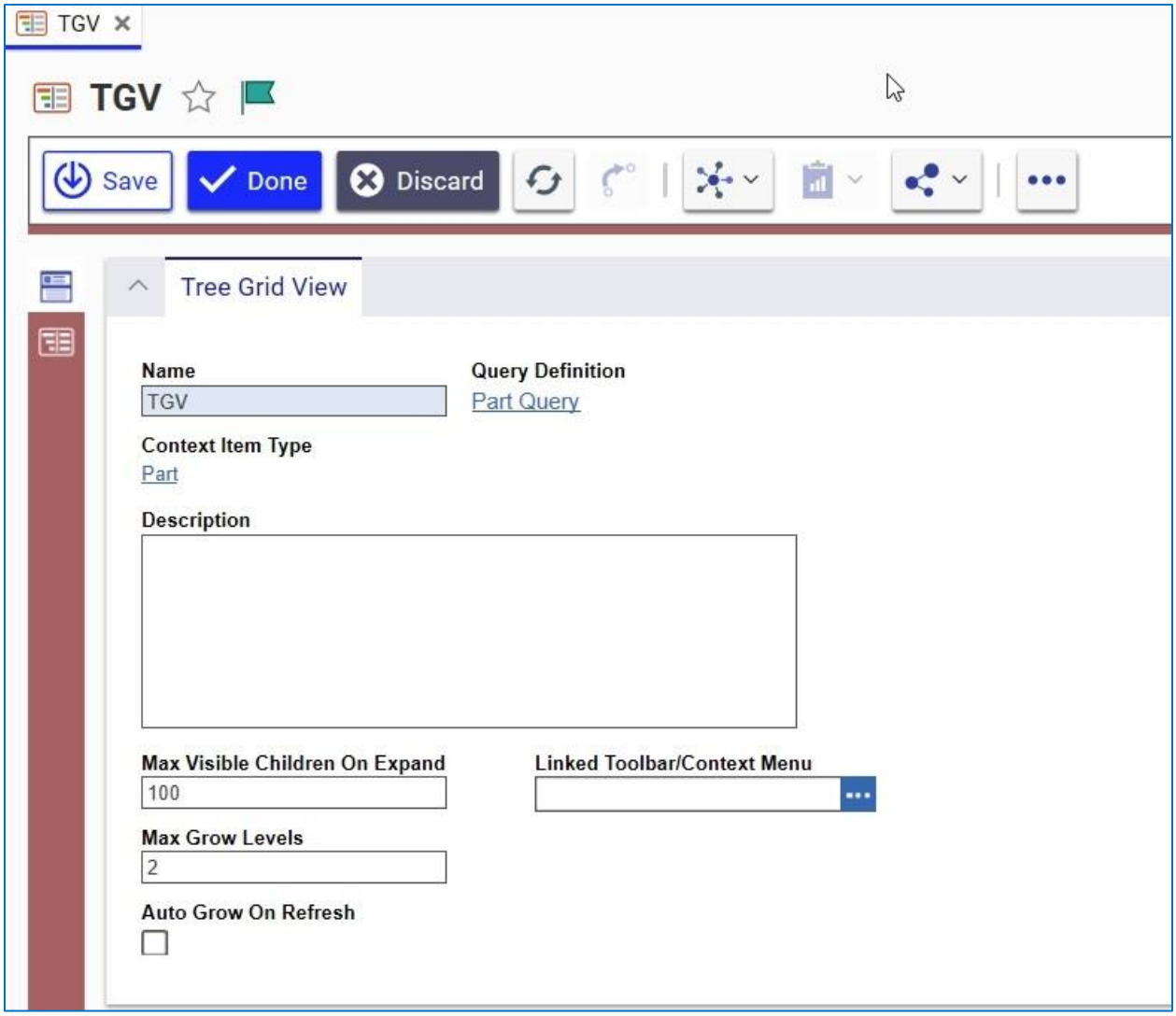



Figure 3.

| **Note:** | The **Name** of the Tree Grid View is automatically used as the name for the RelationshipType generated later. |
|---|---|

2. Select the **Auto Grow on Refresh** checkbox to keep the Tree Grid View expanded to the maximum number of grow levels when you do a refresh.
3. After saving the item, click the **Show Editor** button on the left sidebar to go into Grid-Editing mode.

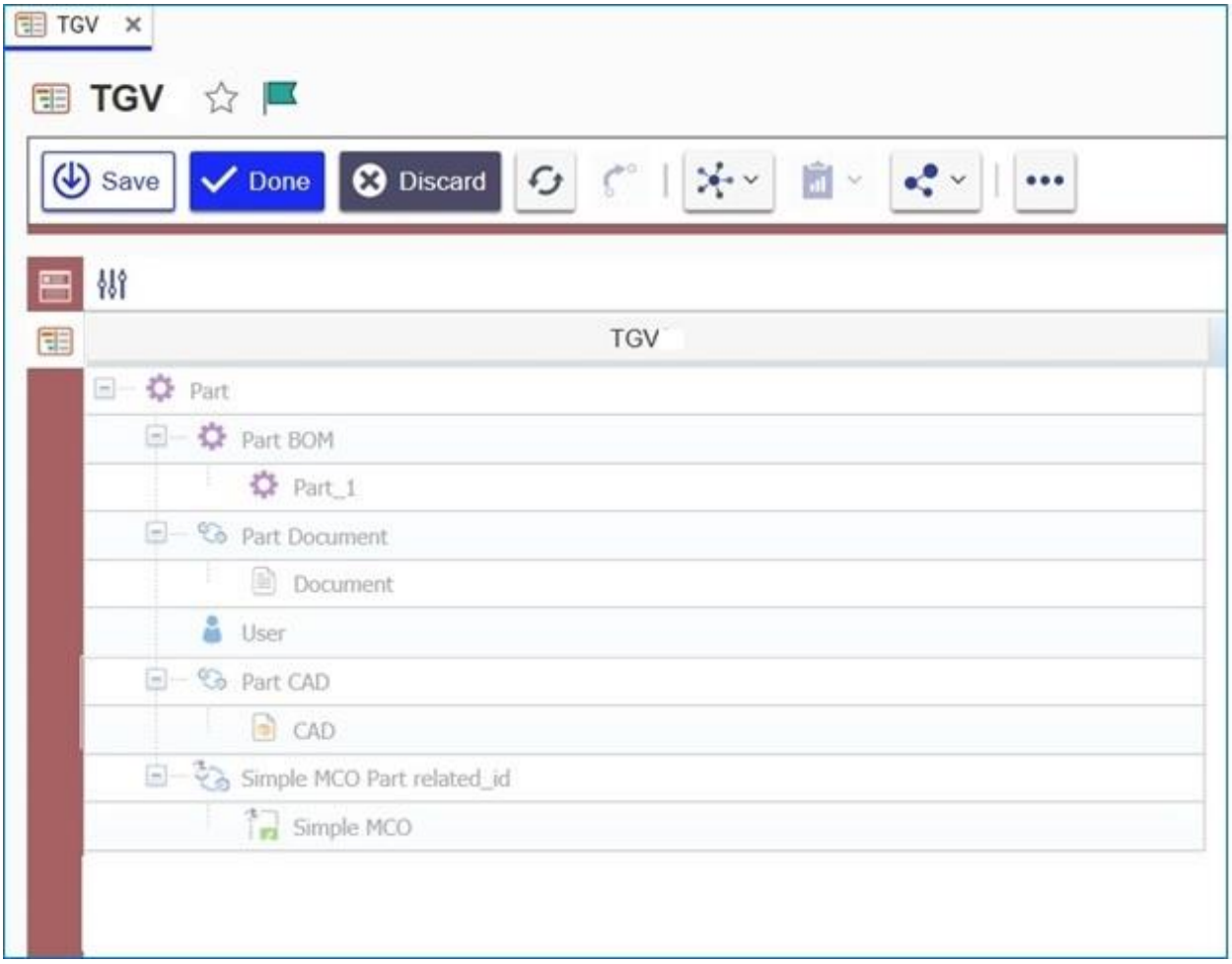



Figure 4.

4. Right-click on each element that should display data in the grid and select **Map element**.

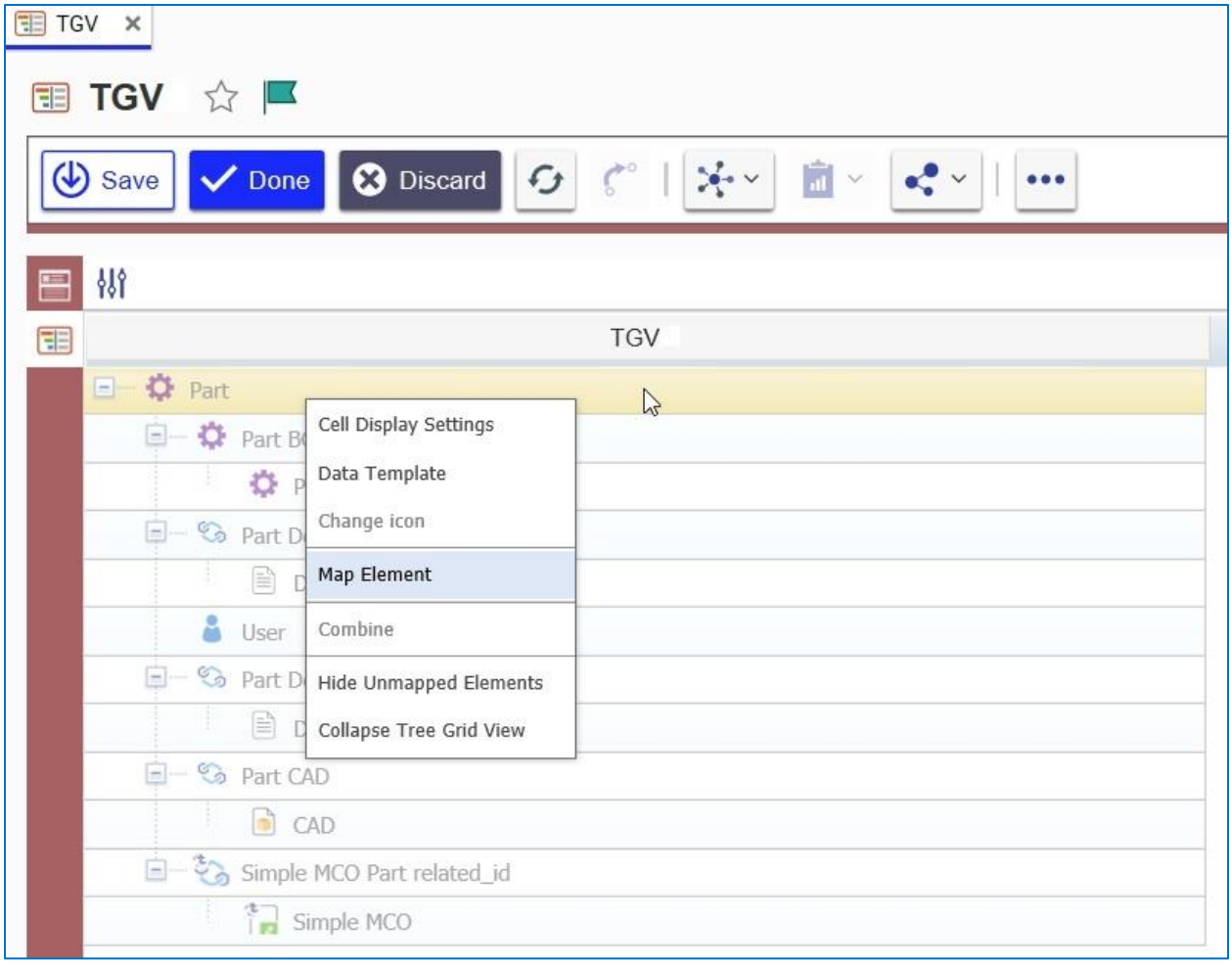



Figure 5.

5. Right-click on the column header and select **Add New Column**.

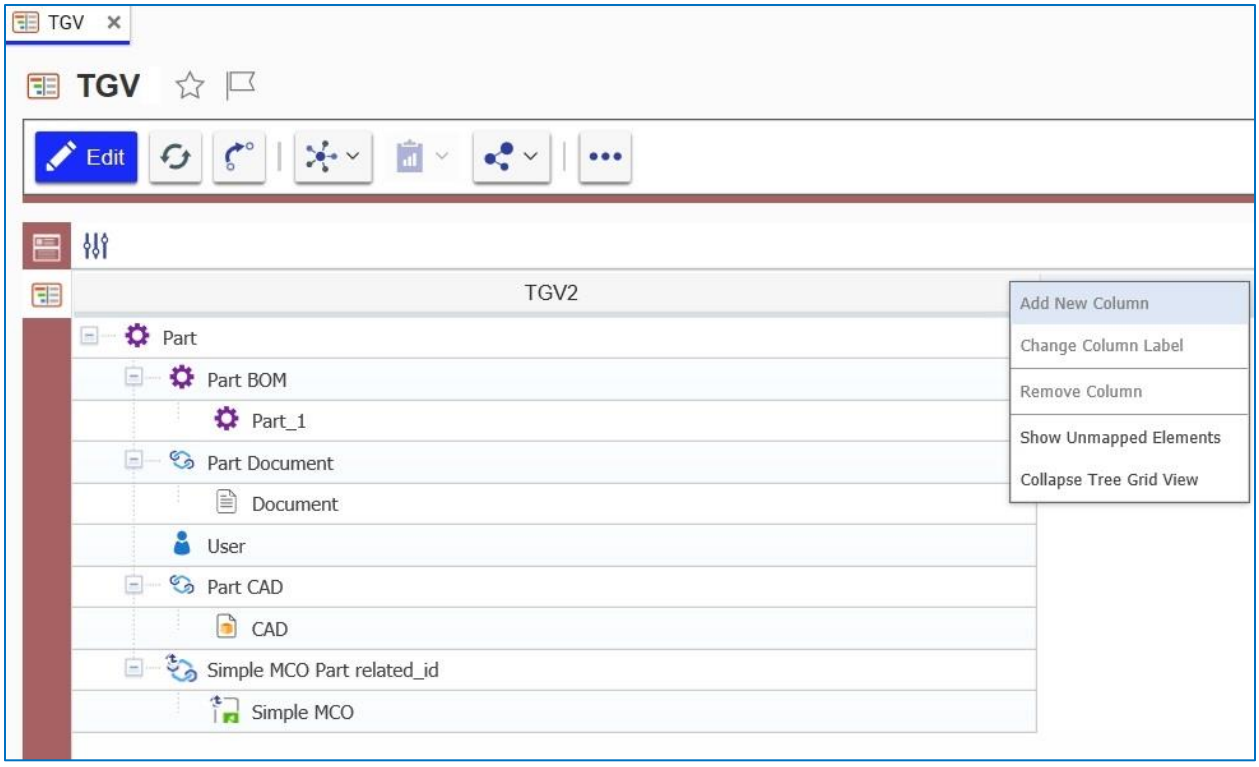



Figure 6.

6. Right-click on the new column, select **Change Column Label** and then call it **Name**.

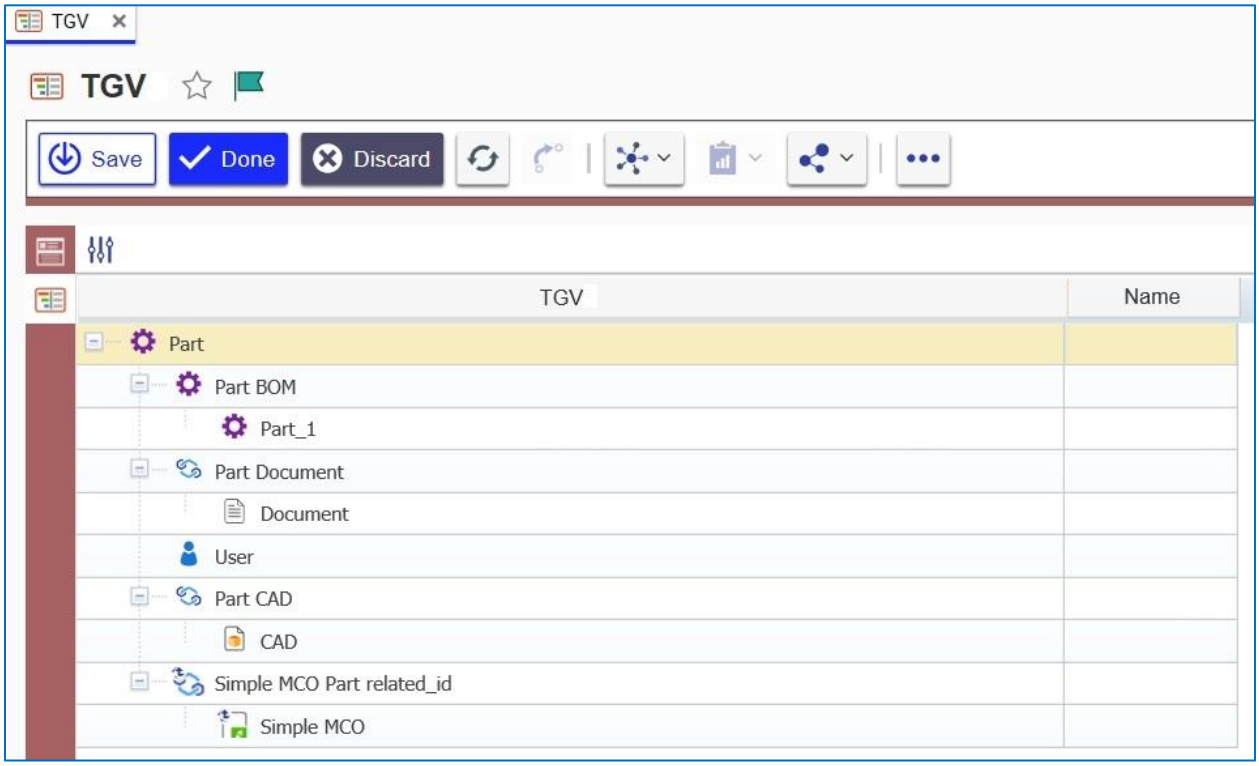



Figure 7.

7. Add 2 more columns named **State** and **Created On**.

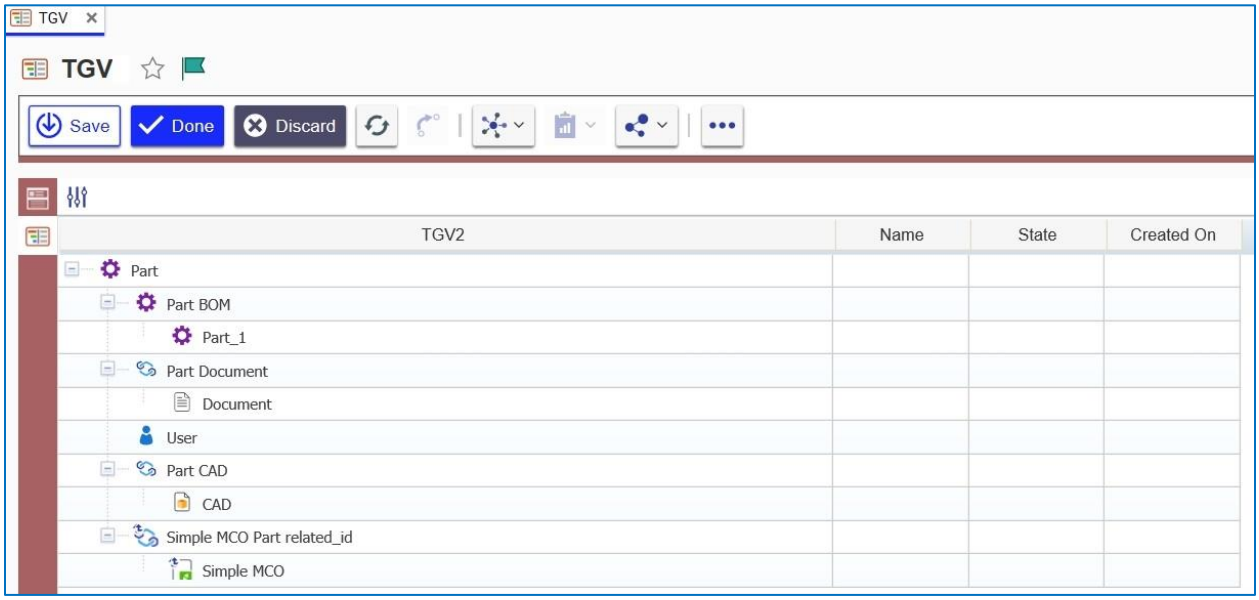



Figure 8.

8. Save the Tree Grid View.

## 2.3 Mapping the Data into the Table

The next step is to map the data from the query into the table created in the previous sections. This is done by defining the data that should go into each cell and how the data should be handled by the UI. The UI supports 8 types of data:

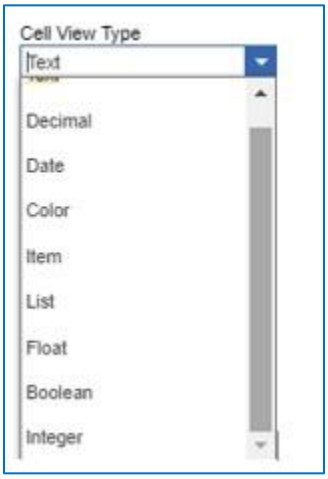



Figure 9.

| Data Type | Description |
|---|---|
| Text | Displays value as plain text |
| Decimal | Parses the decimal number to display decimal delimiter as either “.” or “,” |
| Date | Parses the data to display as a date. Supports short/long/date/time |
| Color | Displays color |
| Item | Displays a hyperlink to the specified Item |
| List | Displays List property value |
| Float | Displays Floating Point Property value |
| Boolean | Displays Boolean property value |
| Integer | Displays an  Integer Property value |

**Note:** The following property types currently cannot be mapped into the Tree Grid View definition: Float, Boolean, Image, Color List, Formatted Text, MD5

Use the following procedure to fill in the sample grid with data:

1. Double-click in the **Part-Name** cell and specify the following:
   a. Cell View Type = `Text`
   b. Text Template = `{Part.name}`

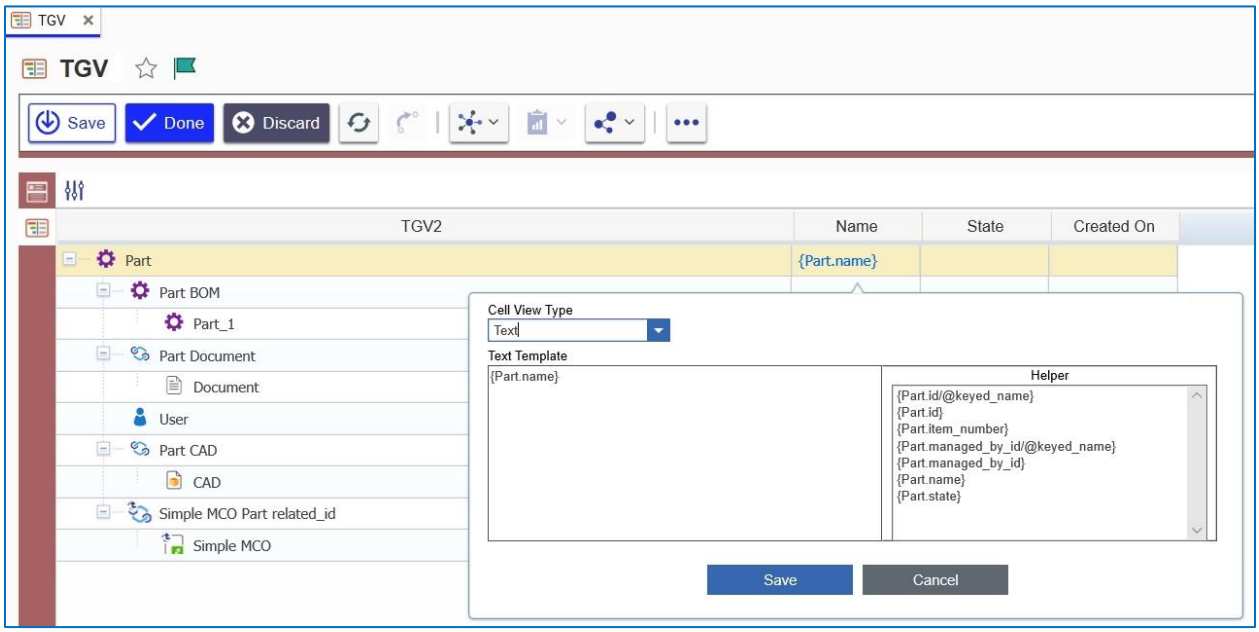



Figure 10.

**Note:** Because there is a recursive structure, the Child Part cell also gets the same mapping.

2. Double-click in the **CAD-Name** and **Document-Name** cells as well, and specify the name properties for both.

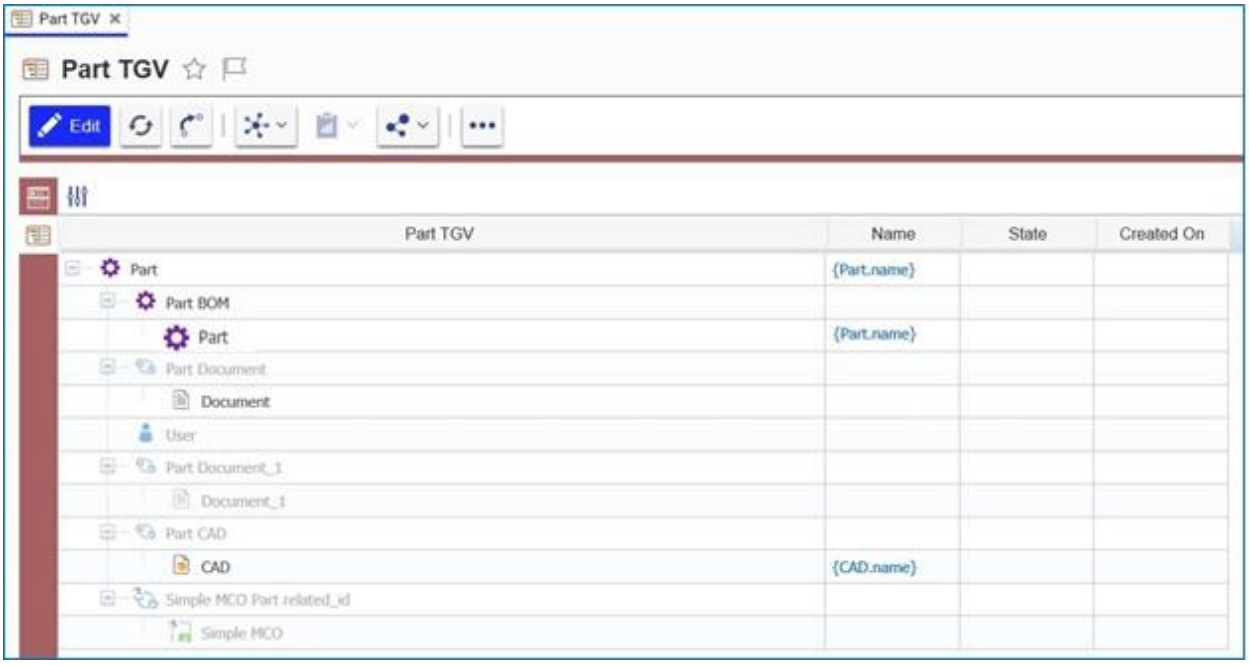



Figure 11.

3. Double-click in the **Part Created By-Name** cell and set the Text Template as `{Part Created By.first_name}{Part Created By.last_name}`.

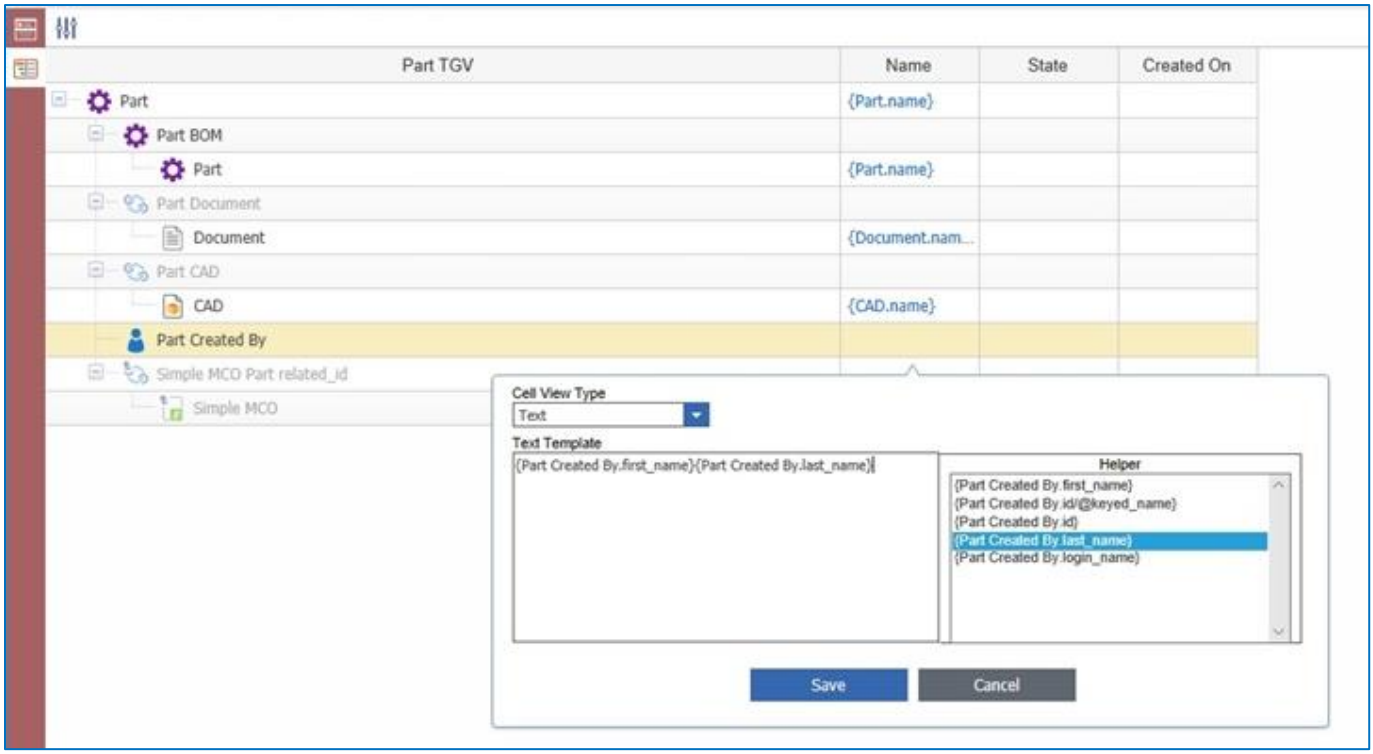



Figure 12.

4. Map the **State** properties for Part, CAD, Document, and Simple MCO to the appropriate cells.

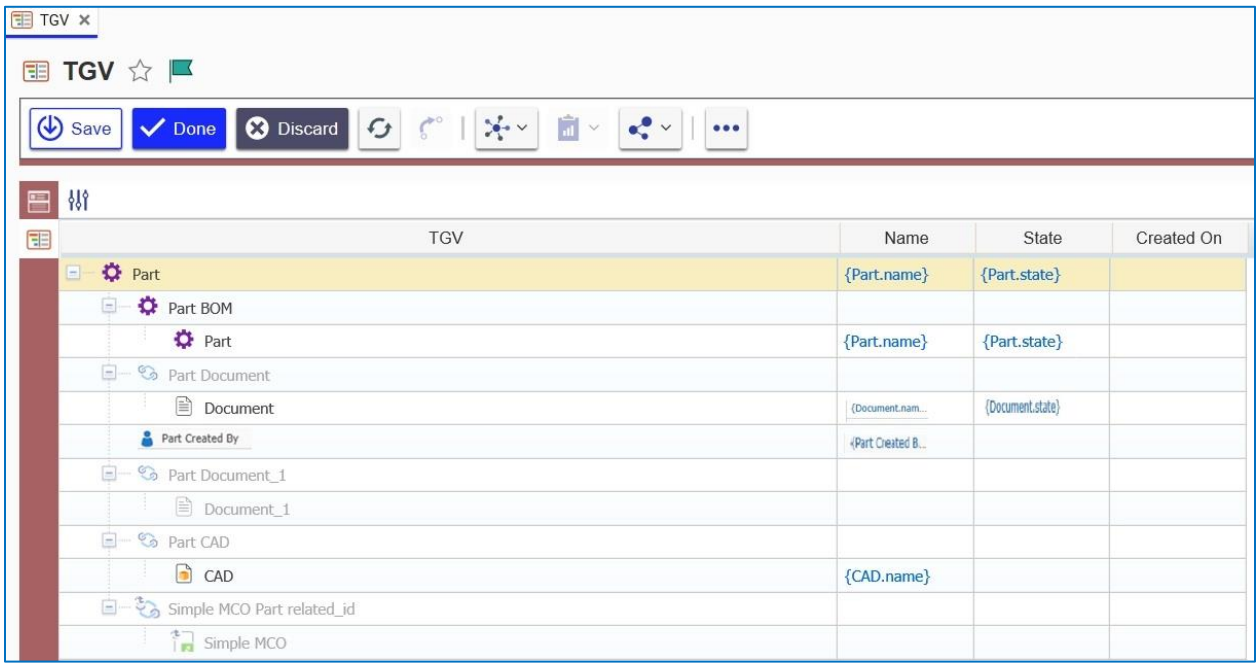



Figure 13.

5. Double-click on the **Simple MCO–Created On** cell and set the following properties:
   a. Cell View Type = `Date`
   b. Text Template = `{Simple MCO.created_on}`

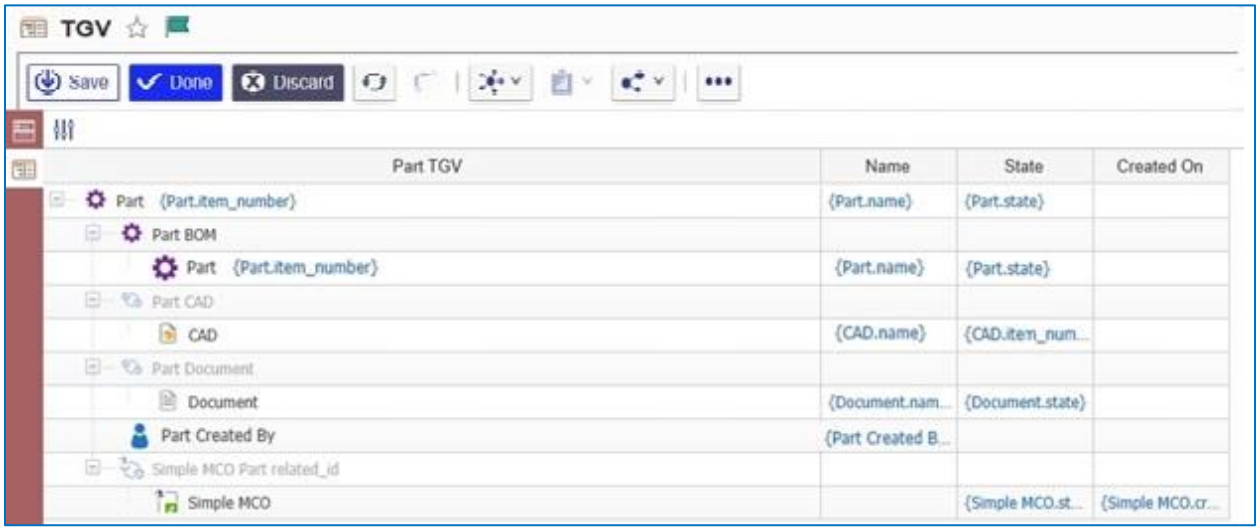



Figure 14.

6. Double-click into the top **Part-Part TGV** cell and set the following properties:
   a. Cell View Type = `Item`
   b. Innovator ItemType Name = `Part`
   c. Id Template = `{CAD.id}`
   d. Text Template = `{CAD.item_number}`

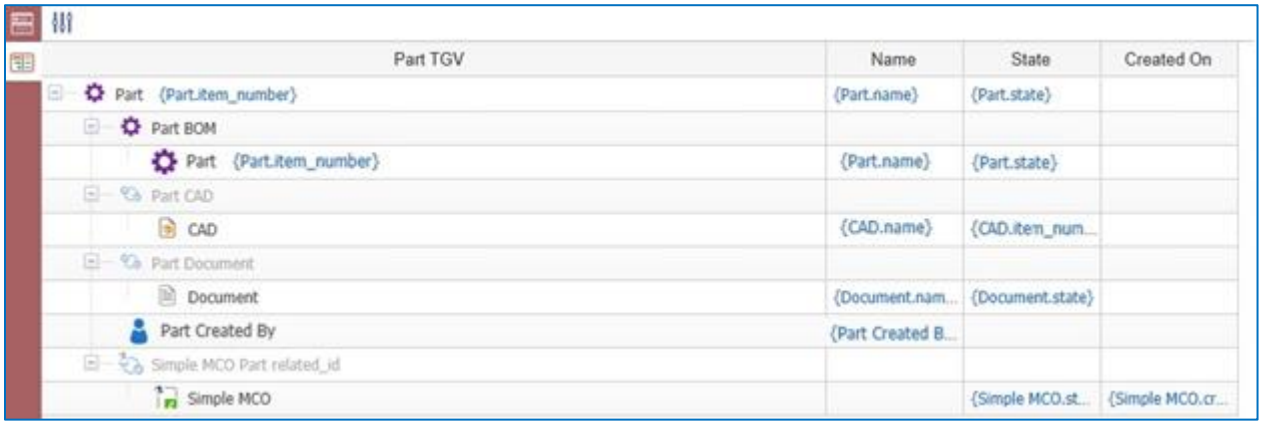



Figure 15.

7. Repeat these steps to create the same link for **CAD-Part TGV**, **Document-Part TGV**, and **Simple MCO-Part TGV** cells.

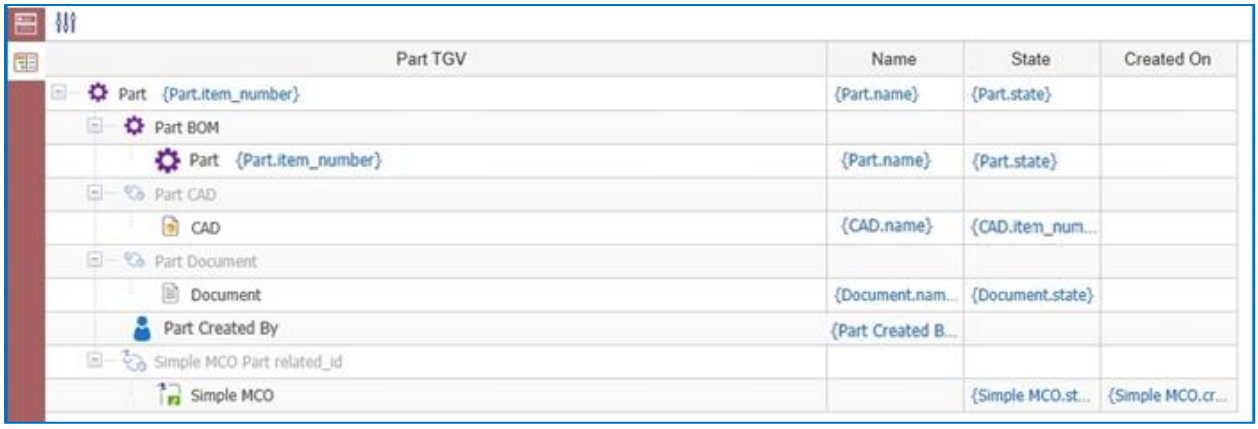



Figure 16.

**Note:** Each item needs the appropriate **Aras Innovator ItemType Name** and **Id Template** values.

8. Repeat the previous steps to create the same link for **Part Created By-Part TGV** cell, but for text display select the `login_name` property.

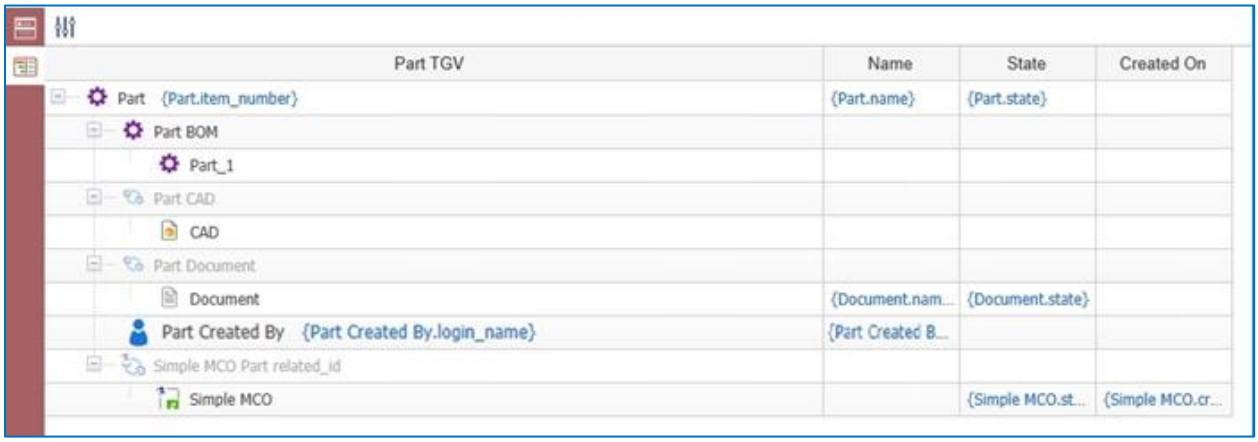



Figure 17.

9. Double-click into the **Part BOM-Part TGV** cell and set the following:
   `{BOM.related_id/@keyed_name} - Qty: {BOM.quantity}`

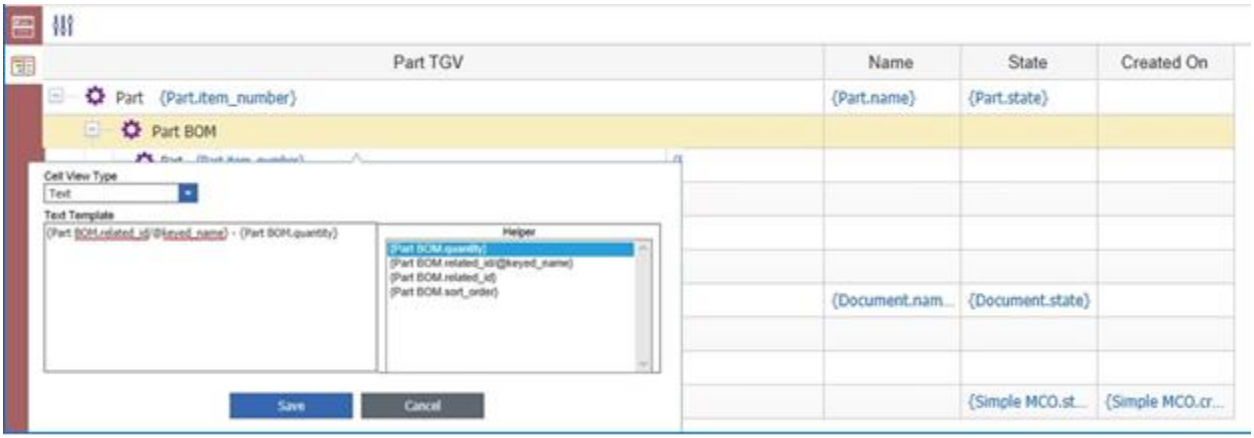



Figure 18.

10. Save the Tree Grid View.

Once you have saved the Tree Grid View, you can export the data to either an Excel file or a Word document by clicking on the appropriate icon in the toolbar. For more information about Cell View Types, refer to Section 4.9

# 3 Attaching the Tree Grid View to an ItemType

The Tree Grid View item includes an Action that automatically creates a RelationshipType and attaches it to the specified Context Item Type. The created RelationshipType automatically inherits the same name as the one used for the Tree Grid View item. The RelationshipType also inherits a custom view as defined by the table which populates based on the query defined in the associated Query Definition item.

1. To create the RelationshipType, select the Tree Grid View.
2. Click the **More** icon  and select **Set Tree Grid View Usage** from the drop-down menu.

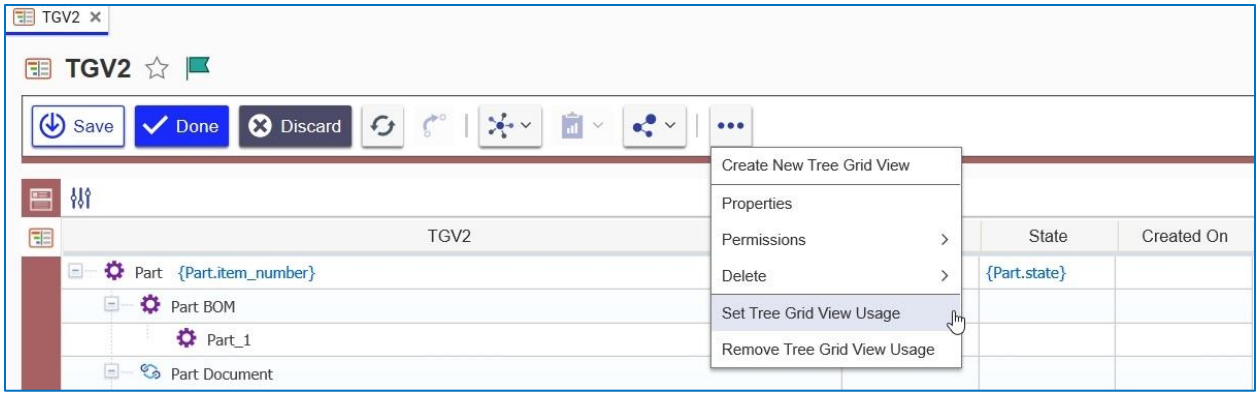



Figure 19.

Only run the action once. Running the action a second time results in an error message because the RelationshipType already exists. However, it is still possible to make changes to both the Tree Grid View and the Query Definition items.

3. Select Relationship Tab as the target usage in the Set Tree Grid Usage dialog box:

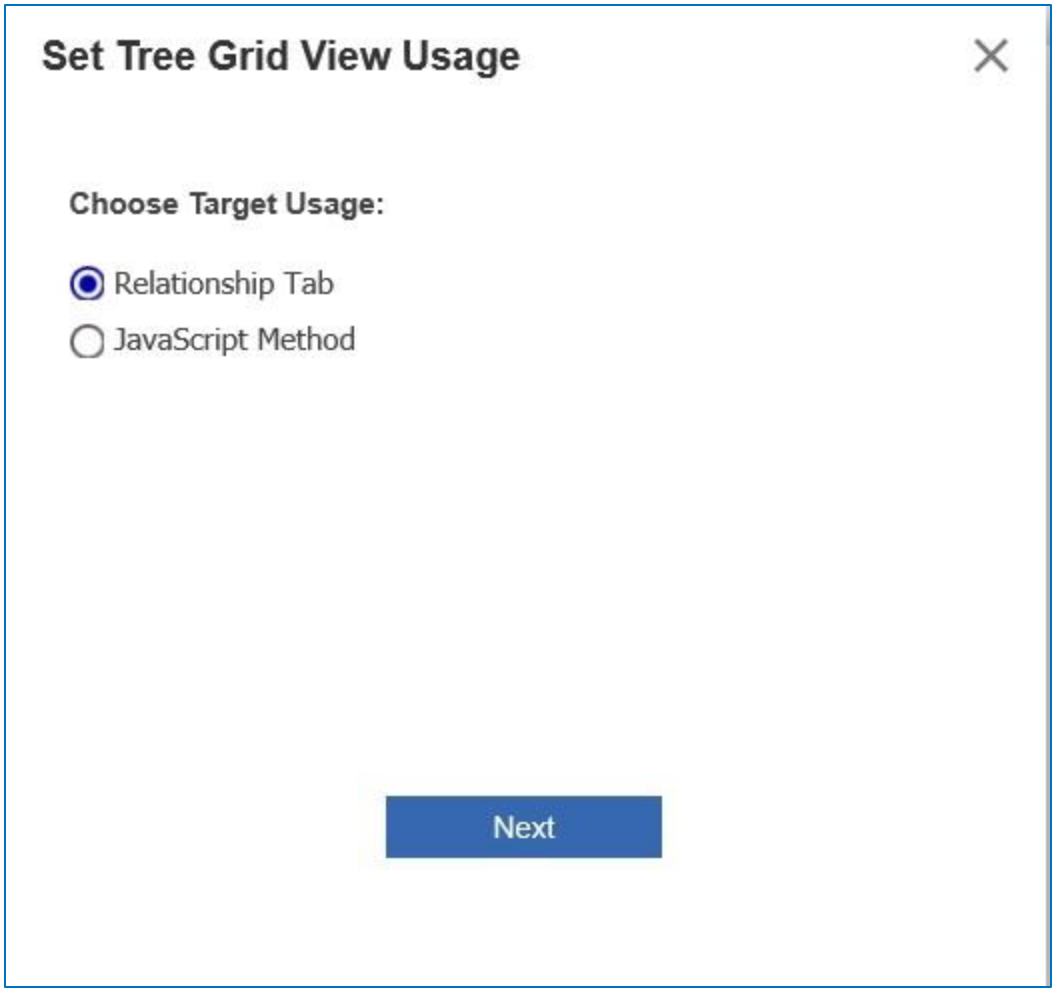



Figure 20.

4. Choose the Relationship Name:

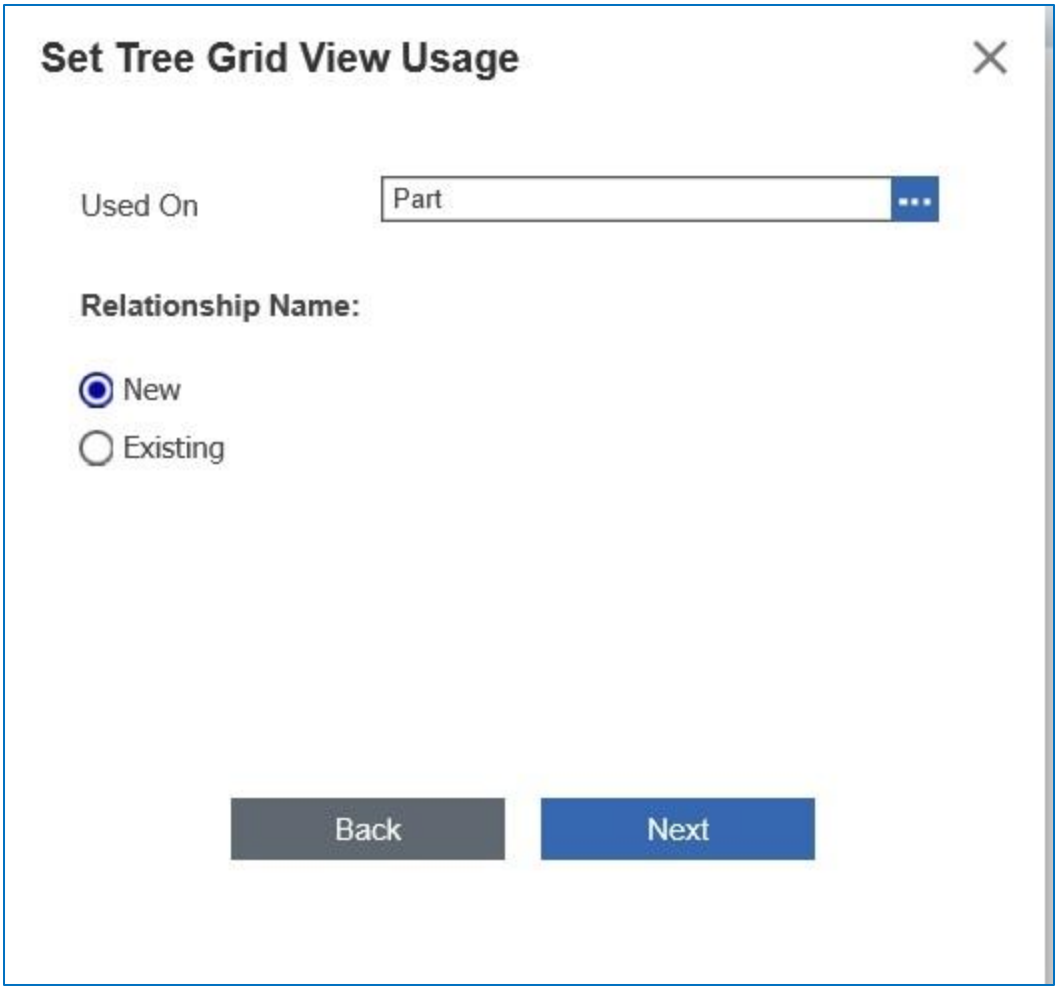



Figure 21.

5. Set the grid conditions:


Figure 22.

**Note:** The Define Starting Conditions checkbox is used to set filters on the Root Node (Part). Otherwise, all results will be returned. In this case all Parts in the database.

6. Define the Starting Conditions. Set to filter based on the Part ID.

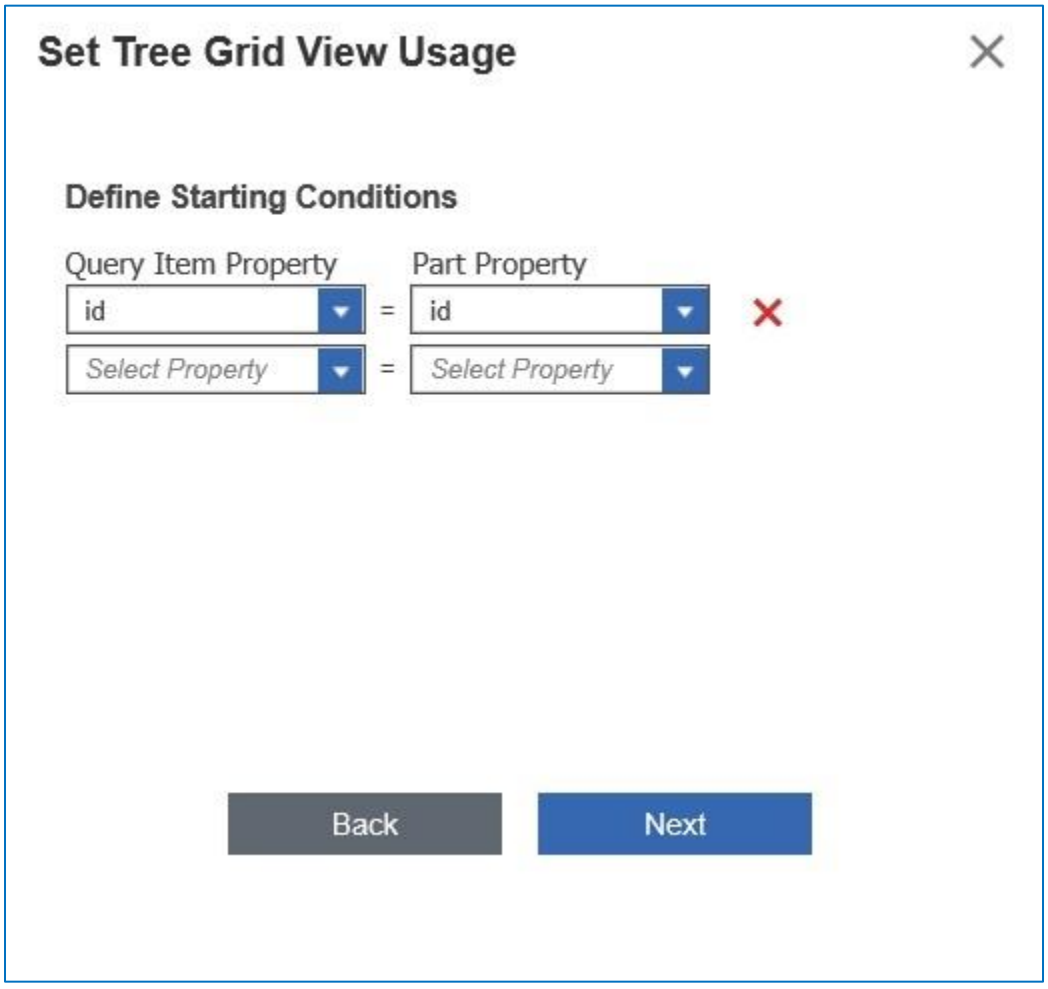



Figure 23.

7. Select **Generate** to create the view


Figure 24.

**Note:** In order to see the new tab on the Items, you may need to log out and log back in.

Activating the **Part TGV** Tree Grid View created in Section 2 Creating Tree Grid Views should result in a relationship tab that looks similar to the following. Clicking on the links should open associated item windows.

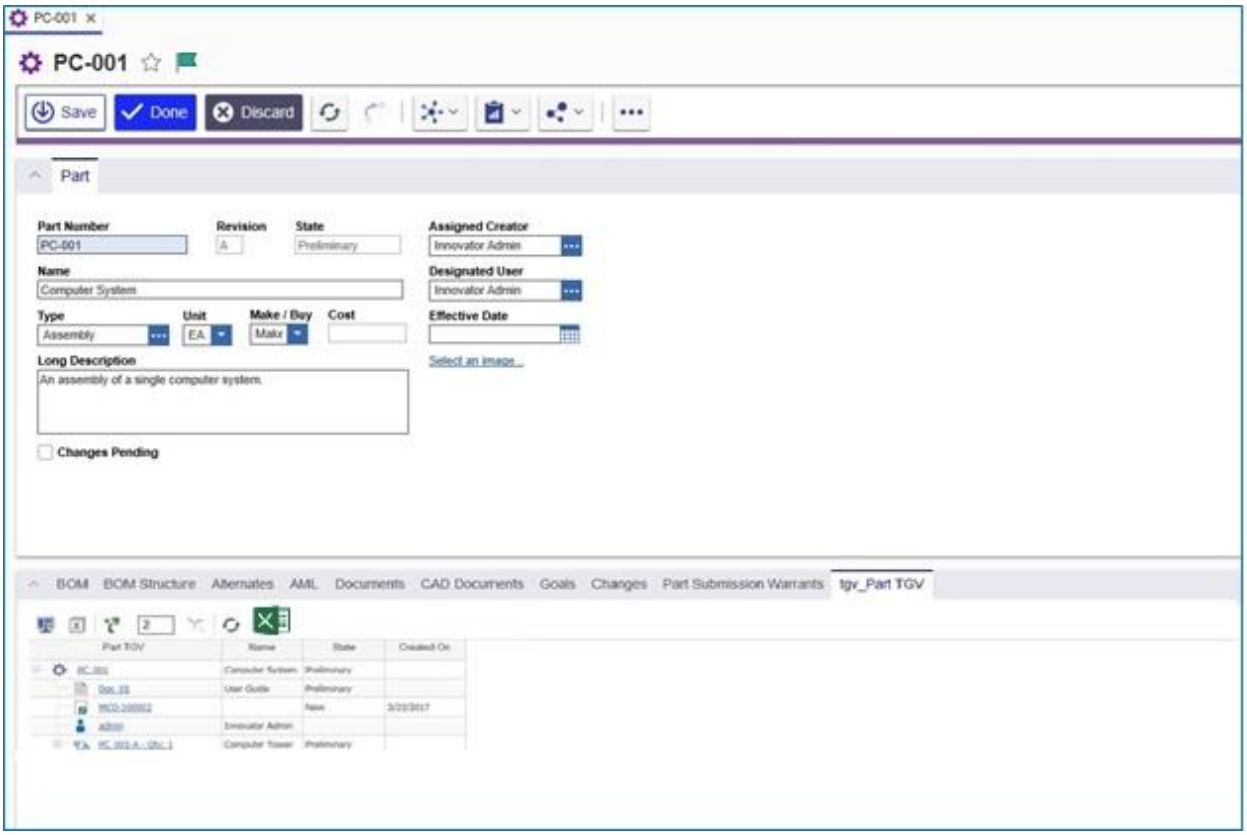



Figure 25.

7. Click the More icon and select **Remove Tree Grid View Usage** to remove a Tree Grid View.

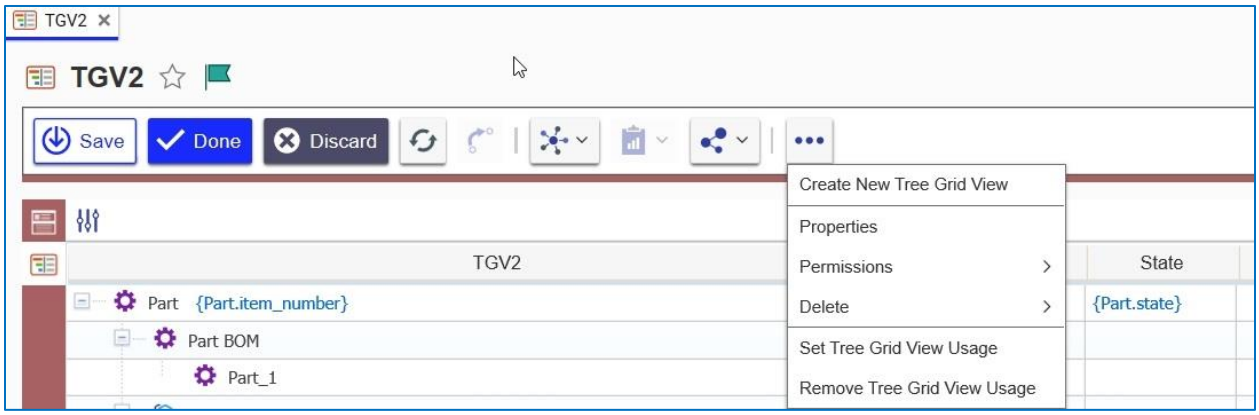



Figure 26.

8. Select the ItemTypes that should be removed from the TGV.

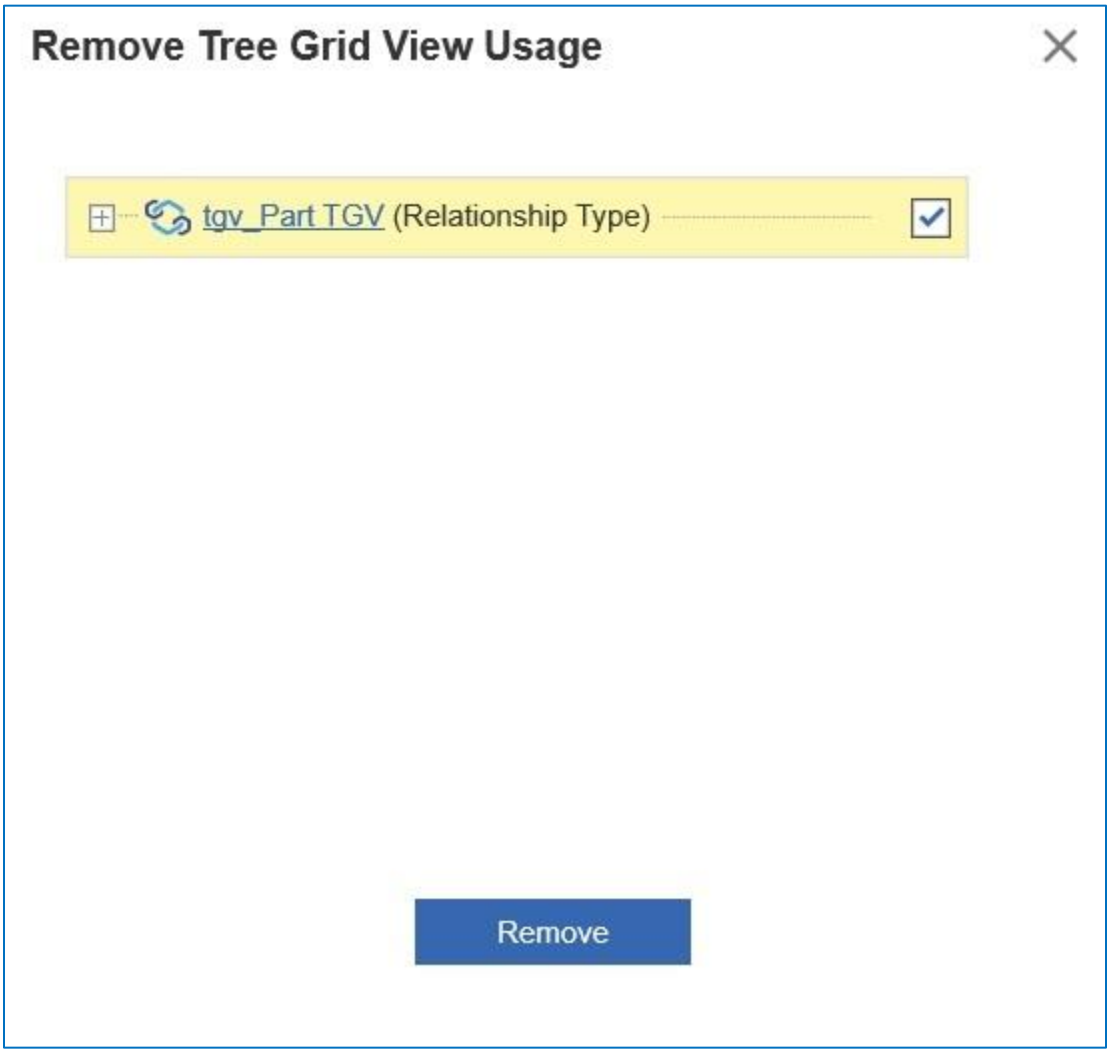



Figure 27.

# 4 Configuration Examples

This section contains examples of how to configure buttons, context menus, toolbars, and Tree Grid rows to use in Tree Grid View. The configuration items in this section are examples of advanced functionality.

**Note:** A release of sample functionality will be available for UI customization in future releases of Tree Grid View.

## 4.1 The Data Template

You can use the Data Template to select data to be used for Configurable User Interface (CUI) handlers. The following example shows the data template associated with a Part:

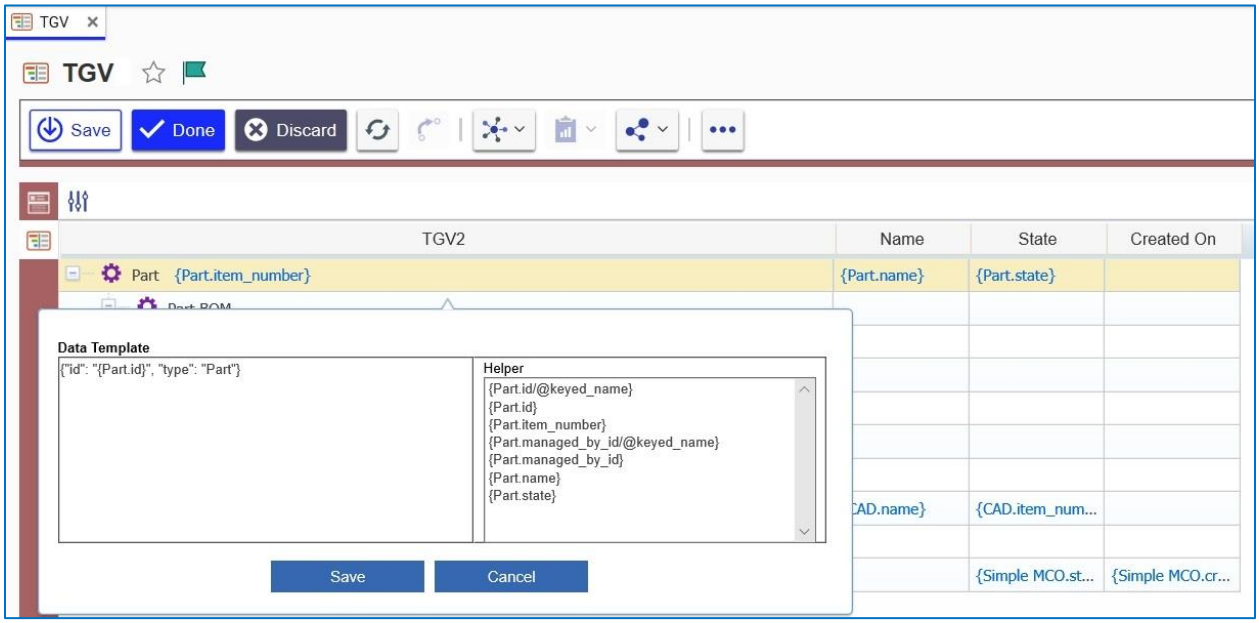



Figure 28.

## 4.2 Changing Icons

You can add or change icons in either Tree Grid View or Query Builder. In the following example, icons identify both Part BOM and User in a query:

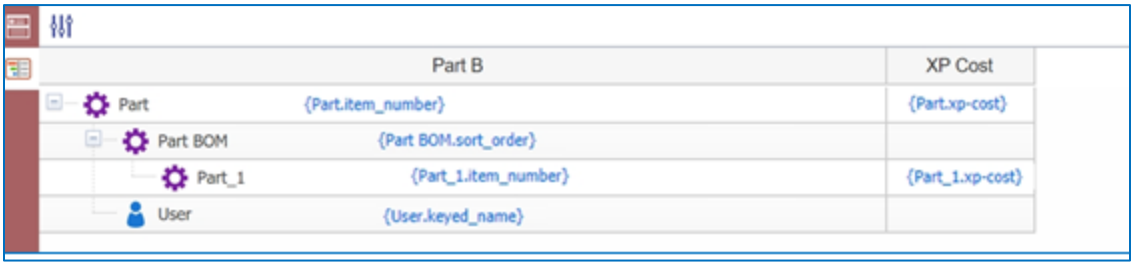



Figure 29.

You can use the Change Icon Template to change the icons that appear. Use the following procedure:

1. Right click on the first row in the grid. The context menu appears:

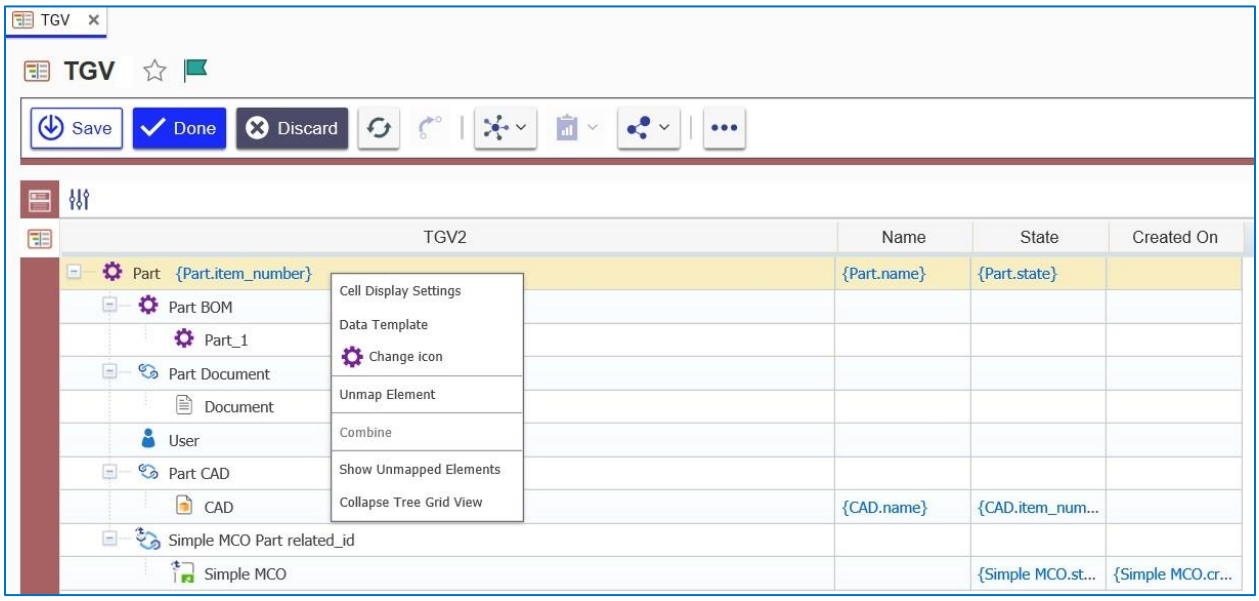



Figure 30.

2. Select **Change Icon**. The Image Browser dialog box appears.

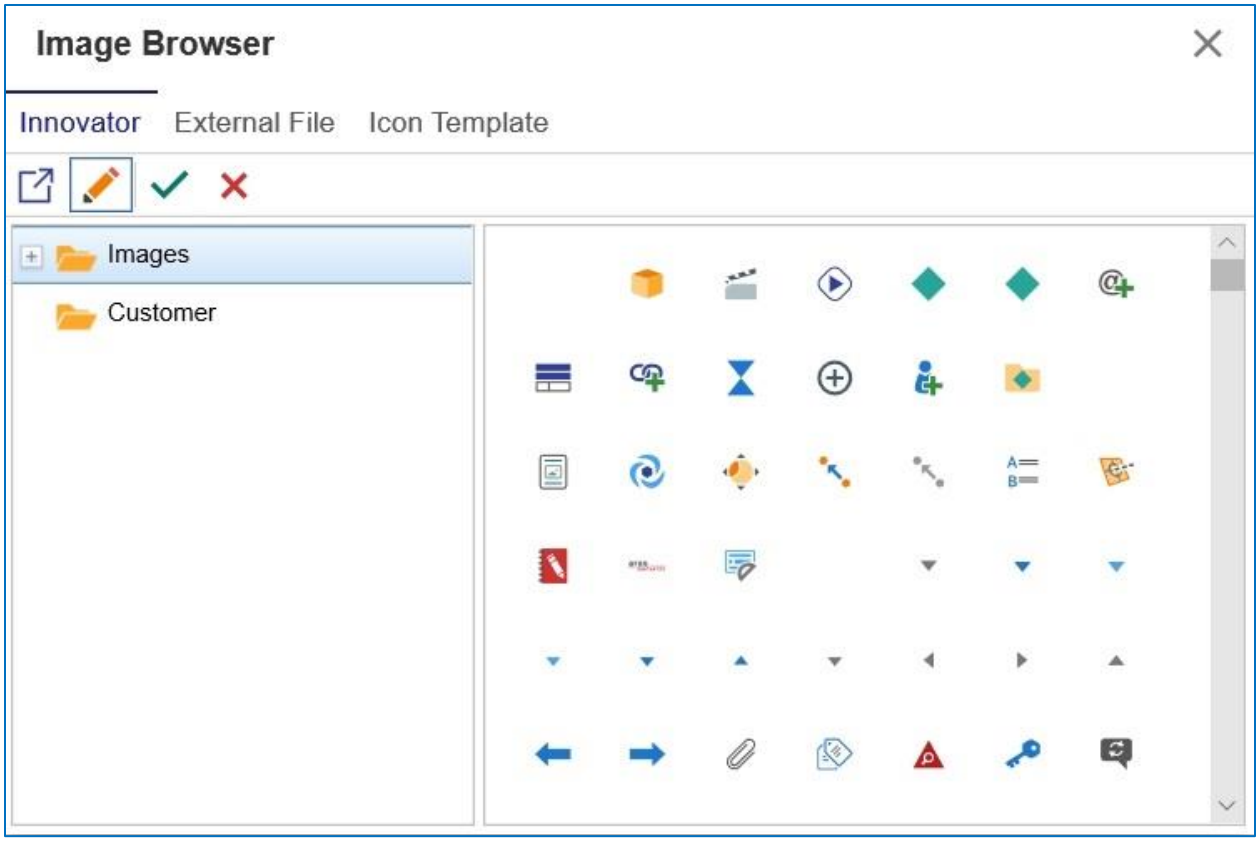



Figure 31.

3. Select **Icon Template** and select **{Part.thumbnail}** in Helper to change to the **{.}** icon.

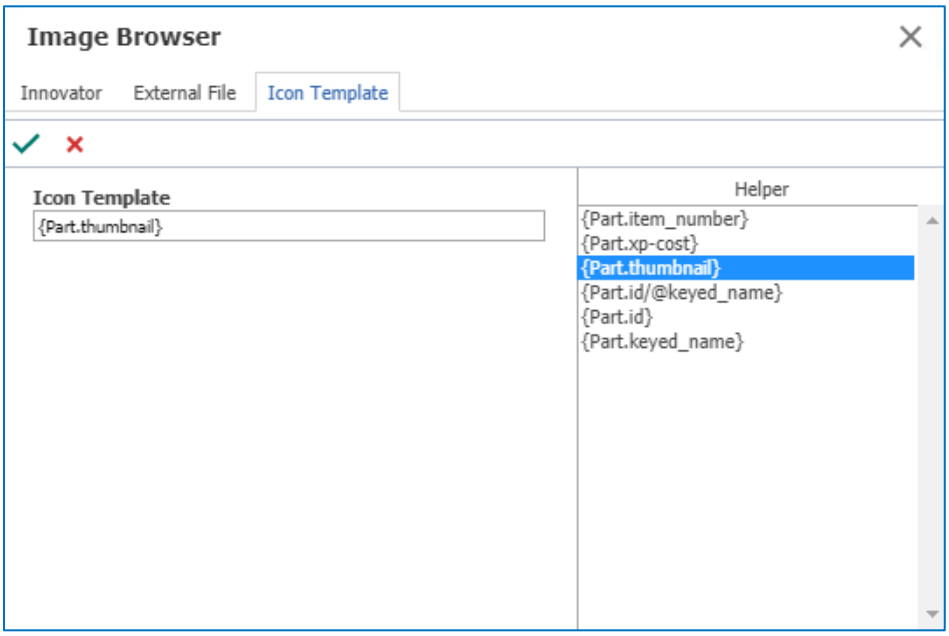



Figure 32.

4. Return to the Part Grid and right click the Part BOM row.

5. Select **Change icon** in the context menu. The Image Browser dialog appears.
6. Select **Innovator** and select an icon.

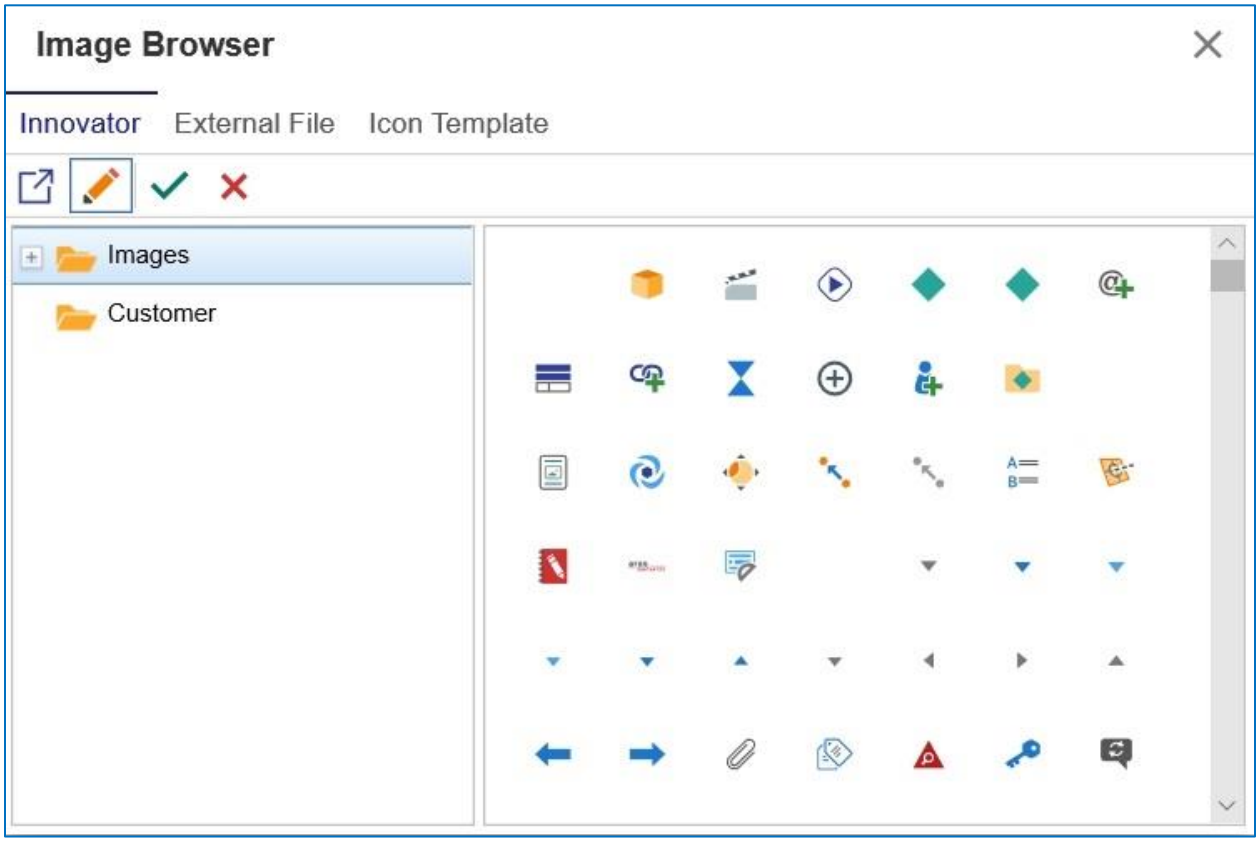



Figure 33.

## 4.3 Configuring a Toolbar Button

The following procedure is an example of how to create a configurable button:

1. Select the **PresentationConfiguration** item type from the grid, click the **TOC Access** tab, and add Administrators:

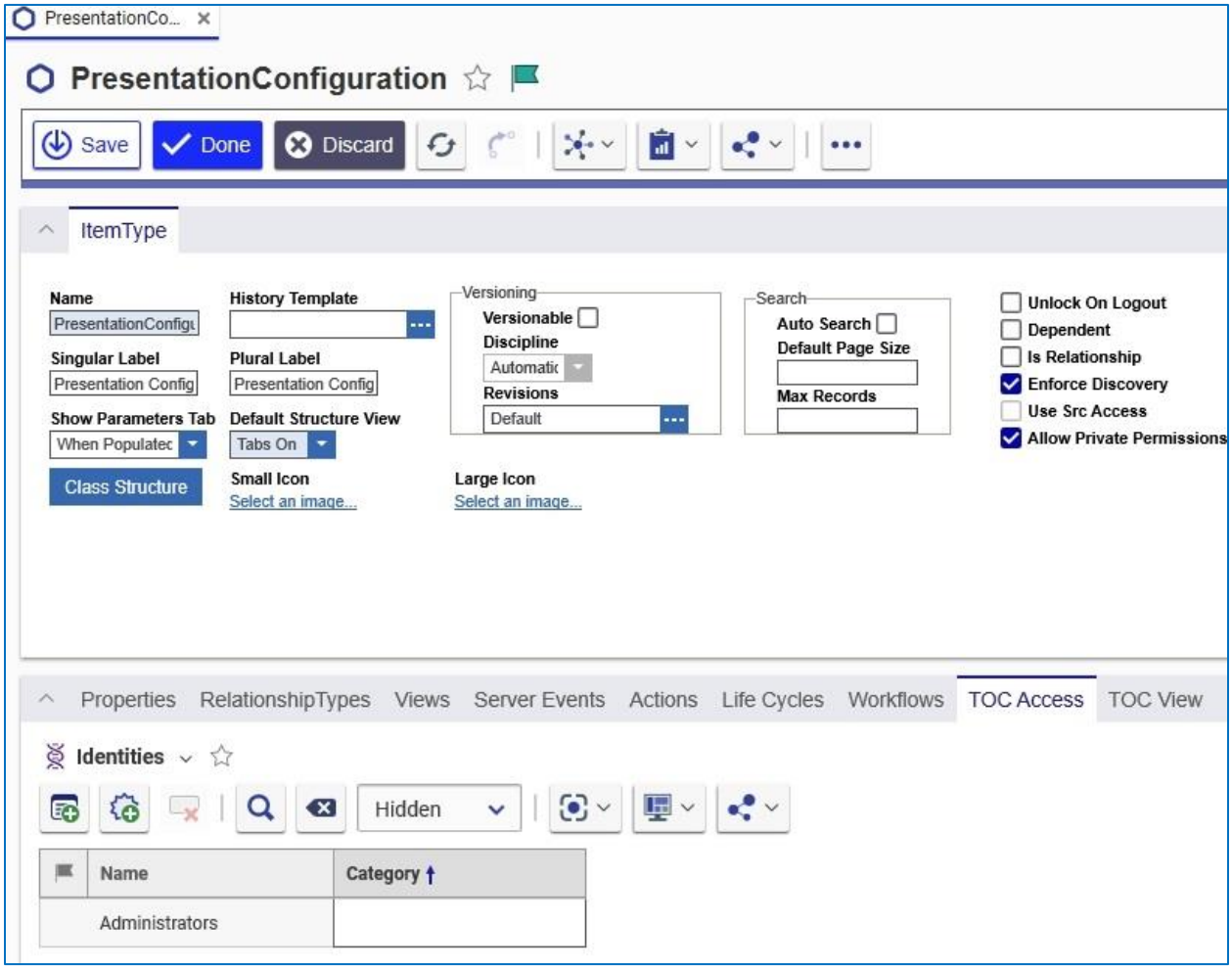



Figure 34.

2. Open the TOC and select **Presentation Configuration**. The following menu appears:

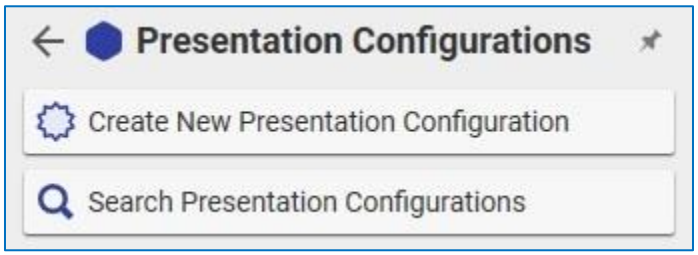



Figure 35.

3. Click **Create New Presentation Configuration**. The Presentation Configuration form appears.
4. Enter **tgv_part** in the Name field and select a color such as orange.

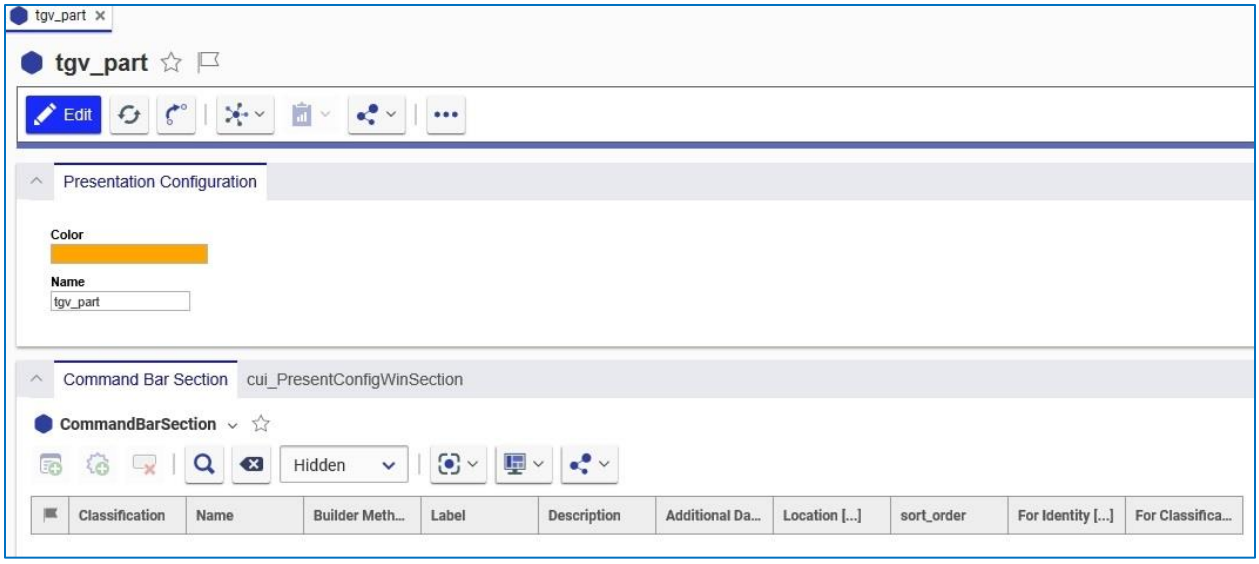



Figure 36.

5. Click the new command bar icon  in the Command Bar Section relationship tab to add a new related item.
6. Enter “tgv*” in the Name cell of the Search grid and add the tgv_part_toolbar and tgv_part_ContextMenu items:

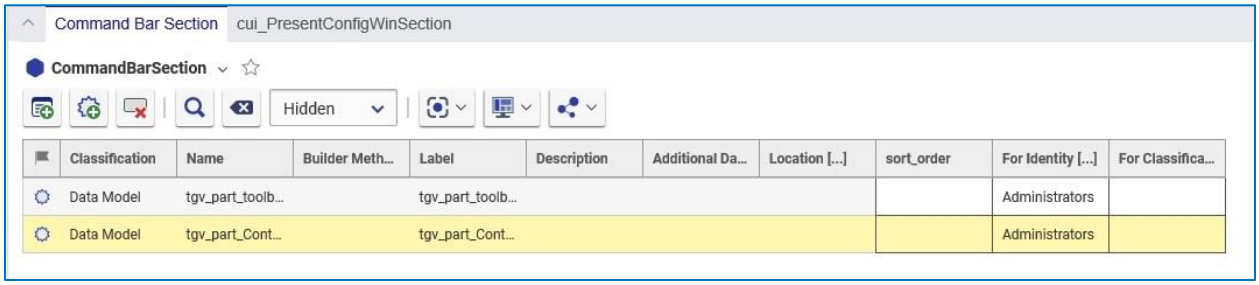



Figure 37.

7. Make sure to add the items to the relationship grid.

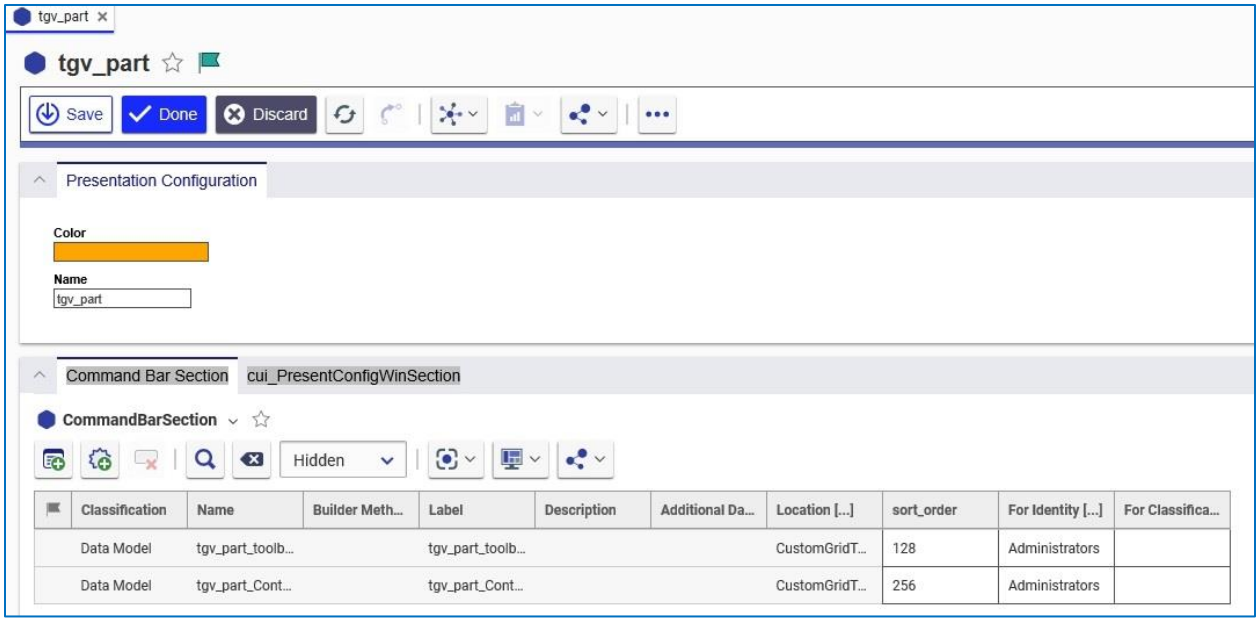



Figure 38.

8. Add an identity for the “For Identity” column. The specified Identity determines which users have access to the buttons/menus.
9. Click the 

button to save the tgv_part Presentation Configuration item.
10. Right click **tgv_part_toolbar** and select **Open** from the context menu:

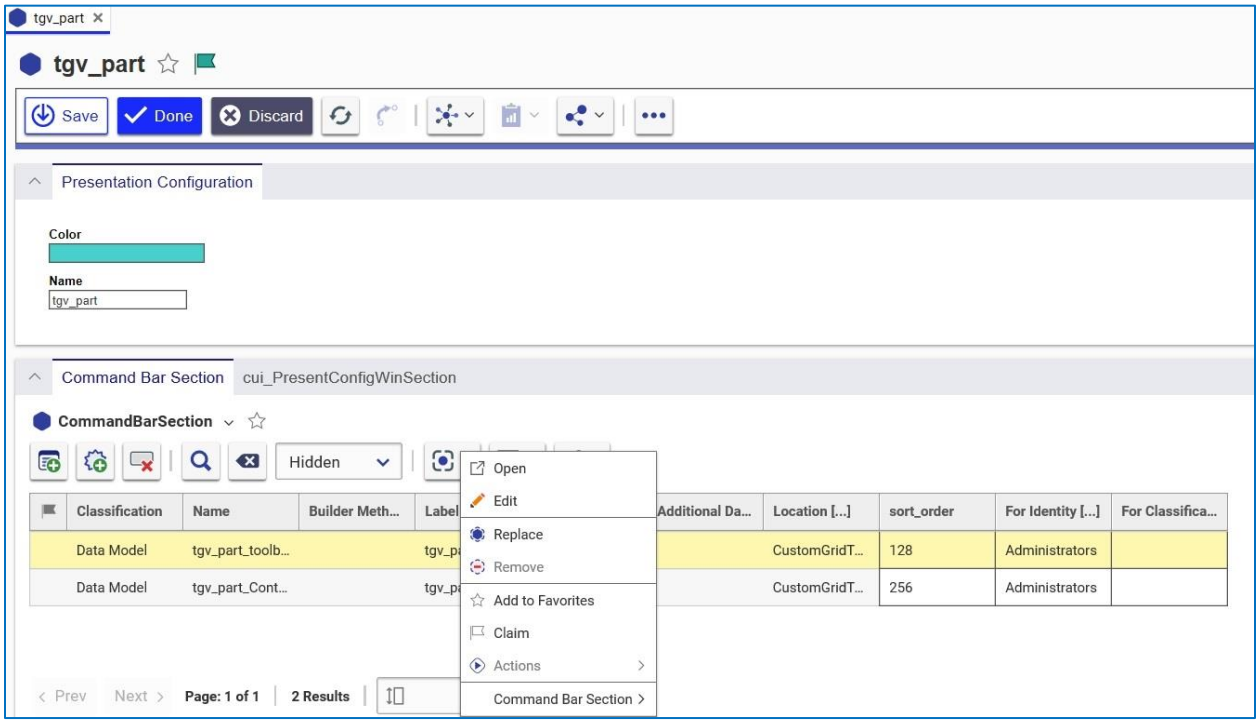



Figure 39.

The tgv_part_toolbar presentation configuration item appears. It contains the tgv_view button:

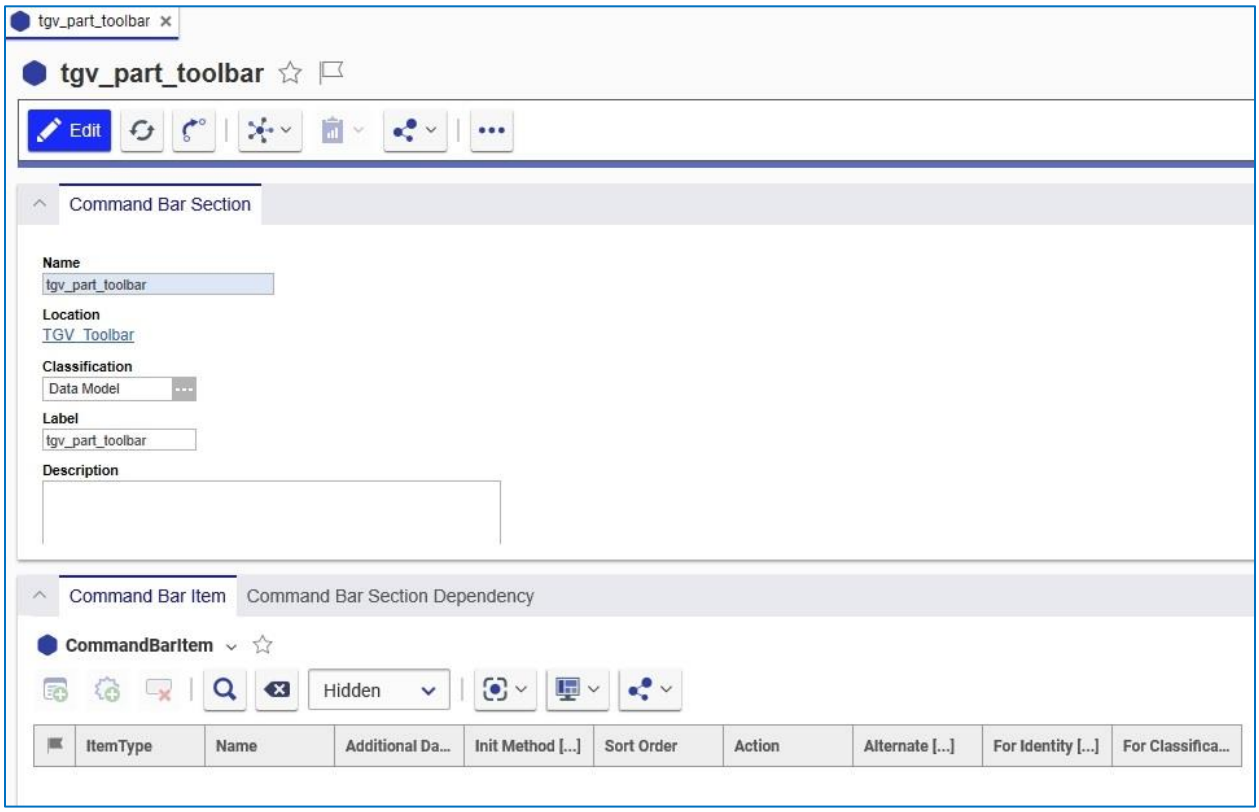



Figure 40.

11. Click the **New Command Bar Item** icon  in the Command Bar Item tab. The Select Item Type dialog appears:

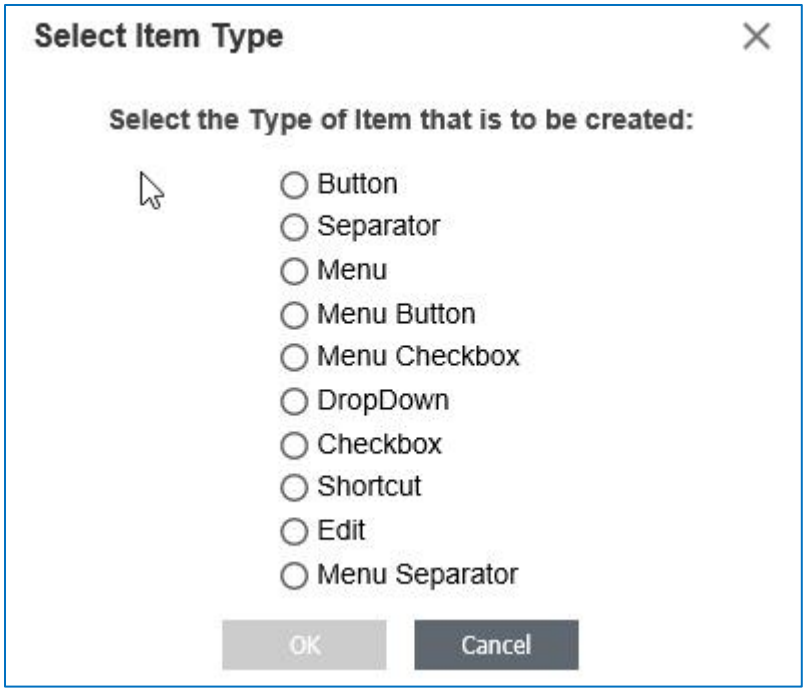



Figure 41.

12. Select **Button** and click **OK** to add it to the command bar.
13. Right click on the Item to access the context menu and select **Open**. The tgv_view presentation item appears:

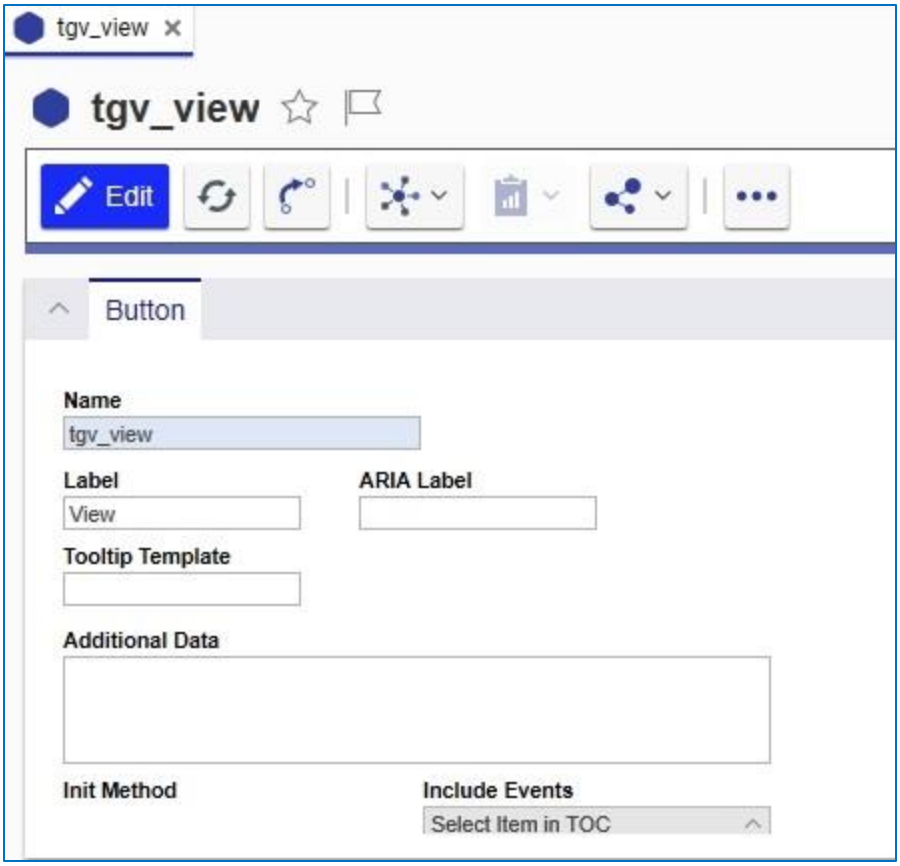



Figure 42.

Additional logic can be added to the context menu or toolbar items when they are initialized by adding an Init Method. Users can review the rb_viewItemHandler method code and develop similar code for their own uses.

### 4.3.1 Adding an Image to a Button

You can select an image for the toolbar button you created using the following procedure:

1. Select the Image button and click the **Select an image** link. The Image Browser dialog appears.

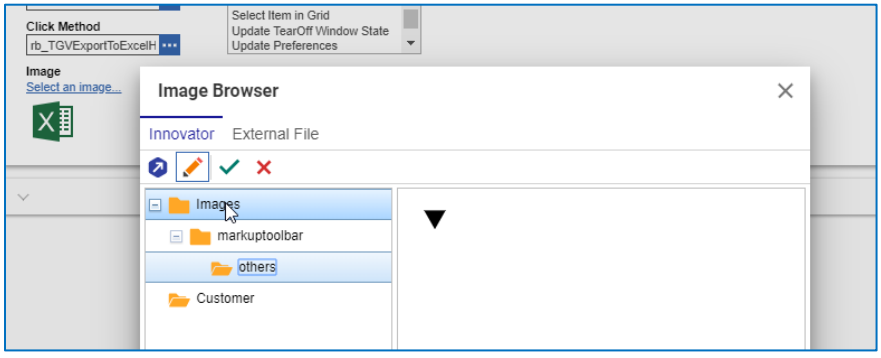



Figure 43.

2. Select the appropriate image and click the ✓ icon.

3. Click 

to save the Presentation Configuration.

### 4.3.2 Linking a Presentation Configuration to a TGV Definition

Once you save the Presentation Configuration you need to link it to a Tree Grid View definition before you can use it. Use the following procedure:

1. Select **Administration>Configuration>Tree Grid Views** and click the search icon. A list of Tree Grid views appears in the search grid.
2. Select **Part TGV**.

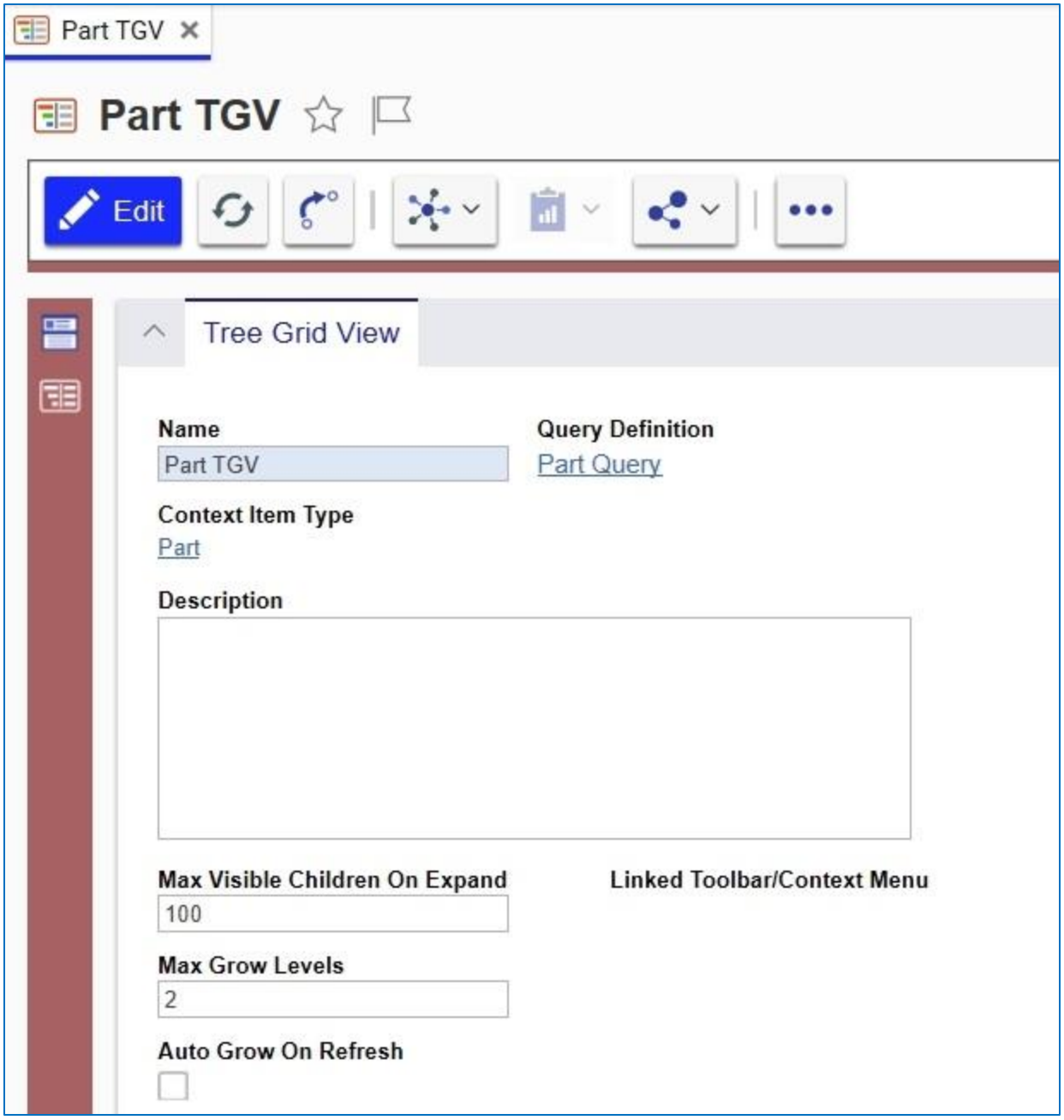



Figure 44.

**Note:** Make sure that the item is locked before continuing.

3. Click the ellipses in the Linked Toolbar/Context Menu field. The Presentation Configuration search dialog appears:

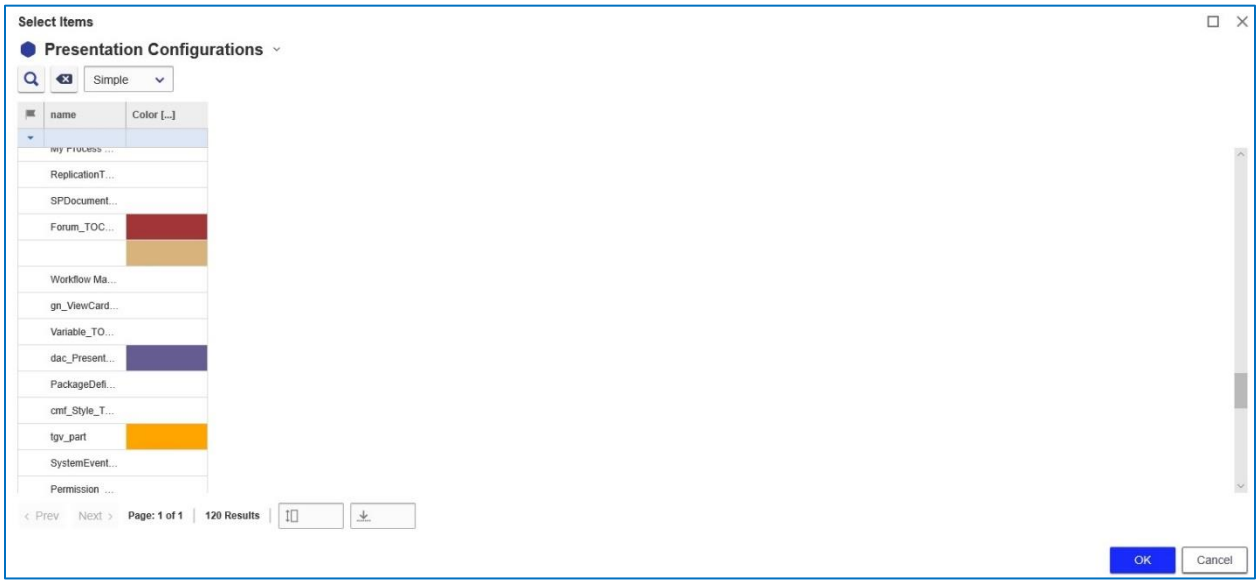



Figure 45.

4. Select tgv_part and click 

.

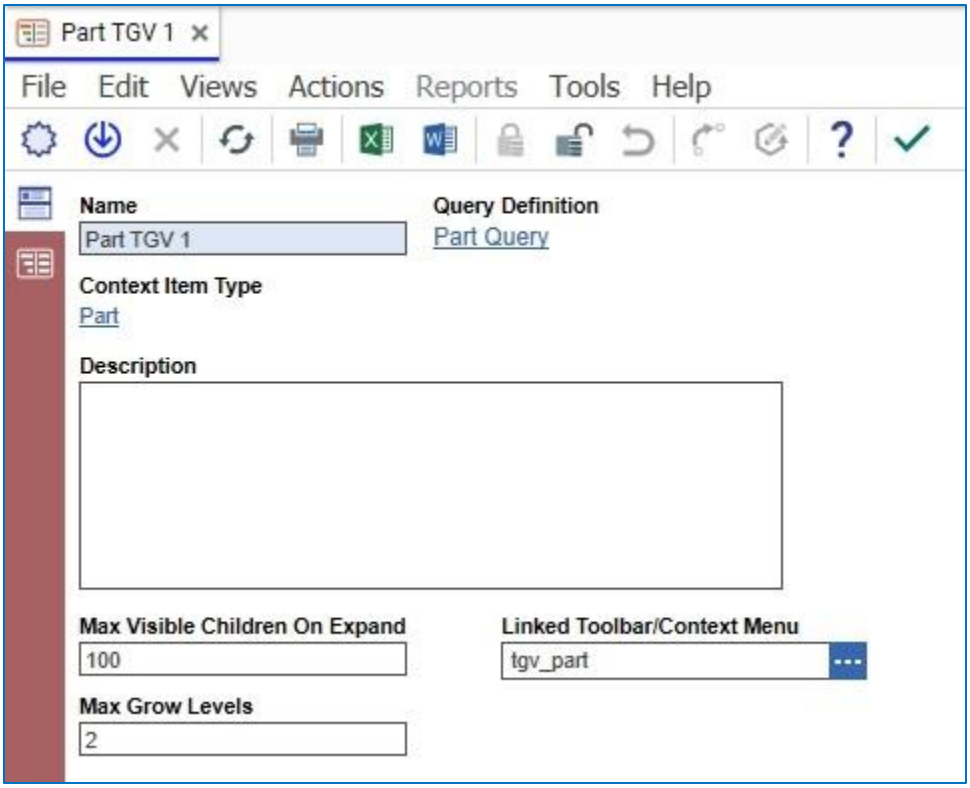



Figure 46.

5. Save the Tree Grid View.

Load an item configured for TGV to see the new toolbar button. For example, the following screenshot shows a Part item

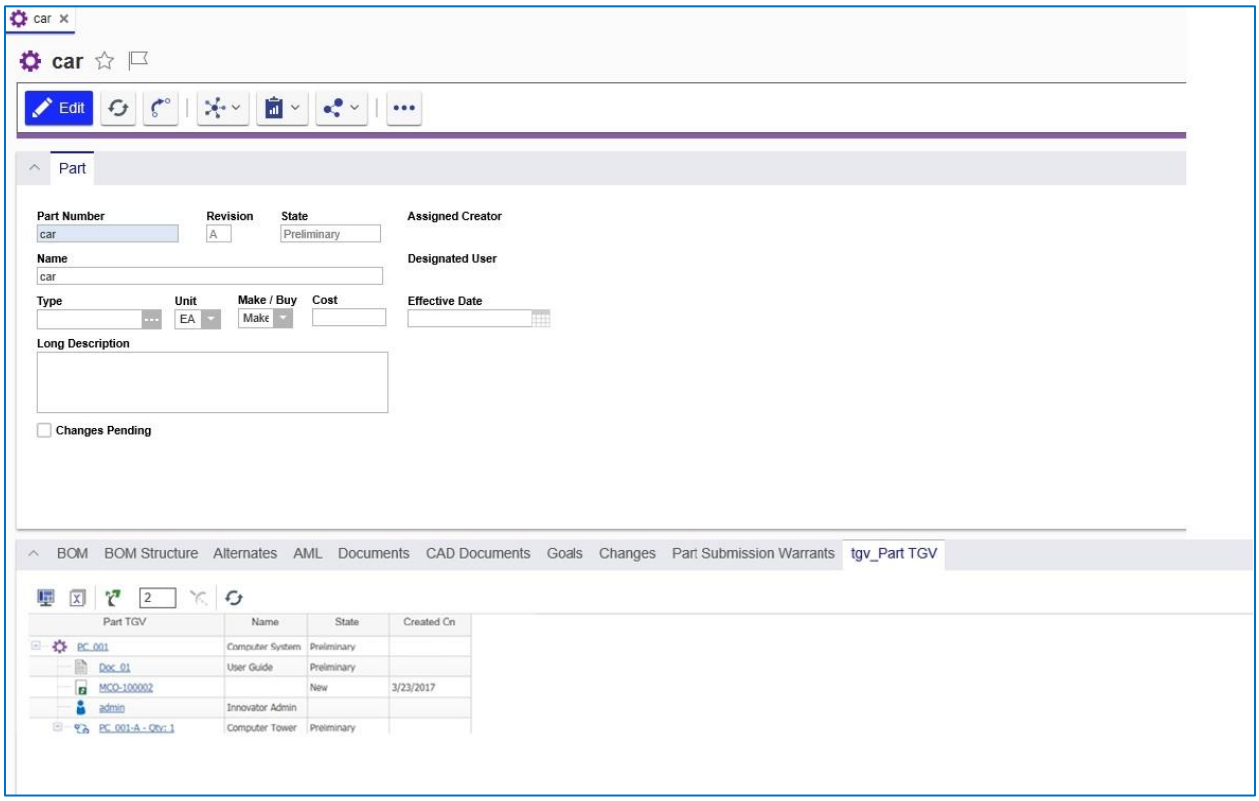



Figure 47.

## 4.4 Configuring a Context Menu

The following is an example of how to configure a context menu using custom code:

1. Select the tgv_part Presentation Configuration item, right click **tgv_part_context** in the Command Bar section grid and select **Open** from the Context menu.

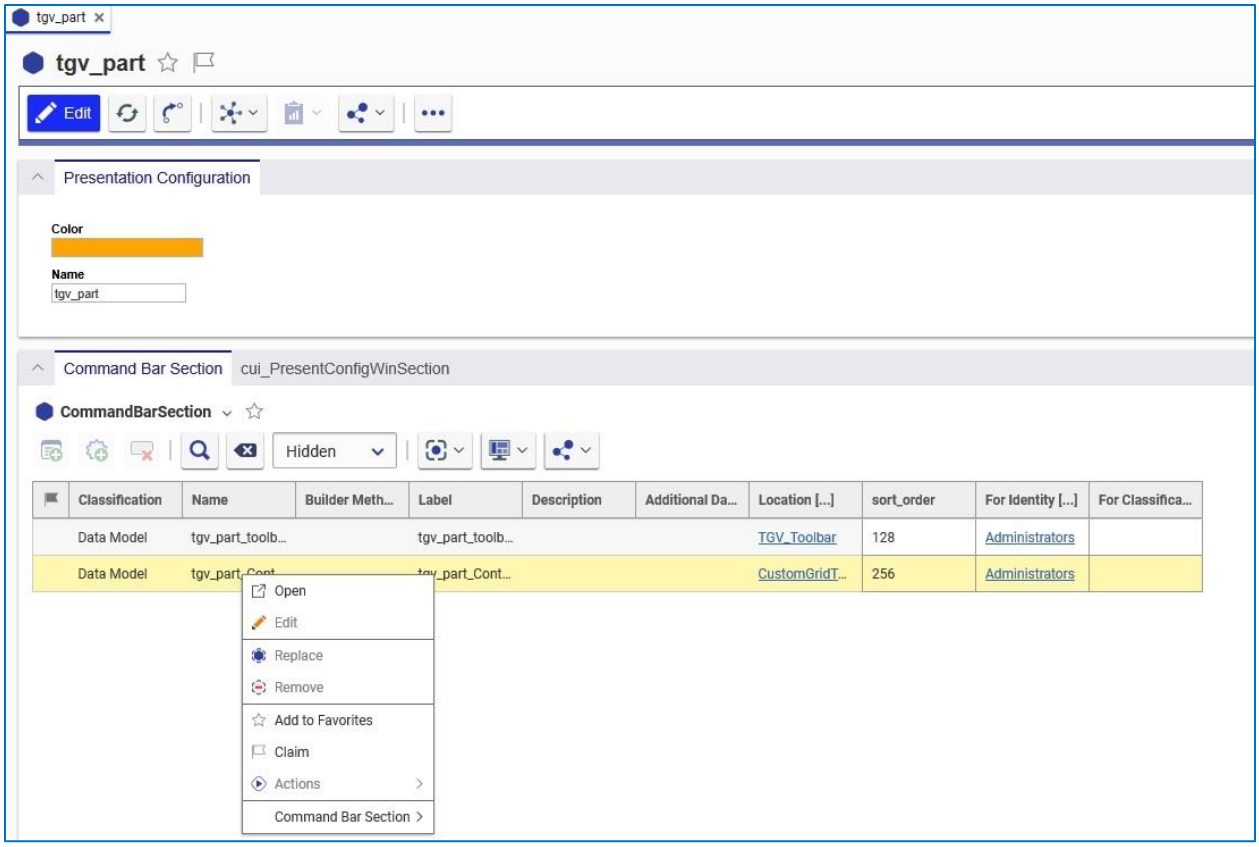



Figure 48.

The tgv_part_context_menu item appears.

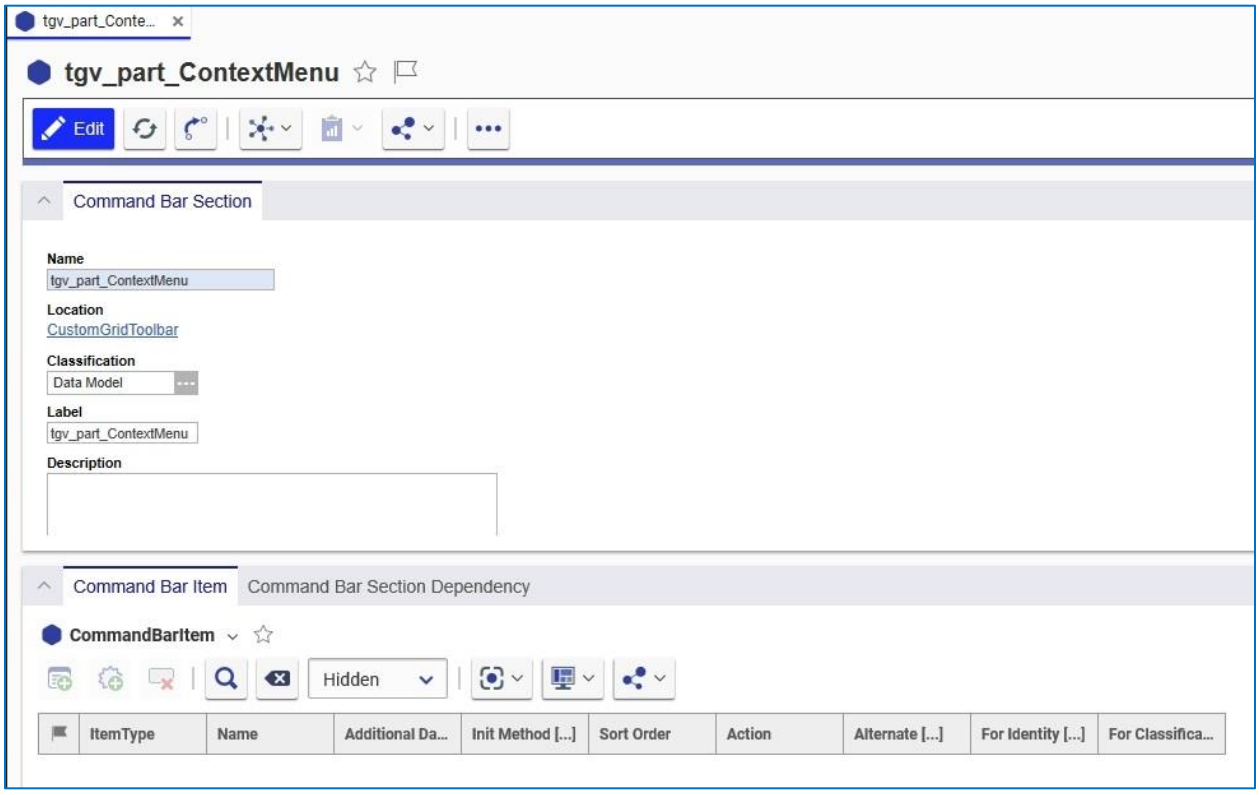



Figure 49.

2. Click  to open the item for editing.
3. Click **Create New Presentation Configuration**. The Presentation Configuration item appears.
4. Enter “tgv_part” in the Name field and select a color such as orange:

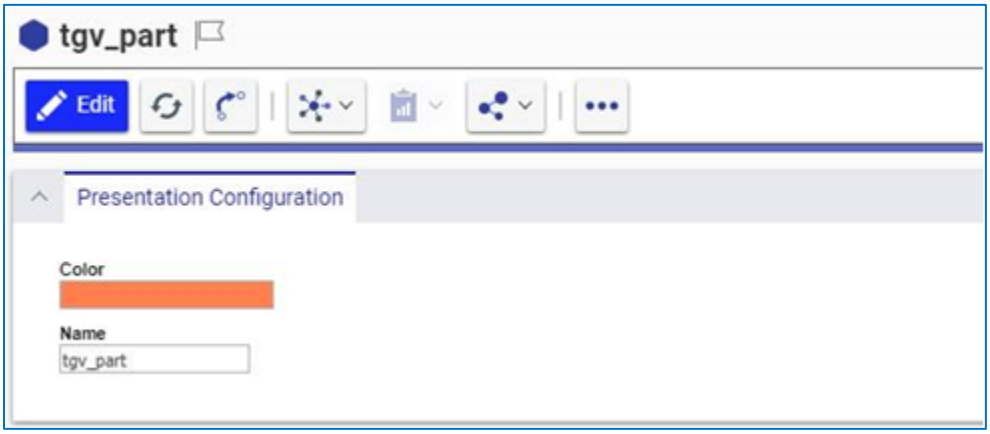



Figure 50.

5. Click the **New Command Bar** icon  in the Command Bar Section to add a new related item.

6. Enter “tgv*” in the **Name** cell of the search grid and add both tgv_part_toolbar and tgv_part_ContextMenu items.


Figure 51.

7. Make sure to add the items to the relationship grid.

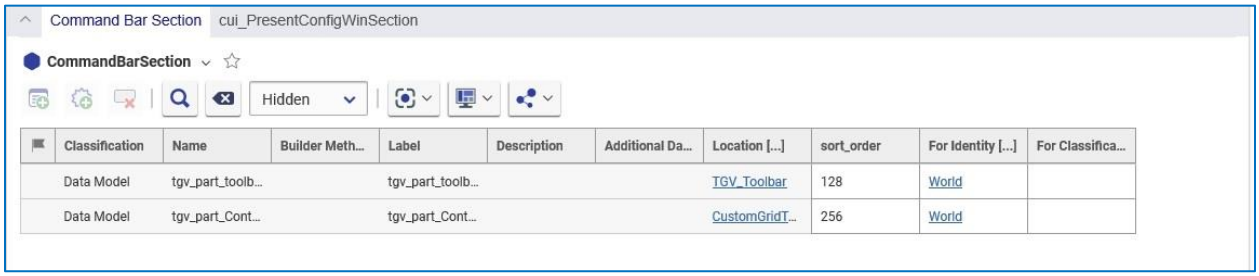



Figure 52.

8. Add an identity in the “For Identity” column. This Identity determines which users will have access to the buttons/menus.
9. Click 

to save the tgv_part Presentation Configuration item.
10. Right click **tgv_part_toolbar** and select **Open** from the context menu.

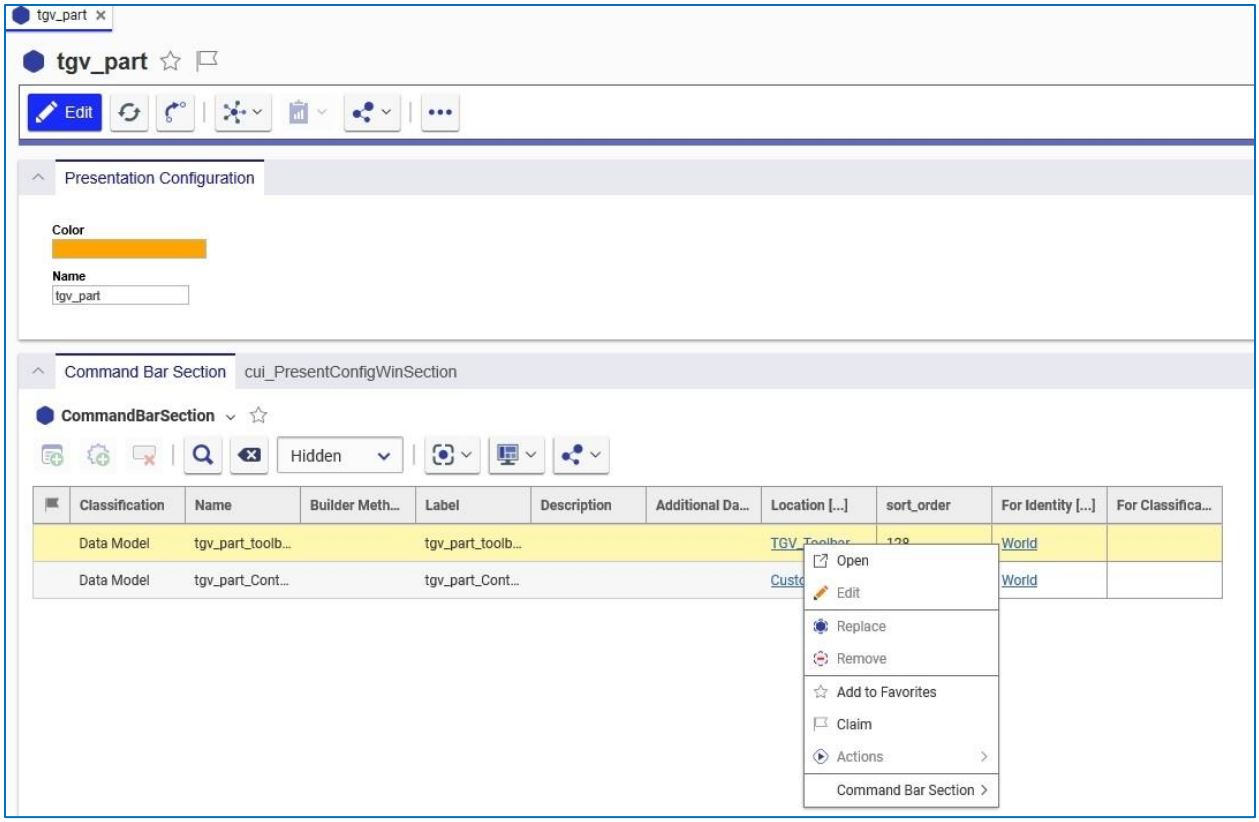



Figure 53.

The tgv_part_toolbar presentation configuration item appears. It contains the tgv_view button.

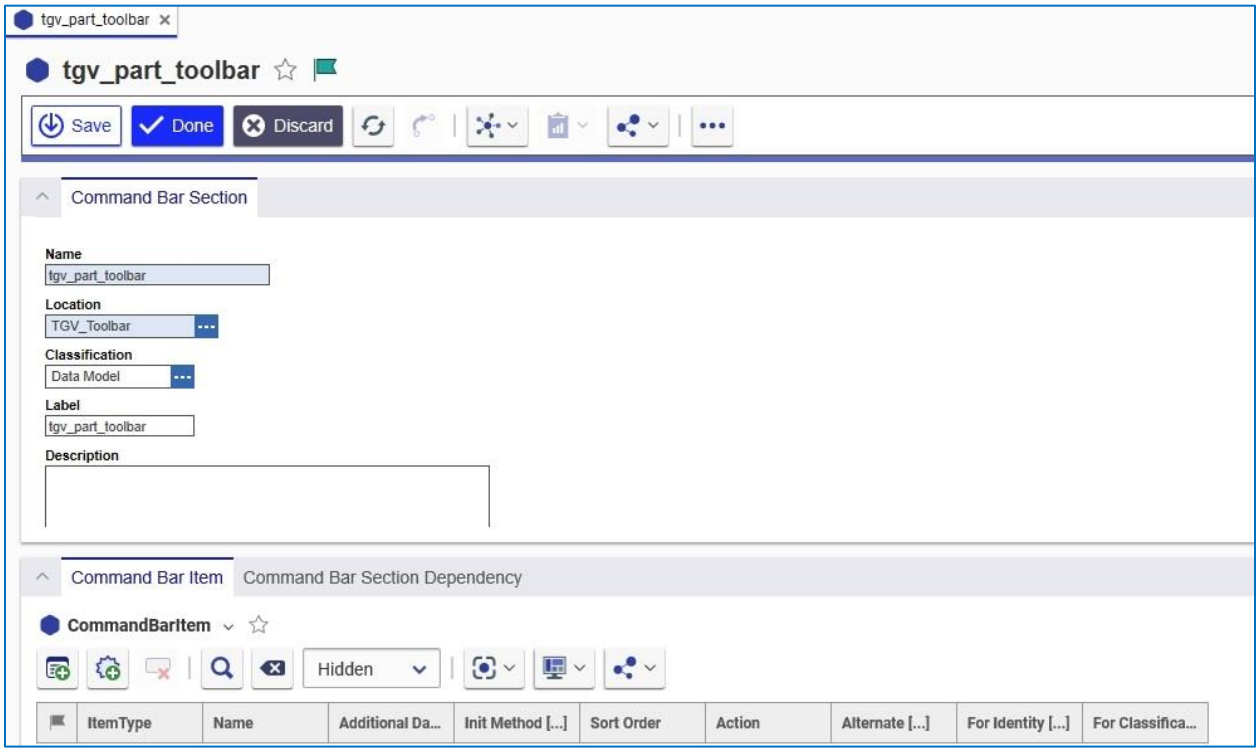



Figure 54.

11. Click the **New Command Bar** icon  in the Command Bar Item tab. The Select Item Type dialog appears:

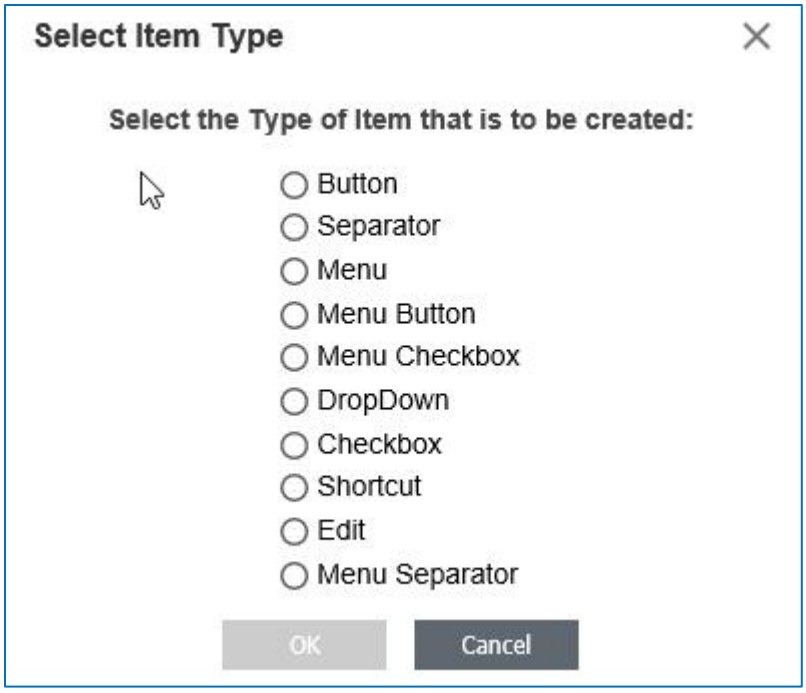



Figure 55.

12. Select **Menu** and click **OK**. It appears in the Command Bar Item tab.
13. Right click on the Menu ItemType and select **Open** from the context menu.
14. Right click tgv_part_menu in the Command Bar Item grid and select “View Command Bar Item.” The tgv_part_menu presentation item appears:

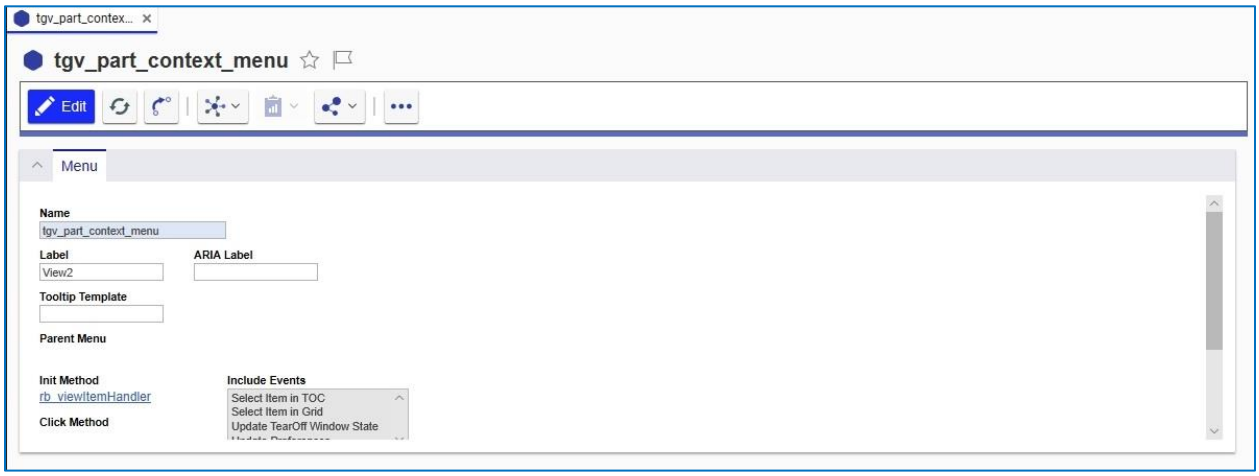



Figure 56.

The rb_viewItemInitHandler method is used to determine whether or not the data template is valid.

# 4.5 Expanding/Collapsing Tree Grid Rows

You can determine the number of rows that appear in a Tree Grid view using one of the following methods:

- Click the Grow  icon to temporarily expand the view.
- Select the **Auto Grow On Refresh** checkbox to keep the view expanded.

These methods are described in more detail in the following sections.

## 4.5.1 Expanding Tree Grid Rows Using the Grow Icon

Use the following procedure:

1. Click the **Grow** icon  to expand the view. The number of levels that appear is determined by the value you enter in the **Grow depth** field.

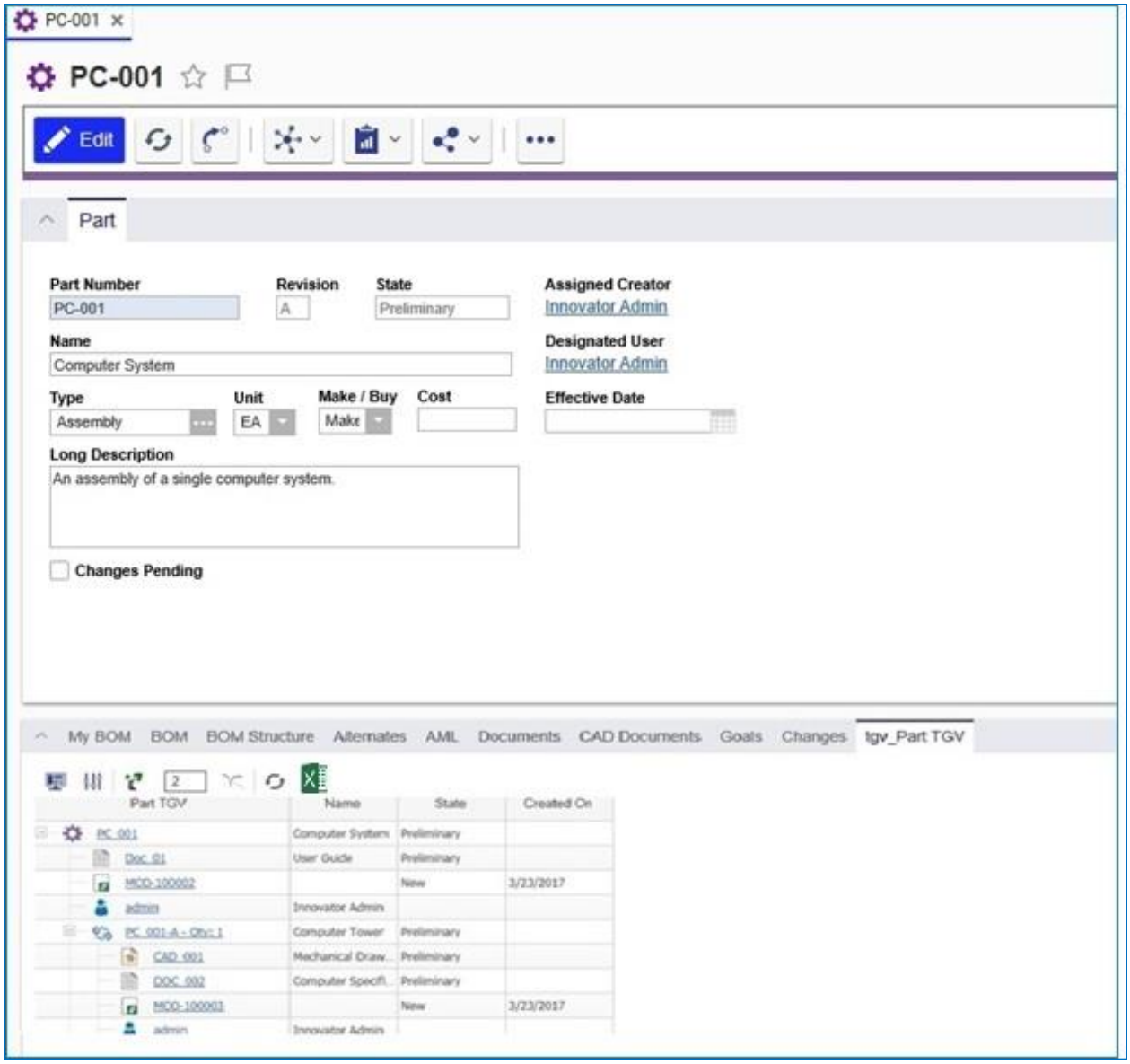



Figure 57.

**Note:** You can also select Grow from the context menu by right-clicking on the row that you want to expand.

2. Click the **Show More** link to further expand the number of related Items at any level.
3. You can change the number of levels that appear in the grid either by either entering a different value in the **Grow depth** field or by editing the related Tree Grid View definition:

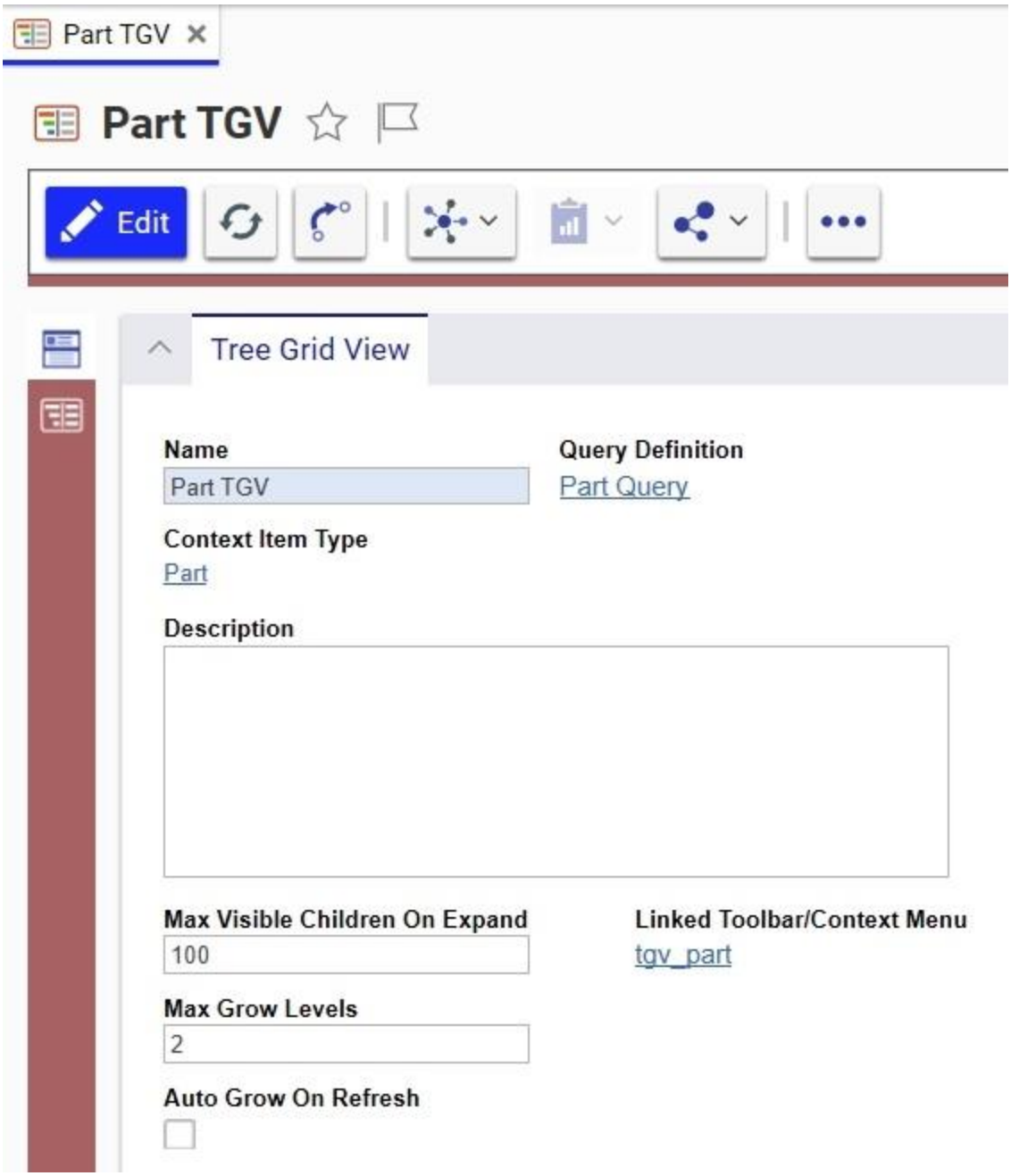



Figure 58.

Any changes you make in the **Max Visible Children on Expand** and **Max Grow Levels** fields automatically appear in the associated item screen. For example, if you change the value in the **Max Grow Levels** field to 4, it will appear in the **Grow Depth** field for the associated Part:

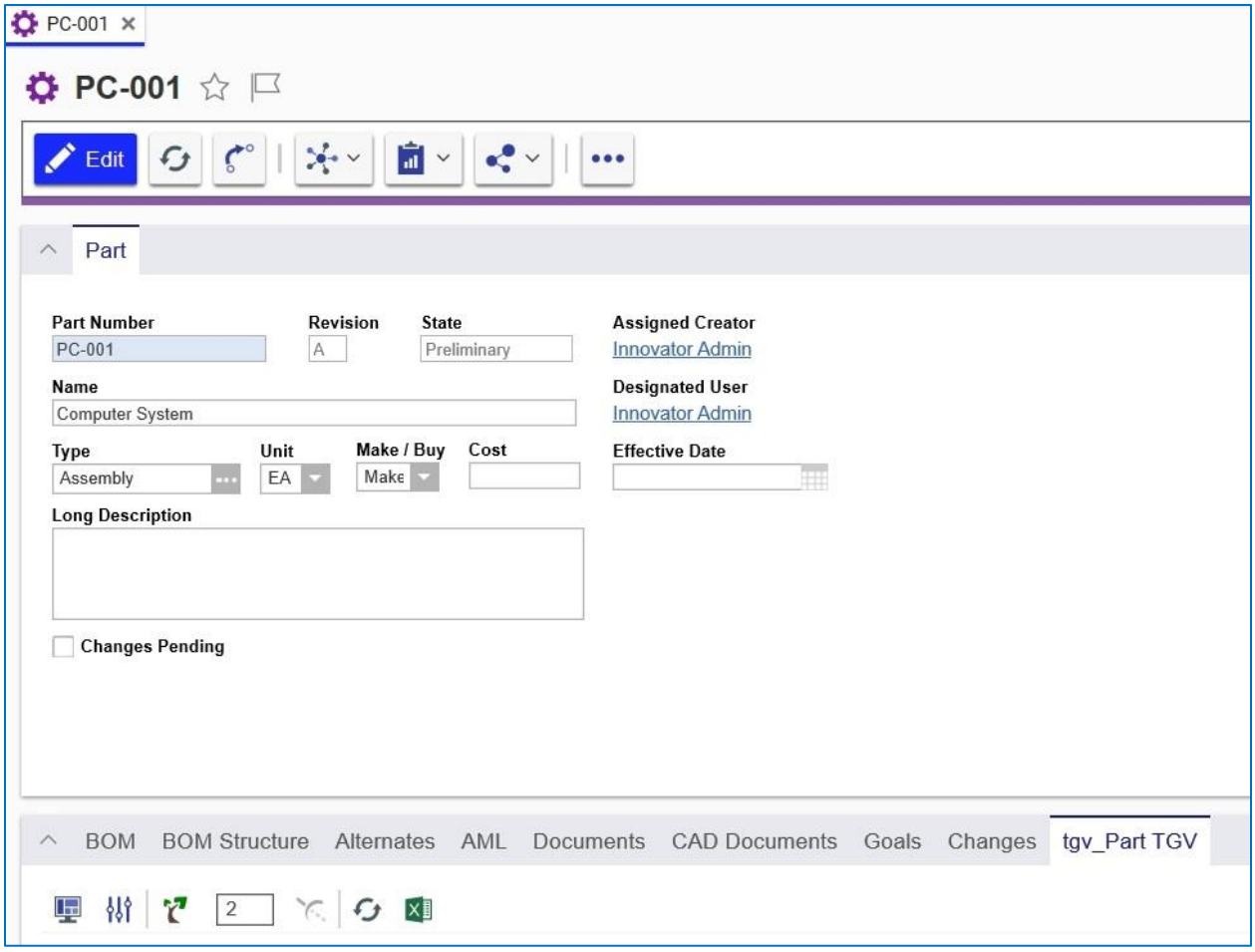



Figure 59.

### 4.5.2 Using Auto Grow on Refresh

The **Auto Grow on Refresh** check box appears on the Tree Grid View form.

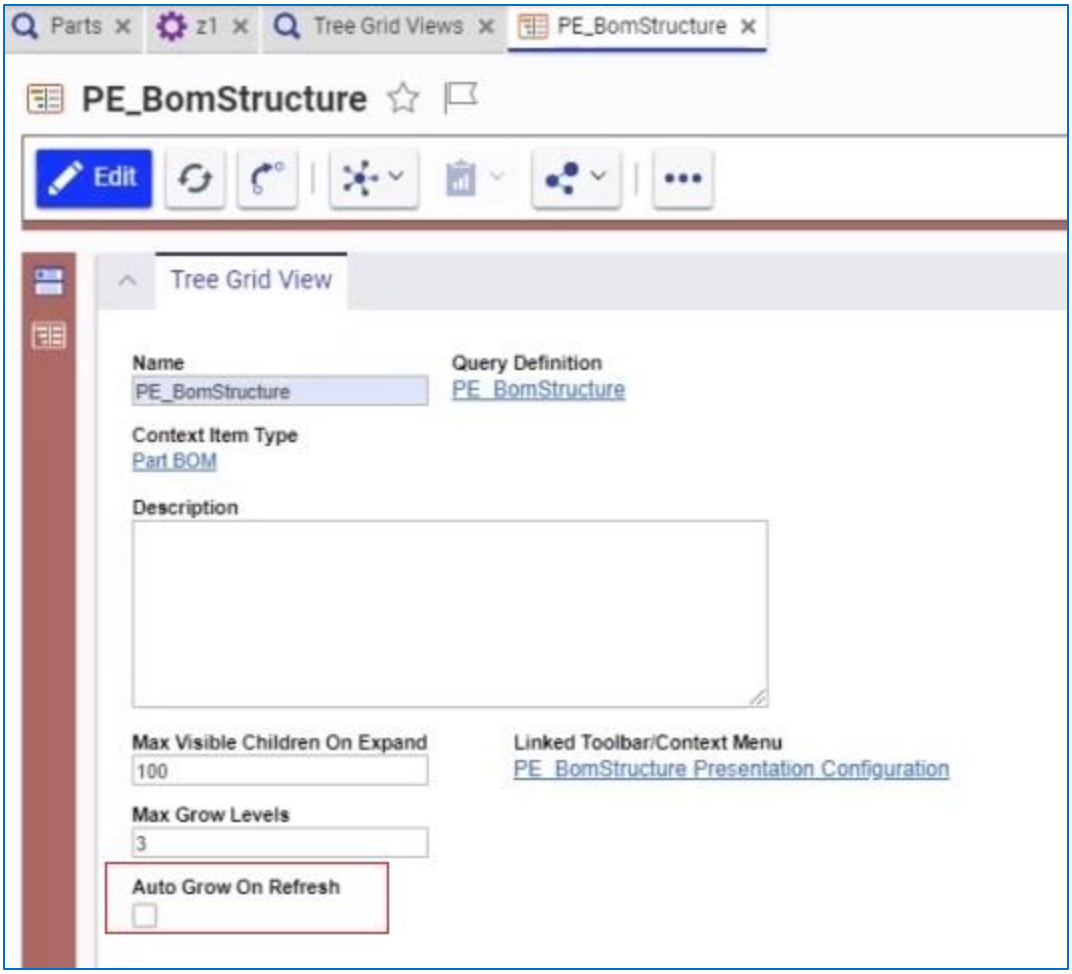



Figure 60.

Selecting this checkbox keeps the Tree Grid View expanded to the default level whenever you do a refresh.

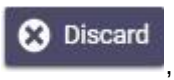
The Max Grow Level property value is used when you:

- Open the tab that displays the Tree Grid View.
- Click 

.

The Grow Depth value displayed on the tab is used when you:

- Click **Apply** on either the Display Settings or Modify Parameters dialog boxes.
- Click the following toolbar buttons:  , 

, 

, 

.
- Click the  icon and select **Create New Revision** from the drop-down menu.

Clicking the Display Settings  or Parameter Mapping  icons also initiates a refresh.

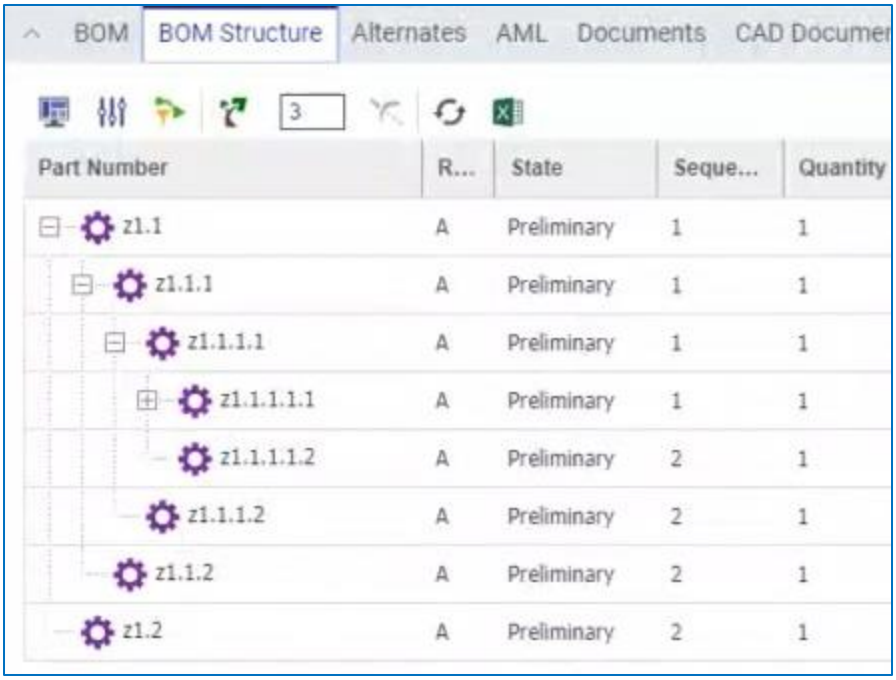



Figure 61.

If you do not check the **Auto Grow on Refresh** box, the Tree Grid view maintains the default behavior of contracting the Tree Grid view on refresh.

### 4.5.3 Collapsing Tree Grid Rows

Use the following procedure:

1. Select a row in the Tree grid view:

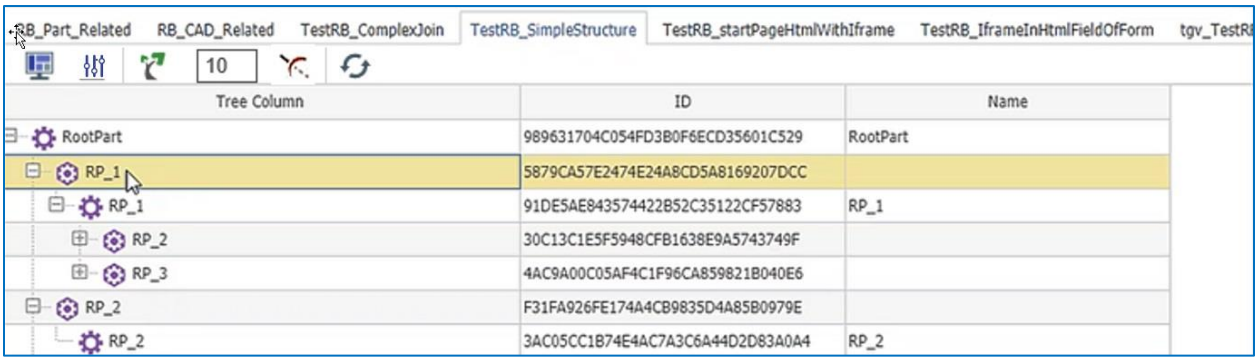



Figure 62.

2. Click the **Trim** icon to collapse the rows:

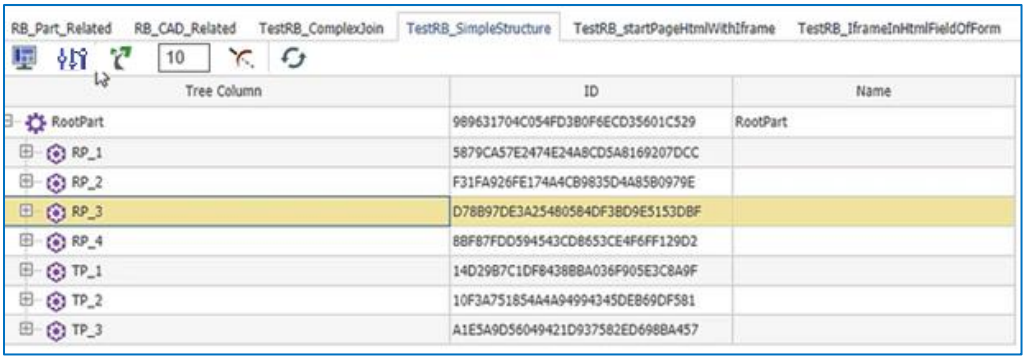



Figure 63.

**Note:** You can also select Trim from the context menu by right-clicking on the row that you want to collapse.

## 4.6 Using Display Settings

The Display Settings dialog enables you to change the way information is displayed in the Tree Grid View. Use the following procedure:

1. Select **Design>Parts** from the TOC. In this example, we use RootPart.

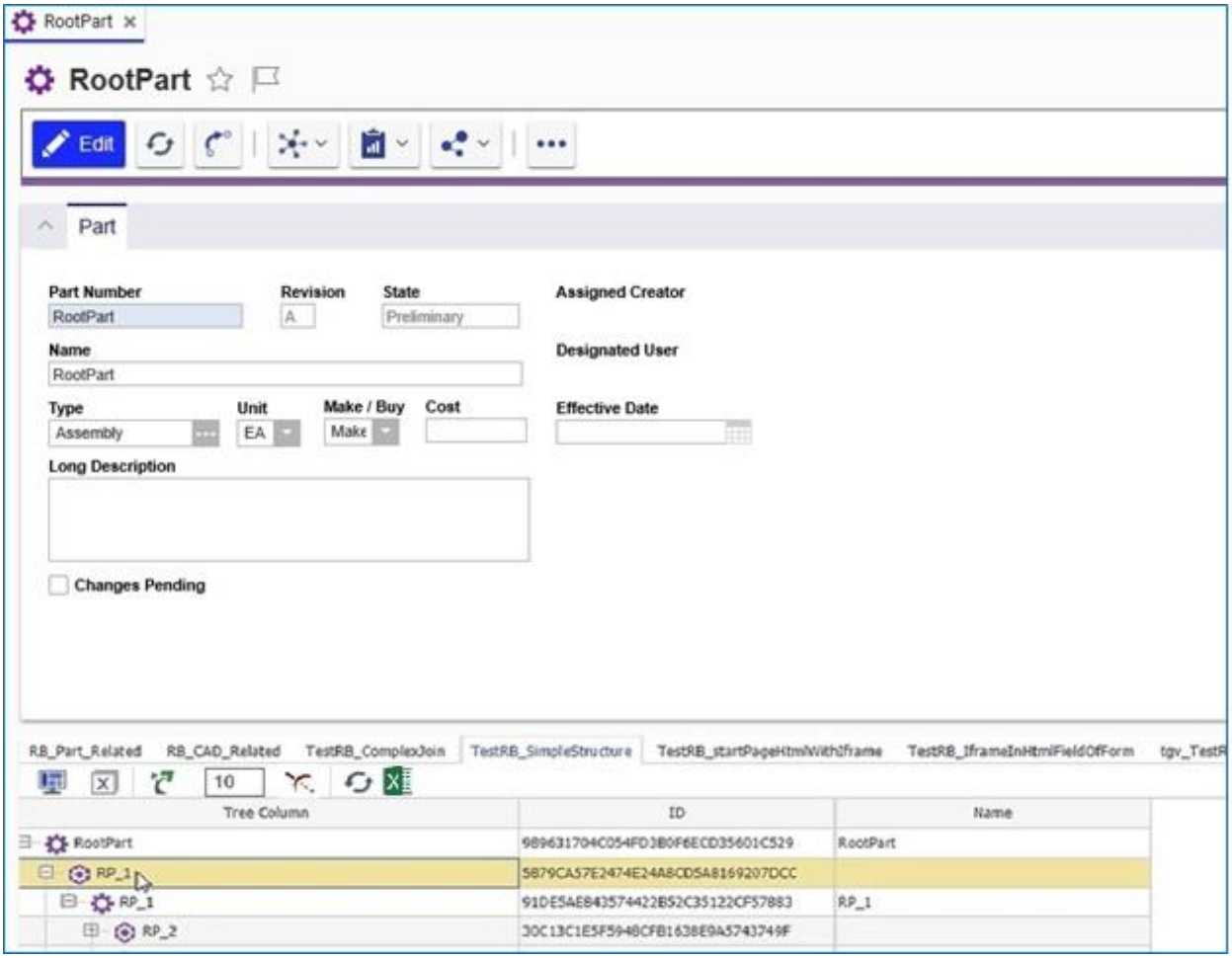



Figure 64.

2. Click the **Display Settings** button. The Display Settings dialog appears. It displays the Tree Grid view configuration.

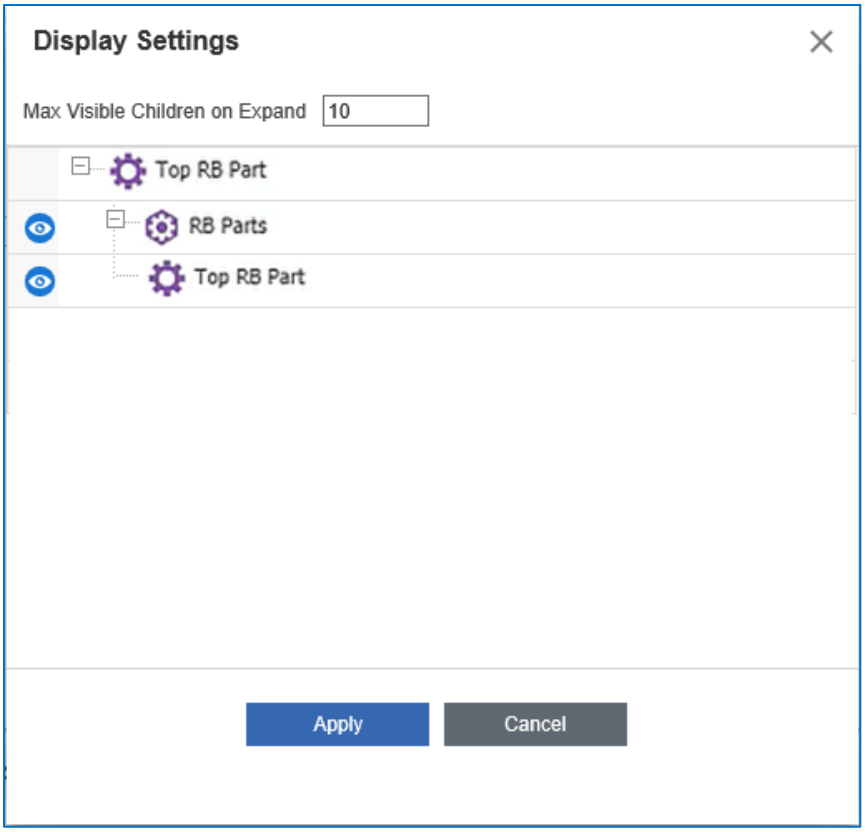



Figure 65.

3. Enter a number in the **Max Visible Children on Expand** field to change the number of child nodes that are returned when the parent Item is expanded.

4. Click the Toggle Visibility icon for a related Item to turn it off. The selected Item is highlighted and grayed out. In the example below, the selected part has a recursive association which is also turned off.

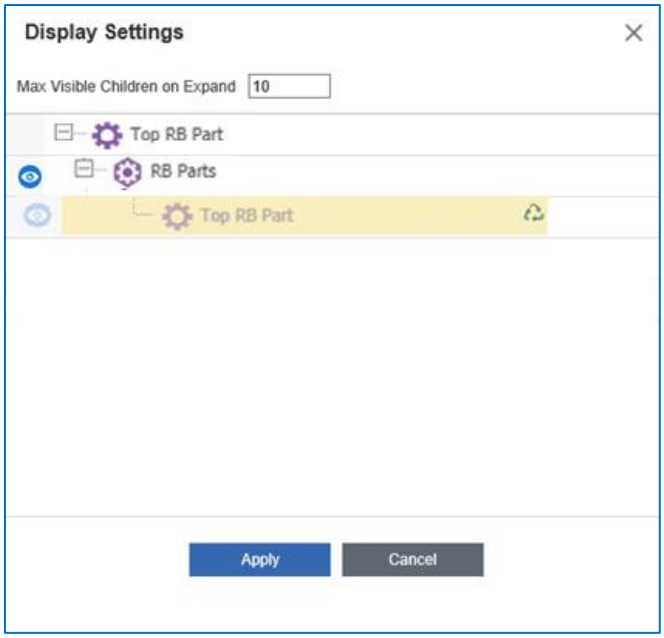



Figure 66.

5. Click **Apply**. The grid is refreshed.

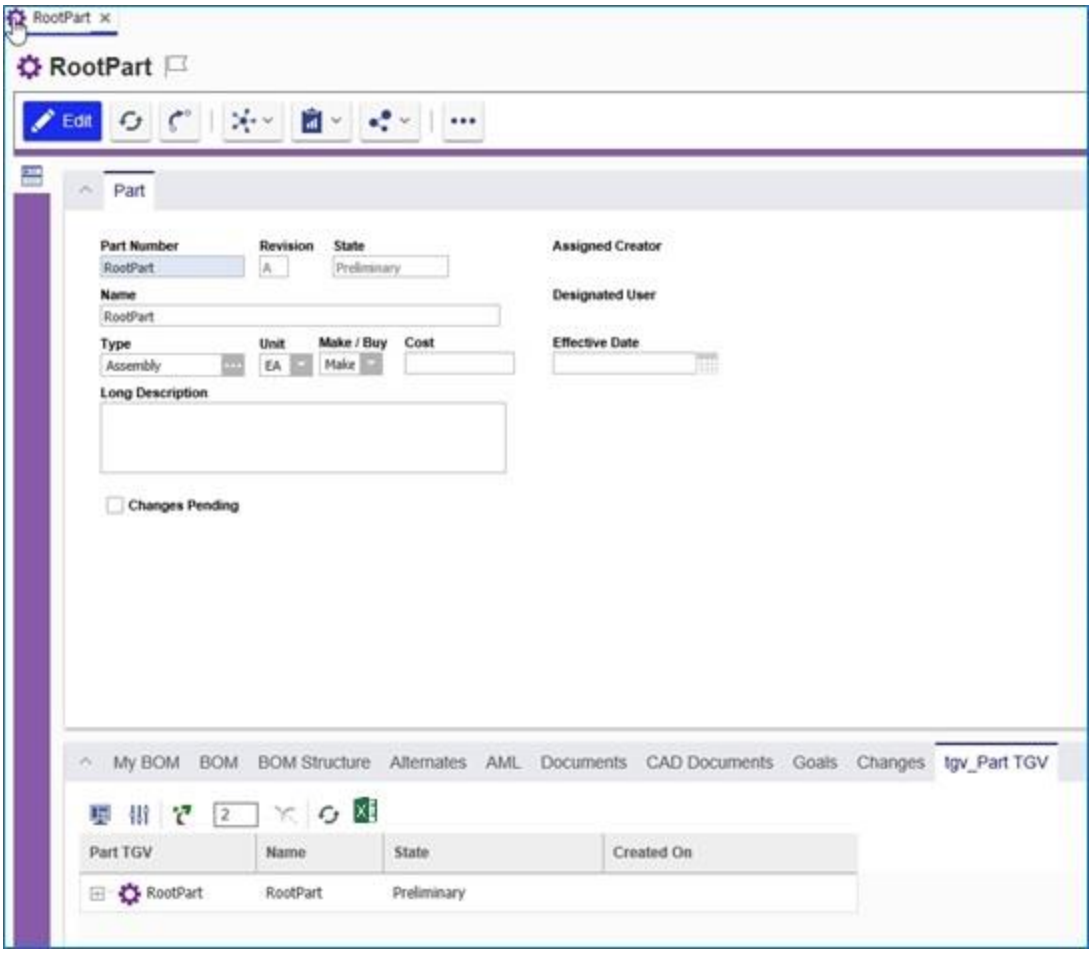



Figure 67.

6. Click **Grow**. The grid does not show related items; it only shows BOM relationships:

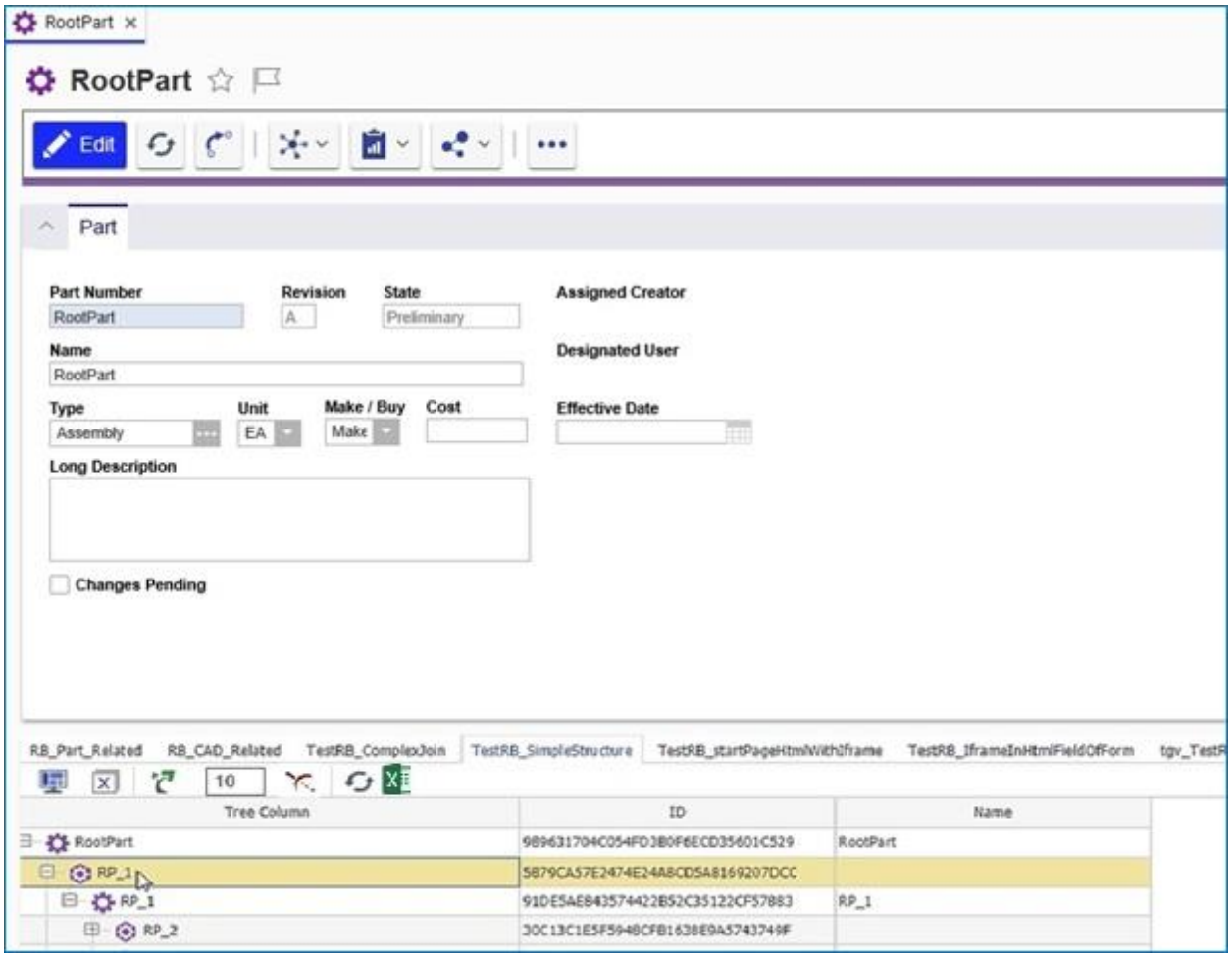



Figure 68.

## 4.7 Show Parameter Mapping

Once you have created the Query Definition and configured the Tree Grid View, click the **Show Parameter** Mapping icon . The Map Parameters dialog box appears:

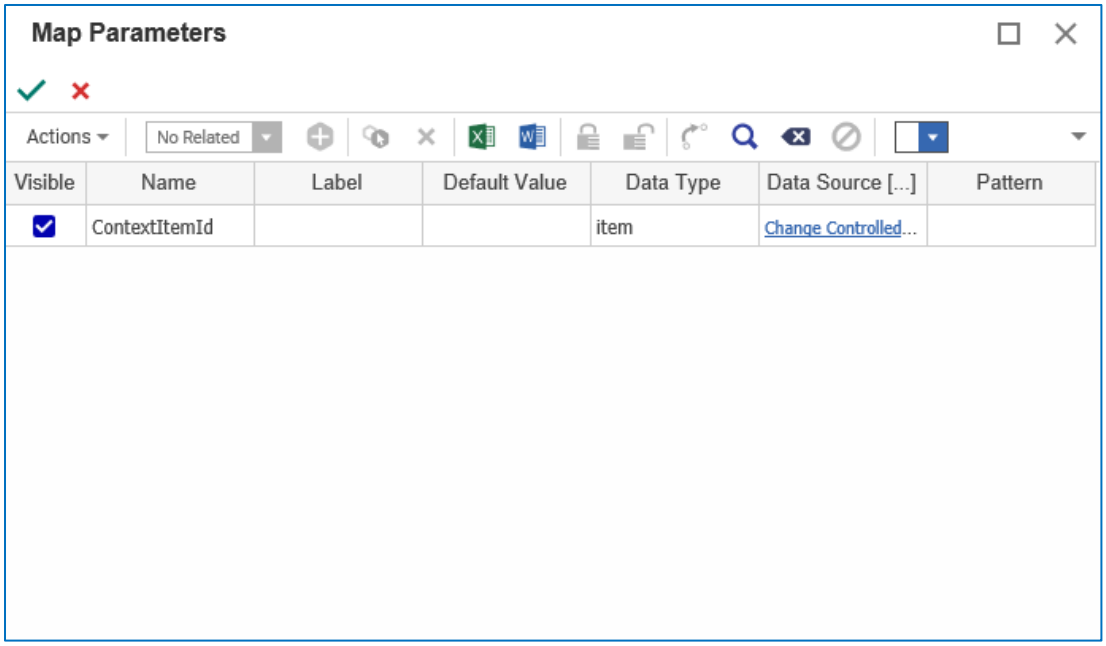



Figure 69.

Click the green arrow to make the doc_state parameter visible to users in the Tree Grid View. Administrators can also modify the parameter mapping and choose alternative values for the Label and Default Values fields. The Data Type List field is similar to the properties assigned to ItemTypes.

## 4.8 Cancelling a Long Running Query

When a query has been running for several seconds, a splash screen similar to the following appears with a Cancel button that, when selected, will stop the request and terminate any server-side processes used to execute the query.

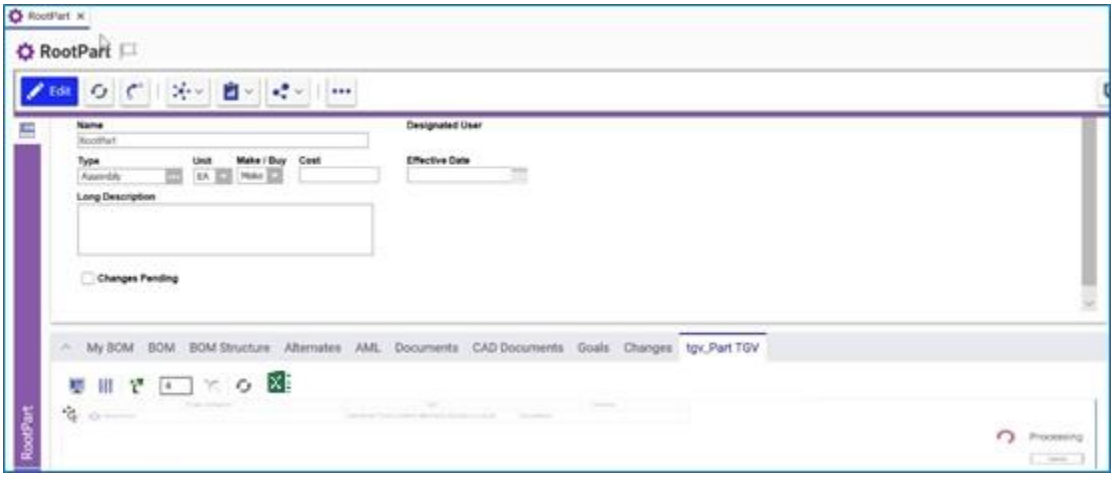



Figure 70.

# 4.9 More About Cell View Types

Support for the following cell view data types has been added in Aras Innovator 12.0:

- List
- Boolean
- Floating Point
- Integer
- Date enhancements for Short/Long Date/time formatting

## 4.9.1 Using the List Cell View Type

You can use the List cell view type to create list properties in a Tree Grid View as shown in the following procedure.

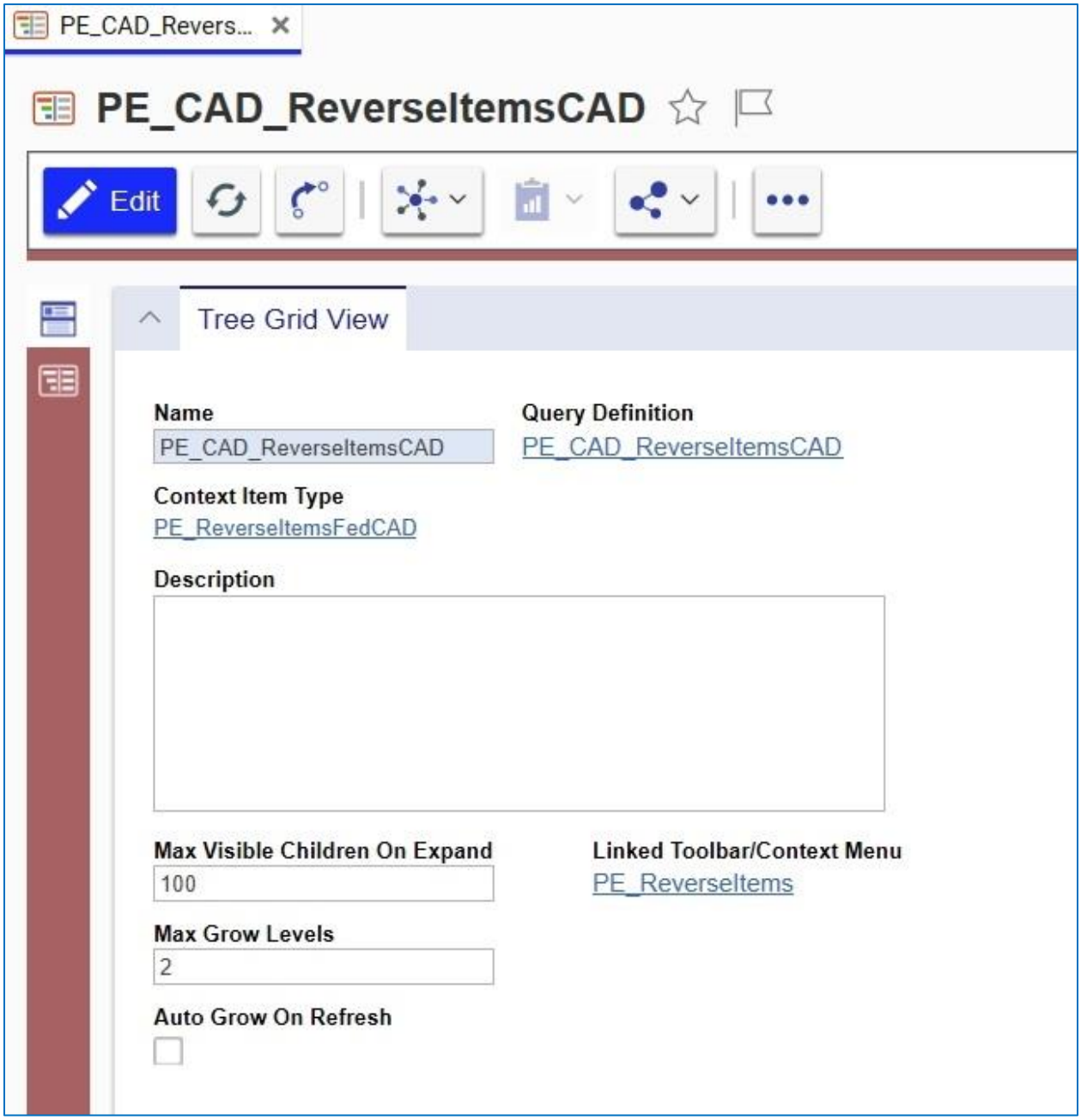



Figure 71.

1. Click the **Show Editor** icon to view the parameters.


Figure 72.

2. Right-click on the entry in the Authoring Tool column and select **Cell Display Settings**

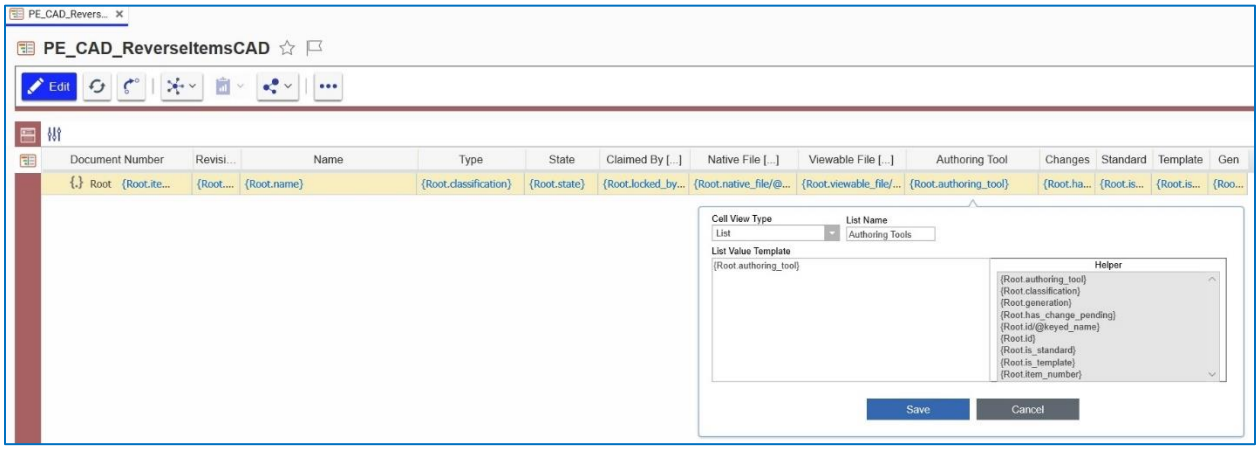



Figure 73.

3. Select **List** from the Cell View Type dropdown and enter the name of the list in the List Name field. When applying a List data type, the Label of the List Item will appear in the Tree Grid View.
4. Click **Save**. The following example shows the result when displaying a List Property:

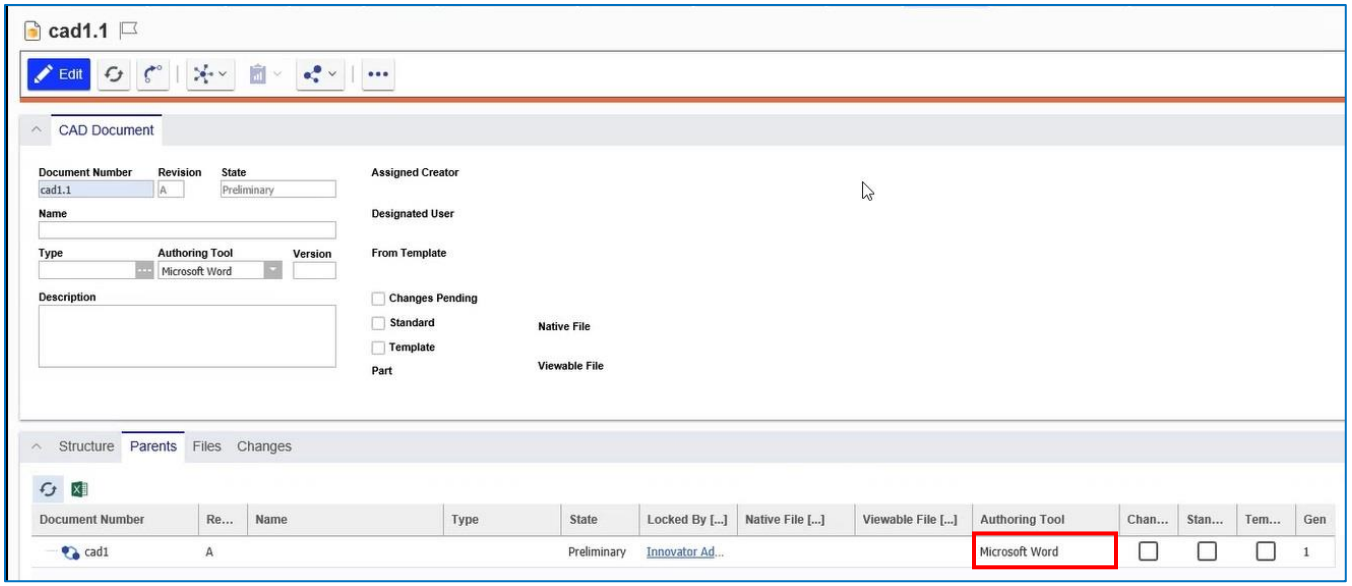



Figure 74.

## 4.9.2 Using the Boolean Cell View Type

The following example shows how to use the Boolean Cell View Type. The example uses a sample Tree Grid View Definition: PE_CAD_ReverseItemsPart.

1. Click the **Show Editor** icon to view the parameters:

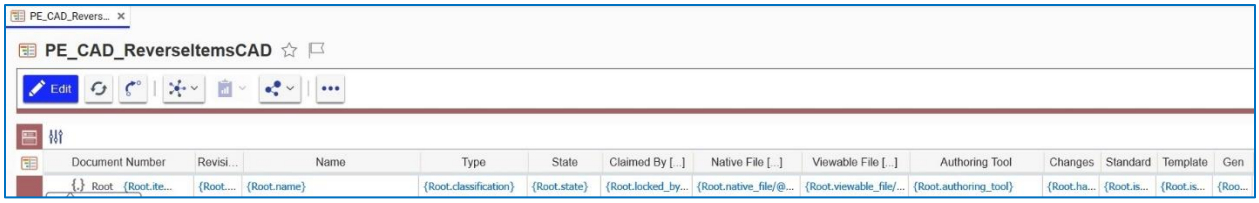



Figure 75.

2. Right click the entry in the **Changes** column and select **Boolean** from the Cell View Type dropdown list:

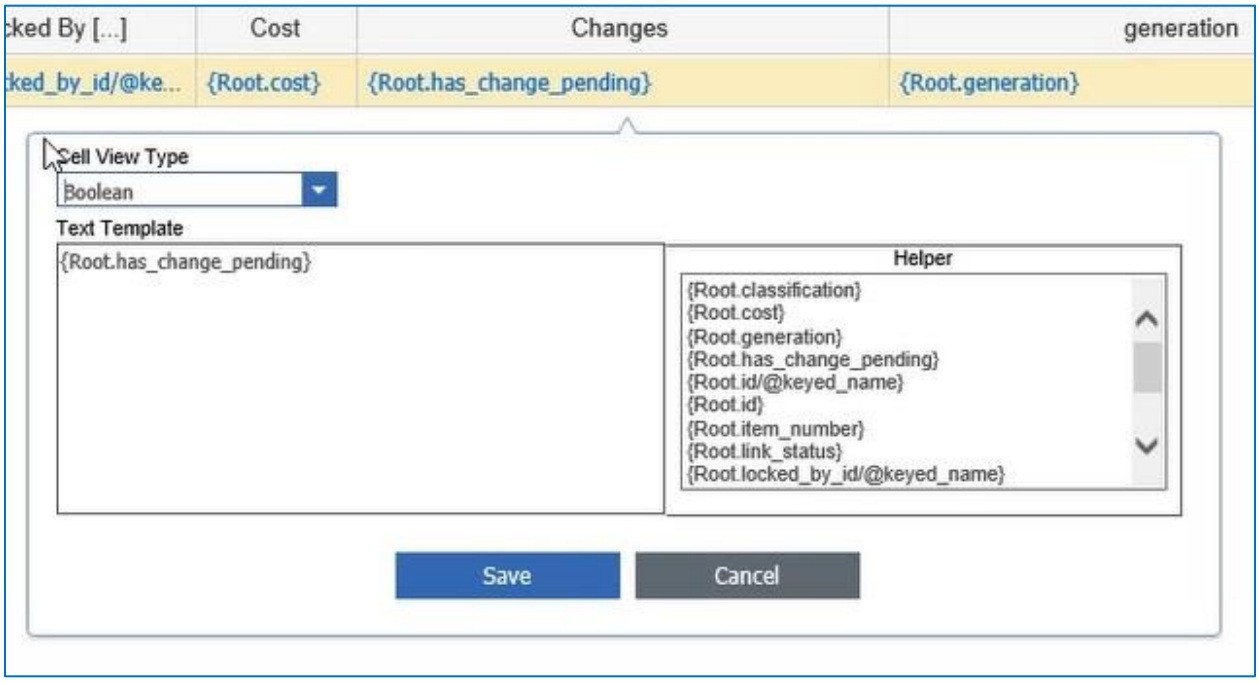



Figure 76.

3. Click **Save**.

### 4.9.3 Using the Decimal Cell View Type

The following example shows the use of the Decimal Cell View Type. This Cell View Type enables the user to specify a Precision and Scale for an associated Double or Floating Point property type. In this example, the “Cost” column in the example Tree Grid View Definition refers to a “Cost” property in the associated Query Definition.

1. Right click the entry in the **Cost** column and select **Decimal** from the Cell View Type dropdown list.
2. Enter the appropriate values in the **Precision** and **Scale** fields and click **Save**.

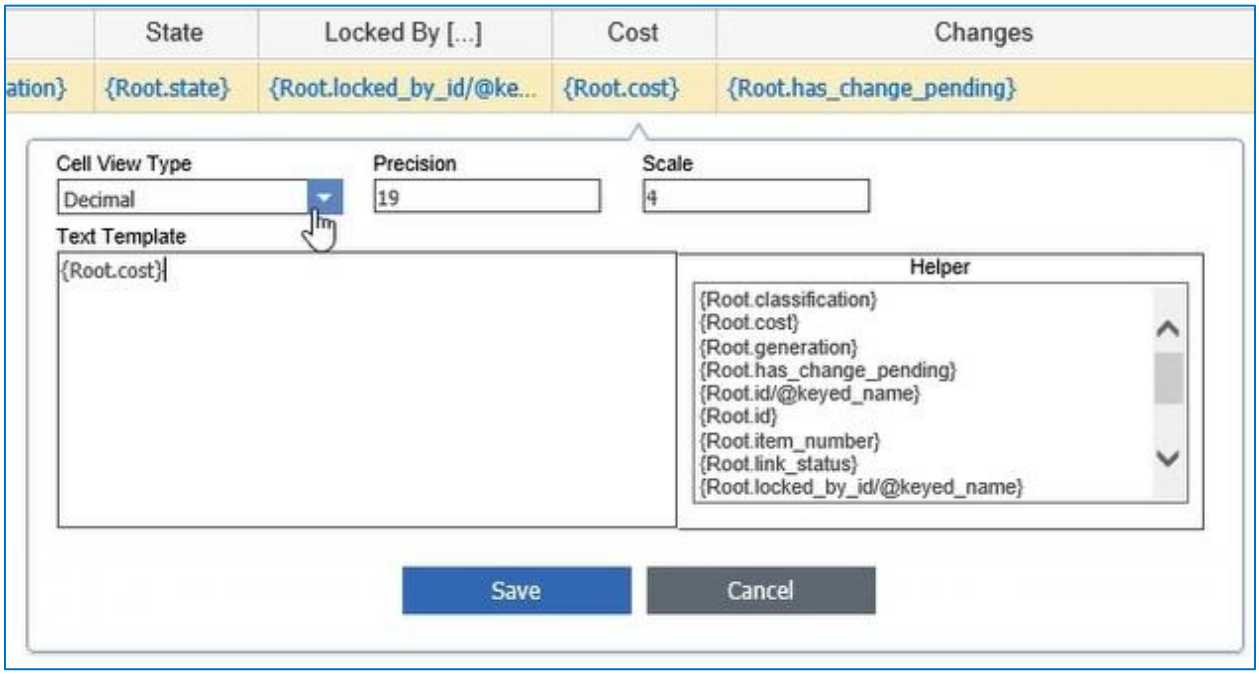



Figure 77.

### 4.9.4 Using the Float Cell View Type

The following example also uses the PE_CAD_ReverseItemsPart Tree Grid View. This example contains a Query Definition that refers to a Floating Point Property – 'Cost.'

1. Click the Show Editor icon to view the parameters:


Figure 78.

2. Right click the entry in the Cost column and select Float from the Cell View Type dropdown list.

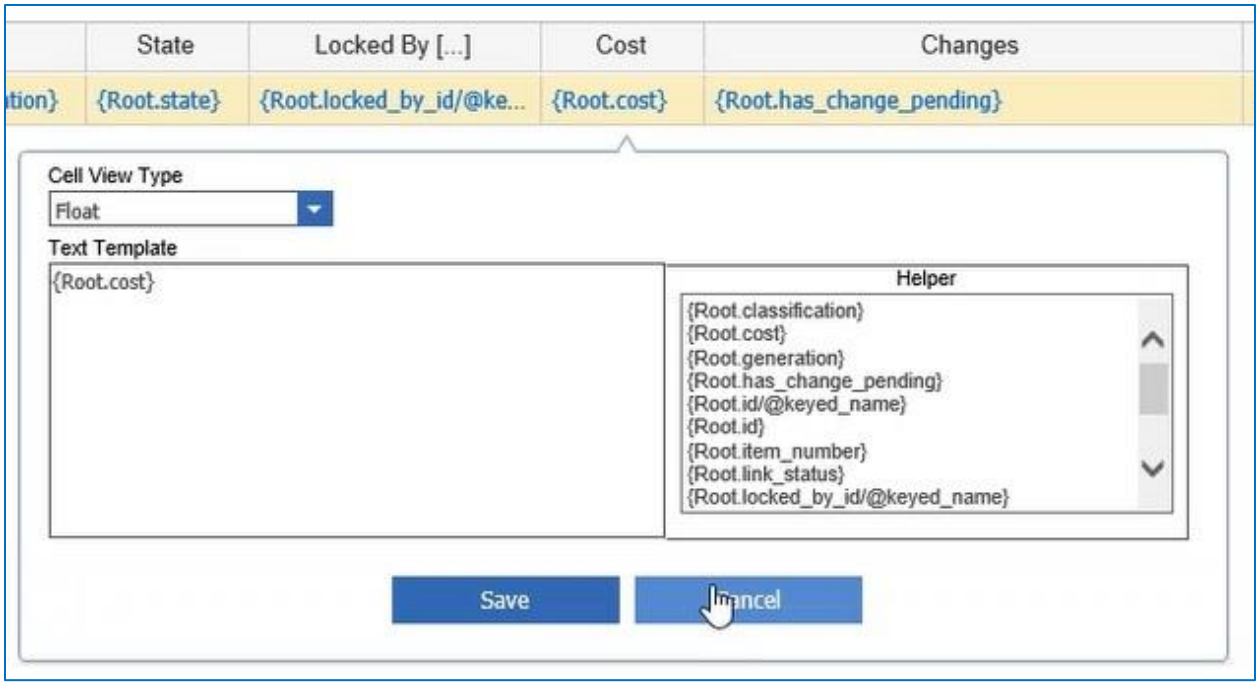



Figure 79.

### 4.9.5 Using Date/Time Cell View Types

The Date cell view type now supports ‘short’ or ‘long’ date and ‘date with time’ formats for date properties. A secondary selection drop-down is automatically displayed when the Data Cell View Type is selected initially.

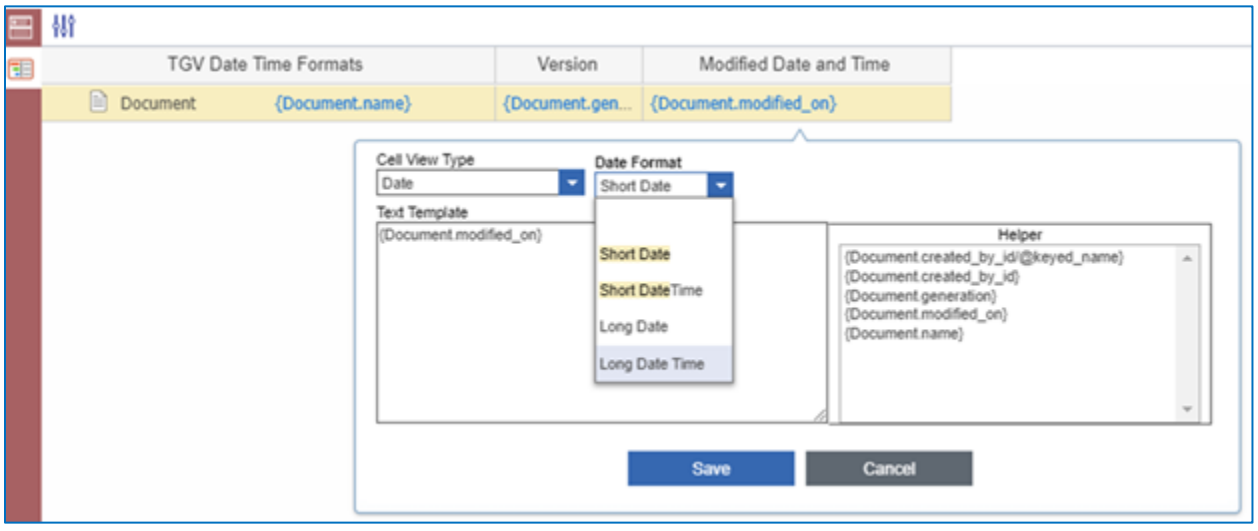



Figure 80.

In this example, the “created_on” Date type property is displayed as Cell View Type “Long Date Time.” The results appear in the following format:

| Version | Modified Date and Time |
|---|---|
| 1 | Friday, October 25, 2019 11:08:01 AM |
| 2 | Friday, October 25, 2019 4:46:52 PM |
| 3 | Friday, October 25, 2019 4:52:30 PM |
| 4 | Friday, October 25, 2019 4:52:33 PM |
| 5 | Friday, October 25, 2019 4:52:36 PM |
| 6 | Friday, October 25, 2019 4:52:40 PM |
| 7 | Friday, October 25, 2019 4:52:43 PM |
| 8 | Friday, October 25, 2019 4:52:47 PM |

Figure 81.

Selecting “Short Date” as shown in the following figure, does not provide enough information.

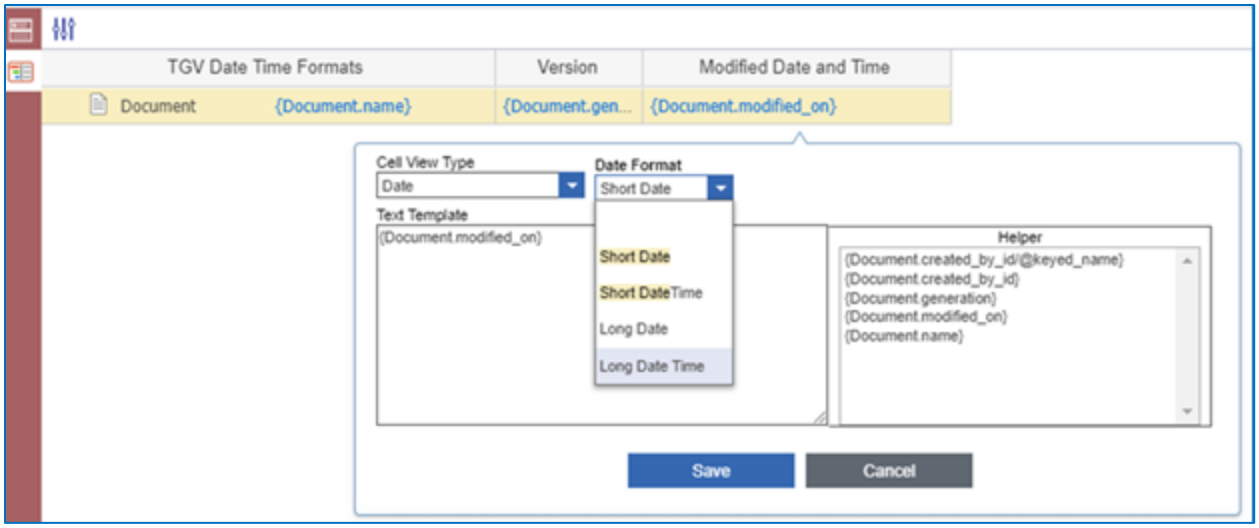



Figure 82.

In this example, the “created_on” Date type property is displayed as Cell View Type “**Long Date Time**.” The result appears in the following format:

| Version | Modified Date and Time |
|---|---|
| 1 | Friday, October 25, 2019 11:08:01 AM |
| 2 | Friday, October 25, 2019 4:46:52 PM |
| 3 | Friday, October 25, 2019 4:52:30 PM |
| 4 | Friday, October 25, 2019 4:52:33 PM |
| 5 | Friday, October 25, 2019 4:52:36 PM |
| 6 | Friday, October 25, 2019 4:52:40 PM |
| 7 | Friday, October 25, 2019 4:52:43 PM |
| 8 | Friday, October 25, 2019 4:52:47 PM |

Figure 83.

### 4.9.6 Using the Integer Cell View Type

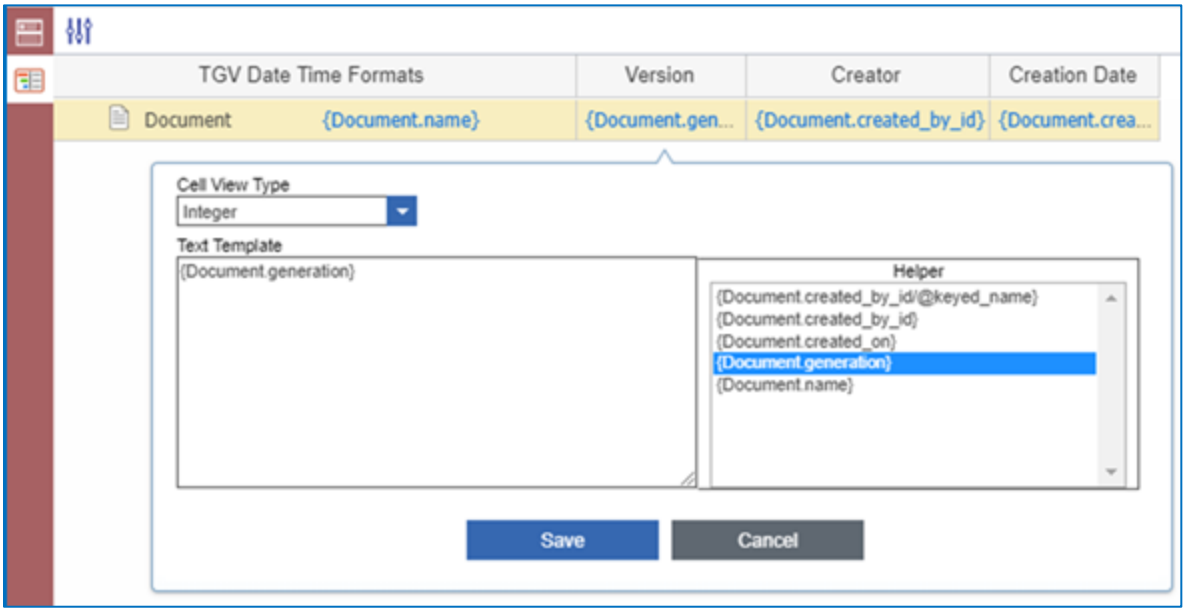



Figure 84.

The Integer Cell View Type provides an alternative to the conversion of numeric values to text for display in the Tree Grid View. Using Integer, the value is sorted numerically (ascending or descending) unlike Text, whose resulting values are sorted numerically instead. A common usage for Integer is Version (generation):

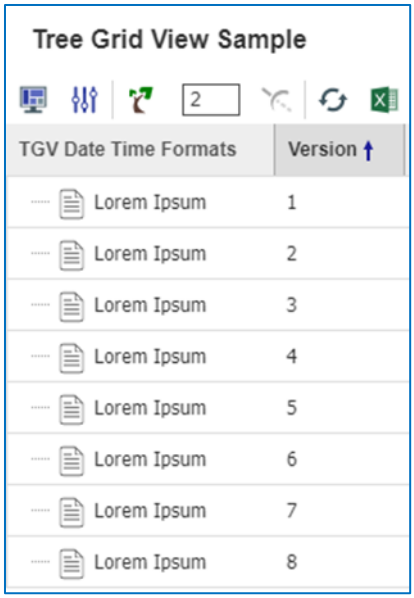



Figure 85.