Infor PLM Accelerate 11

Tree Grid View Administrator Guide

Important Notices

Publication Information

Publication date: May 31, 2018

Contents

About this guide

The purpose of this document is to describe how to use the Tree Grid View application to build a visual data structure which provides information about where a particular item will fit in the context of other items. It also describes how to use the Query Builder application to submit queries to get the data necessary to create a Tree Grid View.

Intended audience

This guide is intended for system administrators. This document assumes that you have at least some knowledge of:

- System architecture and function for your Infor system
- SQL Server database

For the most up-to-date list of software and hardware requirements for Infor products, see the documentation for your system.

Related documents

You can find the documents in the product documentation section of the Infor Xtreme Support portal, as described in the "Contacting Infor" section below.

Contacting Infor

If you have questions about Infor products, go to the Infor Xtreme Support portal at www.infor.com/inforxtreme.

If we update this document after the product release, we will post the new version on this Web site. We recommend that you check this Web site periodically for updated documentation.

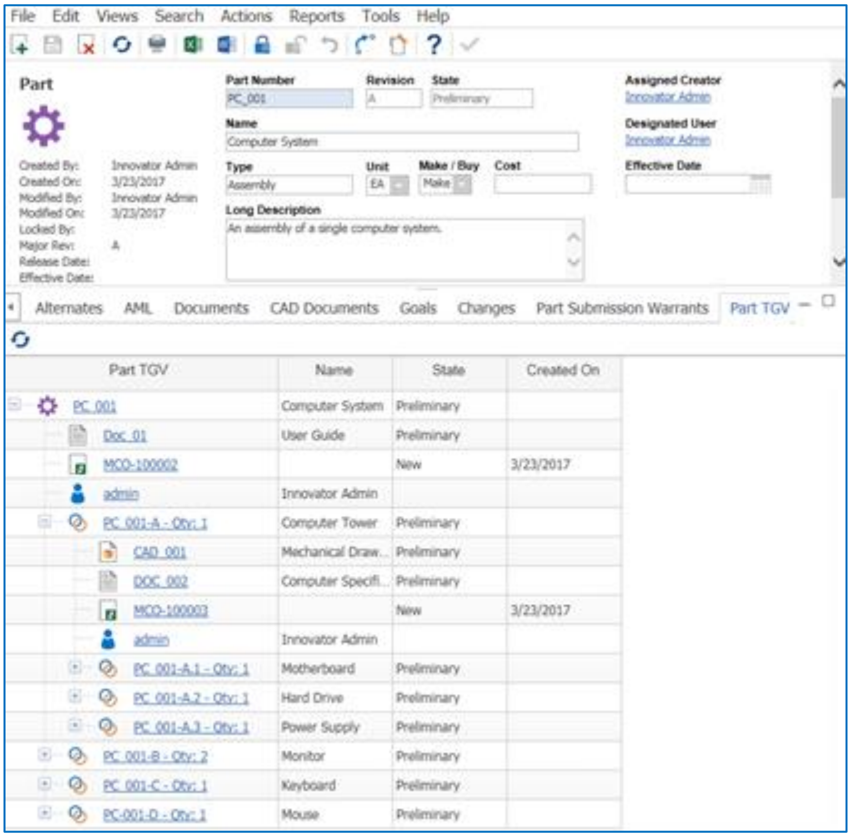
If you have comments about Infor documentation, contact documentation@infor.com.

Chapter 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 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:



Created By: Innovator Admin
Created On: 3/23/2017
Modified By: Innovator Admin
Modified On: 3/23/2017
Locked By:
Major Rev: A
Release Date:
Effective Date:

Part Number: PC_001
Revision: A
State: Preliminary
Name: Computer System
Type: Assembly
Unit: EA
Make / Buy: Make
Cost:
Long Description: An assembly of a single computer system.
Assigned Creator: Innovator Admin
Designated User: Innovator Admin
Effective Date:

Part TGV	Name	State	Created On
PC_001	Computer System	Preliminary	
Doc_01	User Guide	Preliminary	
MCO-100002		New	3/23/2017
admin	Innovator Admin		
PC_001-A - Qty: 1	Computer Tower	Preliminary	
CAD_001	Mechanical Draw...	Preliminary	
DOC_002	Computer Specifi...	Preliminary	
MCO-100003		New	3/23/2017
admin	Innovator Admin		
PC_001-A.1 - Qty: 1	Motherboard	Preliminary	
PC_001-A.2 - Qty: 1	Hard Drive	Preliminary	
PC_001-A.3 - Qty: 1	Power Supply	Preliminary	
PC_001-B - Qty: 2	Monitor	Preliminary	
PC_001-C - Qty: 1	Keyboard	Preliminary	
PC-001-D - Qty: 1	Mouse	Preliminary	

Figure 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.

Chapter 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.

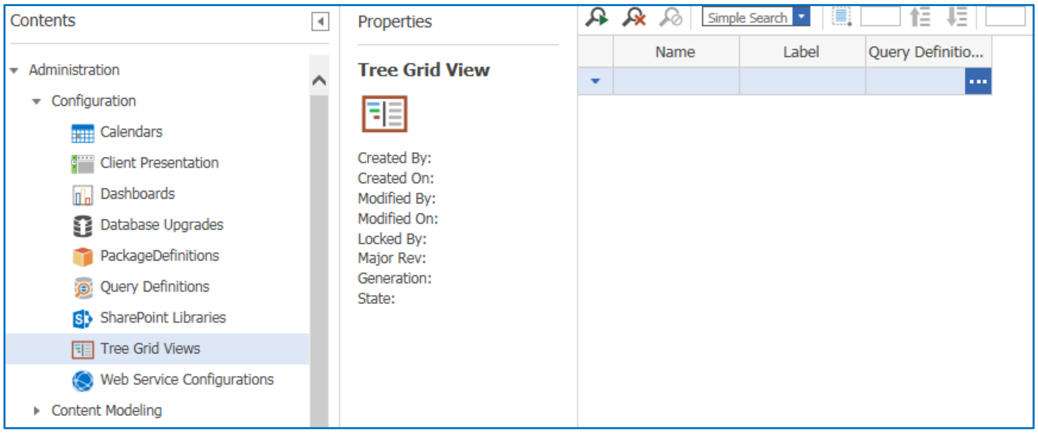


Figure 2.

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.

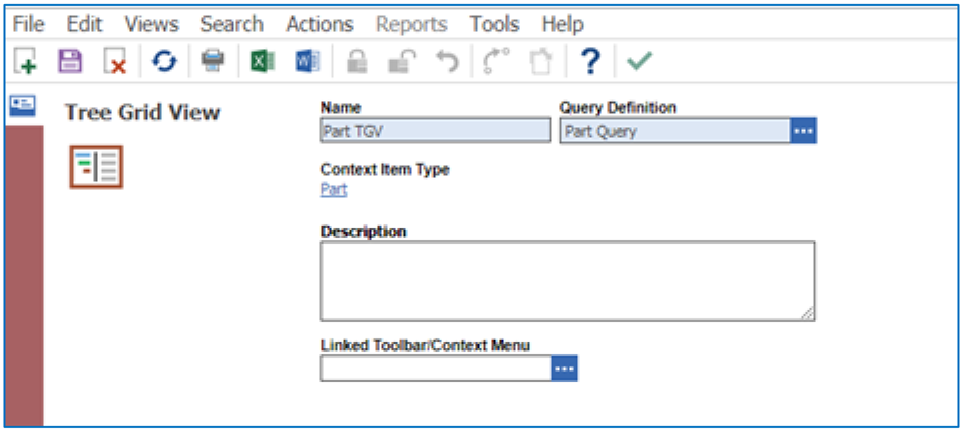
Creating a Query Definition

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.

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.



Name: Part TGV

Query Definition: Part Query

Context Item Type: Part

Description:

Linked Toolbar/Context Menu:

Figure 3.

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

2. After saving the item, click the **Show Editor** button on the left sidebar to go into Grid-Editing mode.

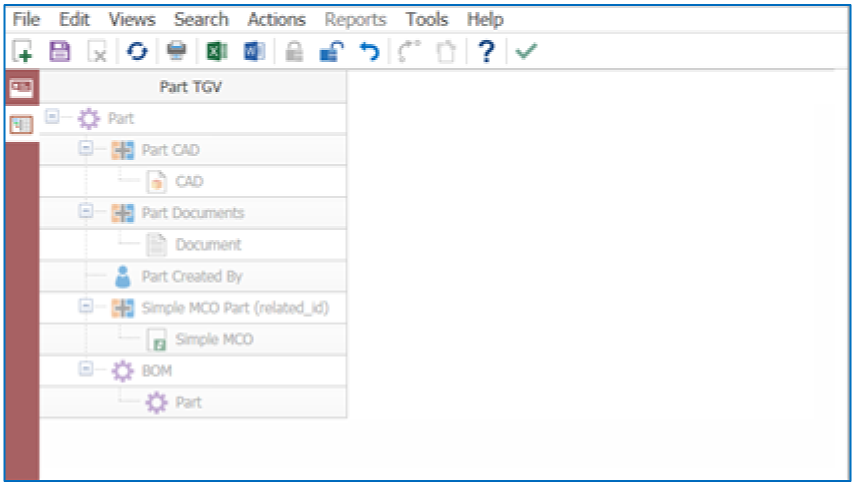


Figure 4.

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

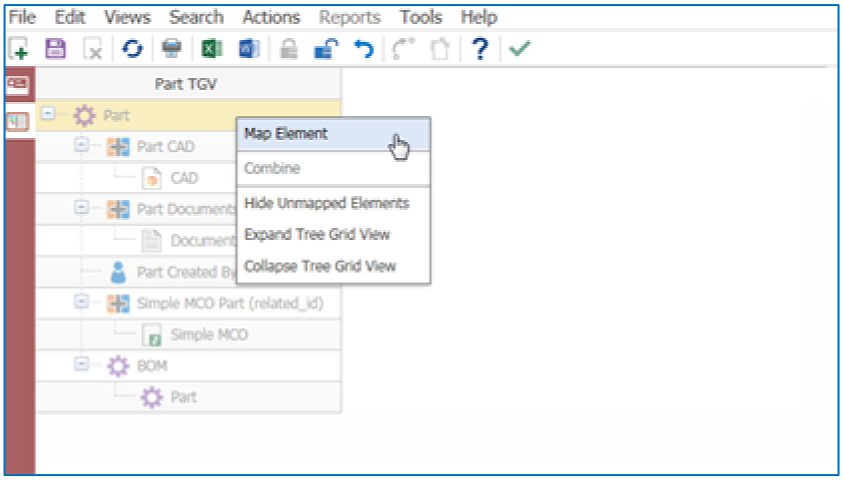


Figure 5.

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

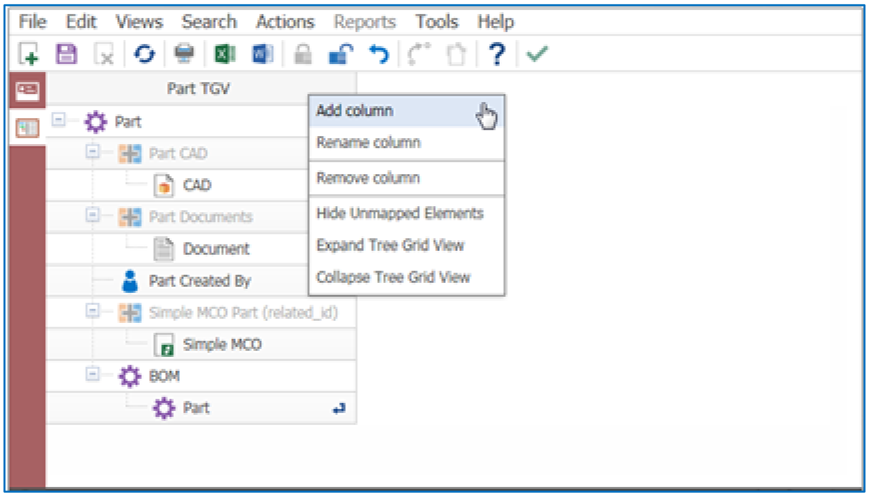


Figure 6.

5. Right-click on the new column, select **Rename Column**.and then call it **Name**.

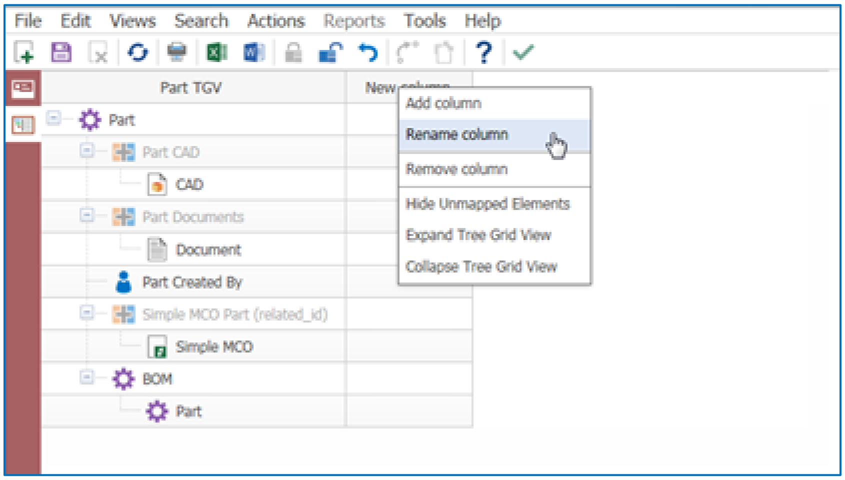


Figure 7.

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

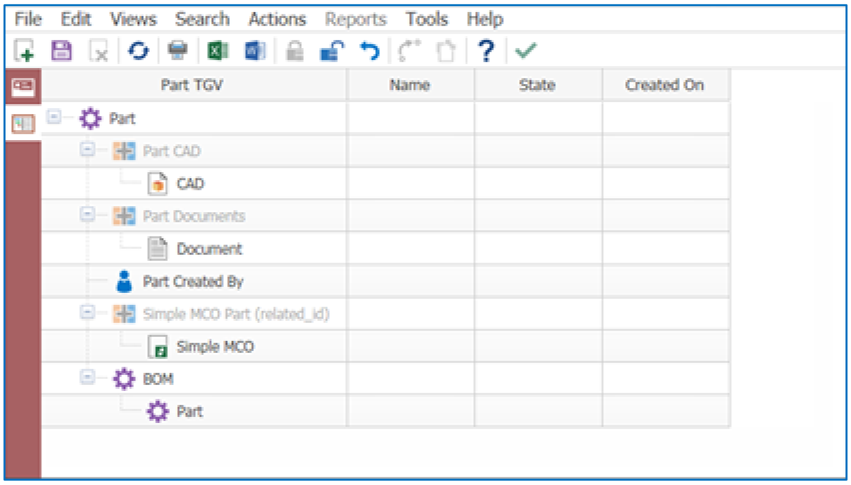


Figure 8.

7. Select both **BOM** and **Part** rows, right-click, and select **Combine**.

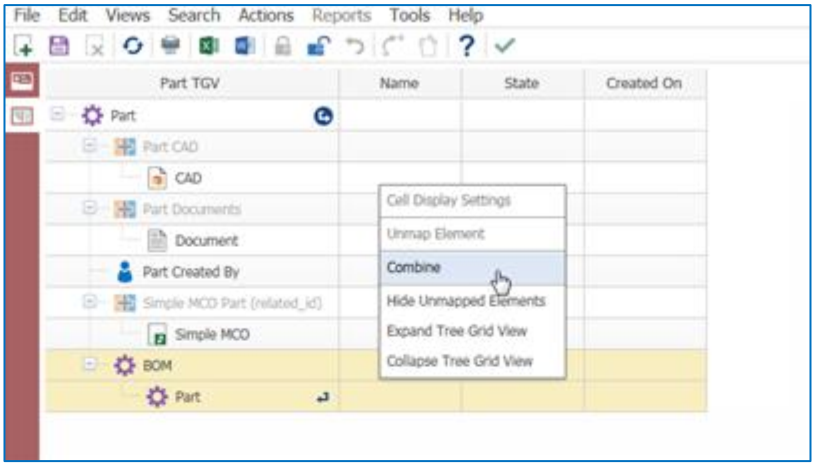


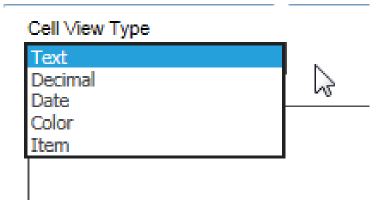
Figure 9.

Note: Combining rows let you map properties from both ItemTypes.

8. Save the Tree Grid View.

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 5 types of data:

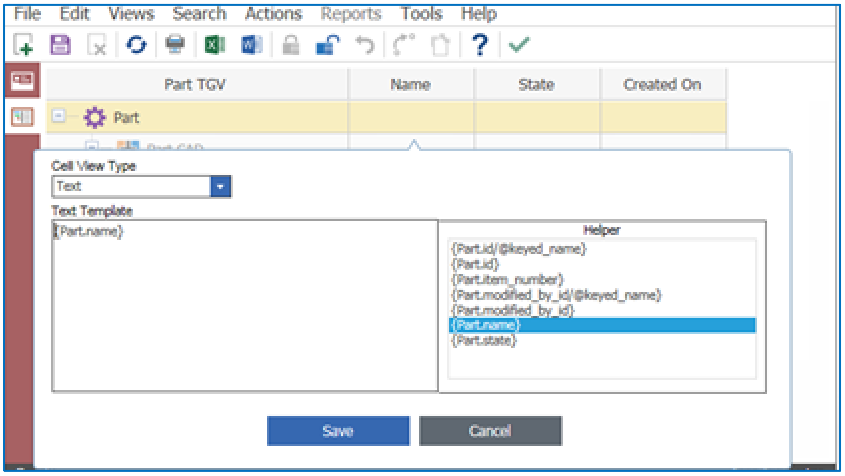


Data Type	Description
Text	Displays value as plain text
Decimal	Parses the decimal number to display decimal delimiter as either "." or ","
Date	Parses the data to display as a date
Color	Displays color
Item	Displays a hyperlink to the specified Item

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}`



Cell View Type: Text

Text Template: {Part.name}

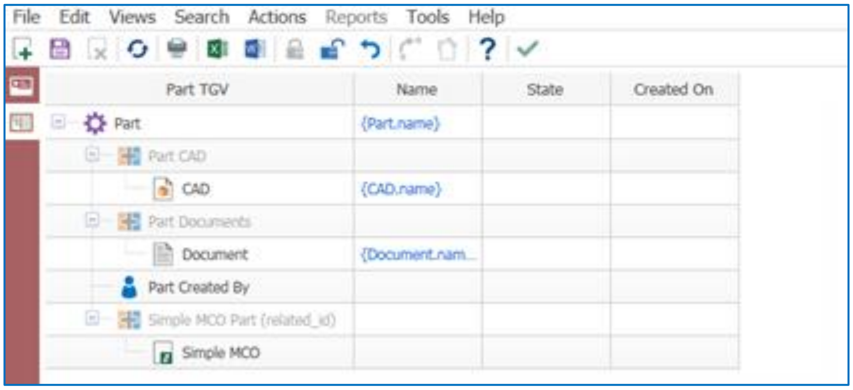
Helper

{Part.id/@keyed_name}
{Part.id}
{Part.item_number}
{Part.modified_by_id/@keyed_name}
{Part.modified_by_id}
{Part.name}
{Part.state}

Figure 10.

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

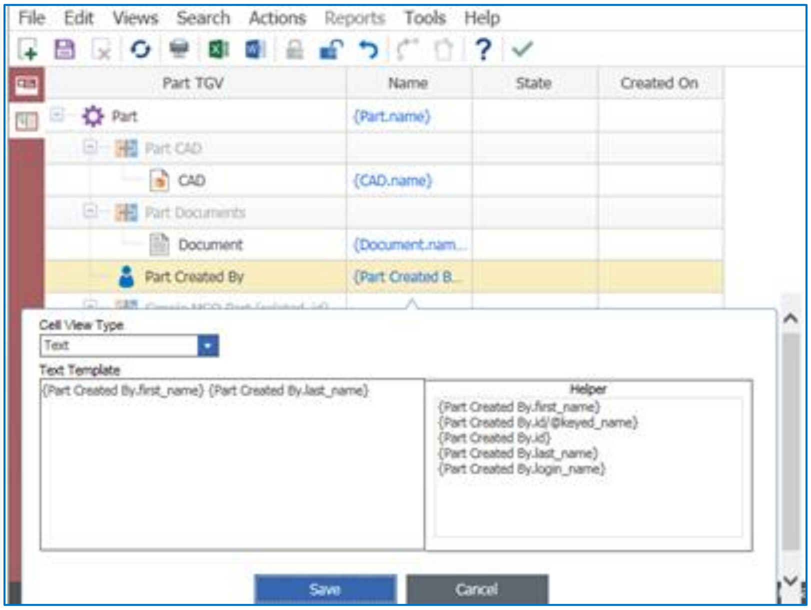
2. Double-click in the **CAD-Name** and **Document-Name** cells as well, and specify the name properties for both.



Part TGV	Name	State	Created On
Part	{Part.name}		
Part CAD			
CAD	{CAD.name}		
Part Documents			
Document	{Document.nam...		
Part Created By			
Simple MCO Part (related_id)			
Simple MCO			

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}`.



Part TGV	Name	State	Created On
Part	{Part.name}		
Part CAD			
CAD	{CAD.name}		
Part Documents			
Document	{Document.nam...		
Part Created By	{Part Created B...		

Cell View Type: Text

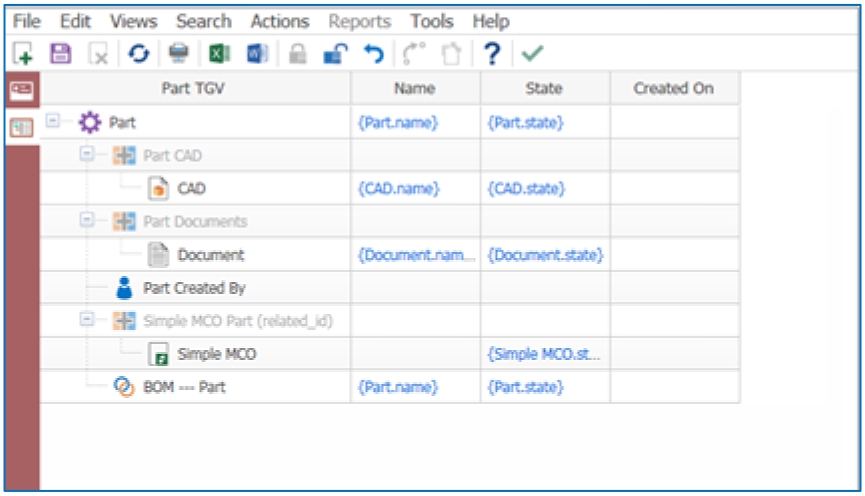
Text Template: {Part Created By.first_name} {Part Created By.last_name}

Helper

{Part Created By.first_name}
{Part Created By.id/@keyed_name}
{Part Created By.id}
{Part Created By.last_name}
{Part Created By.login_name}

Figure 12.

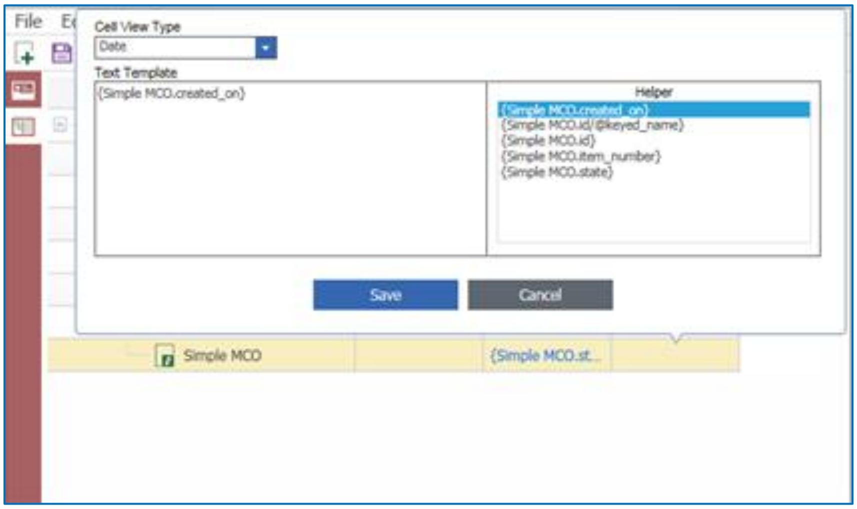
4. Map the **State** properties for Part, CAD, Document, and Simple MCO to the appropriate cells.



Part TGV	Name	State	Created On
Part	{Part.name}	{Part.state}	
Part CAD			
CAD	{CAD.name}	{CAD.state}	
Part Documents			
Document	{Document.nam...	{Document.state}	
Part Created By			
Simple MCO Part (related_id)			
Simple MCO		{Simple MCO.st...	
BOM --- Part	{Part.name}	{Part.state}	

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 Part (related_id).created_on}`



Cell View Type: Date

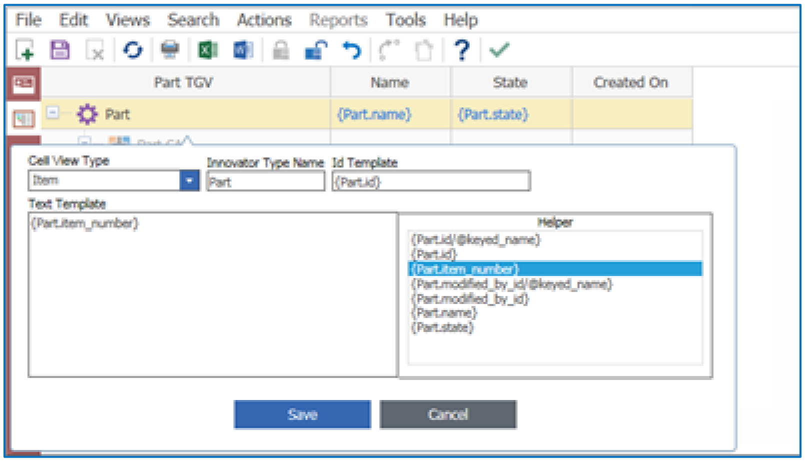
Text Template: {Simple MCO.created_on}

Helper
- {Simple MCO.created_on}
- {Simple MCO.id/@keyed_name}
- {Simple MCO.id}
- {Simple MCO.item_number}
- {Simple MCO.state}

Simple MCO | {Simple MCO.st...

Figure 14.

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



Part TGV	Name	State	Created On
Part	{Part.name}	{Part.state}	

Cell View Type: Item

Innovator Type Name: Part

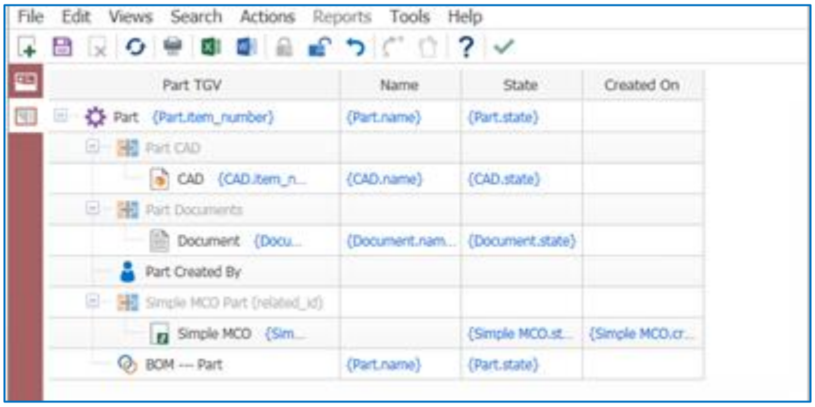
Id Template: {Part.id}

Text Template: {Part.item_number}

Helper:
{Part.id/@keyed_name}
{Part.id}
{Part.item_number}
{Part.modified_by_id/@keyed_name}
{Part.modified_by_id}
{Part.name}
{Part.state}

Figure 15.

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

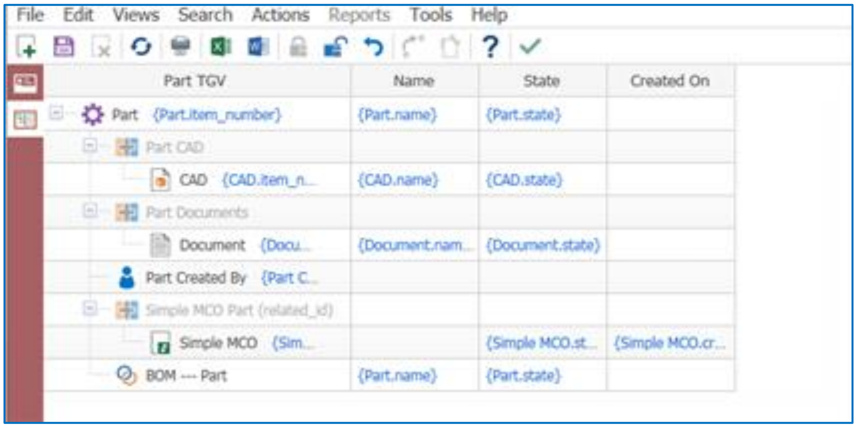


Part TGV	Name	State	Created On
Part {Part.item_number}	{Part.name}	{Part.state}	
Part CAD			
CAD {CAD.item_n...	{CAD.name}	{CAD.state}	
Part Documents			
Document {Docu...	{Document.nam...	{Document.state}	
Part Created By			
Simple MCO Part (related_id)			
Simple MCO {Sim...		{Simple MCO.st...	{Simple MCO.cr...
BOM --- Part	{Part.name}	{Part.state}	

Figure 16.

Note: Each item needs appropriate **Innovator Type Name** and **Id Template** values.

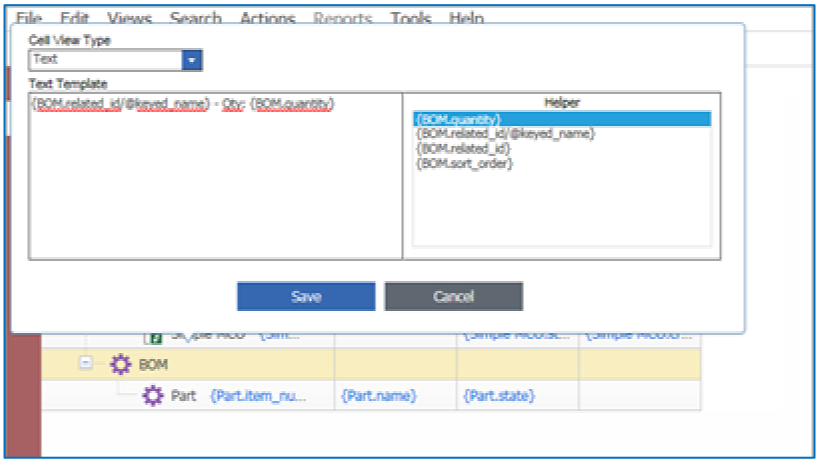
8. Repeat the previous steps to create the same link for **Part Created By-Part TGV** cell, but for text display select the `login_name` property.



Part TGV	Name	State	Created On
Part {Part.item_number}	{Part.name}	{Part.state}	
Part CAD			
CAD {CAD.item_n...	{CAD.name}	{CAD.state}	
Part Documents			
Document {Docu...	{Document.nam...	{Document.state}	
Part Created By {Part C...			
Simple MCO Part (related_id)			
Simple MCO {Sim...		{Simple MCO.st...	{Simple MCO.cr...
BOM --- Part	{Part.name}	{Part.state}	

Figure 17.

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



Cell View Type: Text

Text Template: {BOM.related_id/@keyed_name} - Qty: {BOM.quantity}

Helper:
{BOM.quantity}
{BOM.related_id/@keyed_name}
{BOM.related_id}
{BOM.sort_order}

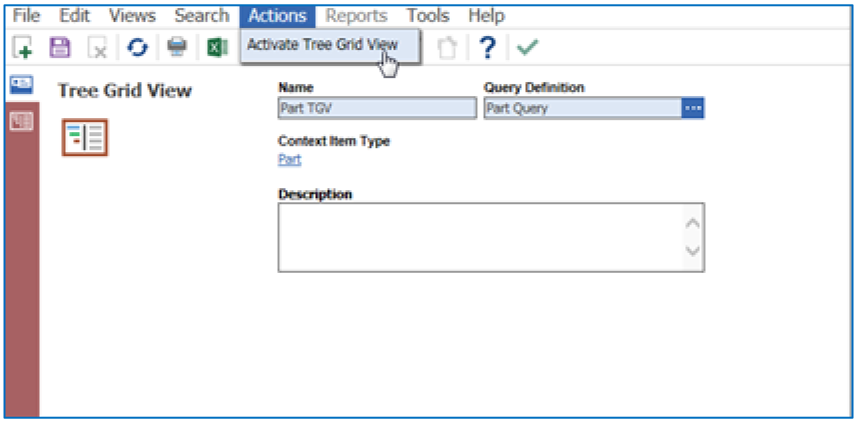
Figure 18.

10. Save the Tree Grid View.

Chapter 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.

To create the RelationshipType, select the Tree Grid View and run the **Activate Tree Grid View** Action.



Name
Part TGV

Query Definition
Part Query

Context Item Type
Part

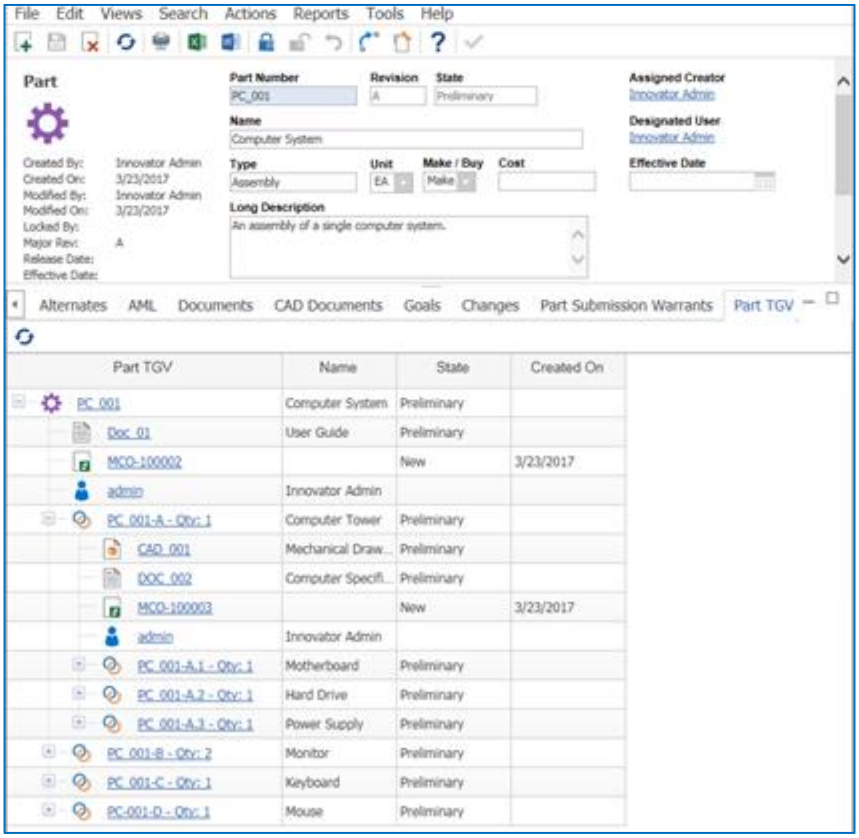
Description

Figure 19.

Only run the action once. Running the action a second time would result 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.

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 Chapter 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.



Part

Created By: Innovator Admin
Created On: 3/23/2017
Modified By: Innovator Admin
Modified On: 3/23/2017
Locked By:
Major Rev: A
Release Date:
Effective Date:

Part Number: PC_001
Revision: A
State: Preliminary
Name: Computer System
Type: Assembly
Unit: EA
Make / Buy: Make
Cost:
Long Description: An assembly of a single computer system.

Assigned Creator: Innovator Admin
Designated User: Innovator Admin
Effective Date:

Part TGV	Name	State	Created On
PC_001	Computer System	Preliminary	
Doc_01	User Guide	Preliminary	
MCO-100002		New	3/23/2017
admin	Innovator Admin		
PC_001-A - Qty: 1	Computer Tower	Preliminary	
CAD_001	Mechanical Draw...	Preliminary	
DOC_002	Computer Specifi...	Preliminary	
MCO-100003		New	3/23/2017
admin	Innovator Admin		
PC_001-A.1 - Qty: 1	Motherboard	Preliminary	
PC_001-A.2 - Qty: 1	Hard Drive	Preliminary	
PC_001-A.3 - Qty: 1	Power Supply	Preliminary	
PC_001-B - Qty: 2	Monitor	Preliminary	
PC_001-C - Qty: 1	Keyboard	Preliminary	
PC-001-D - Qty: 1	Mouse	Preliminary	

Figure 20.

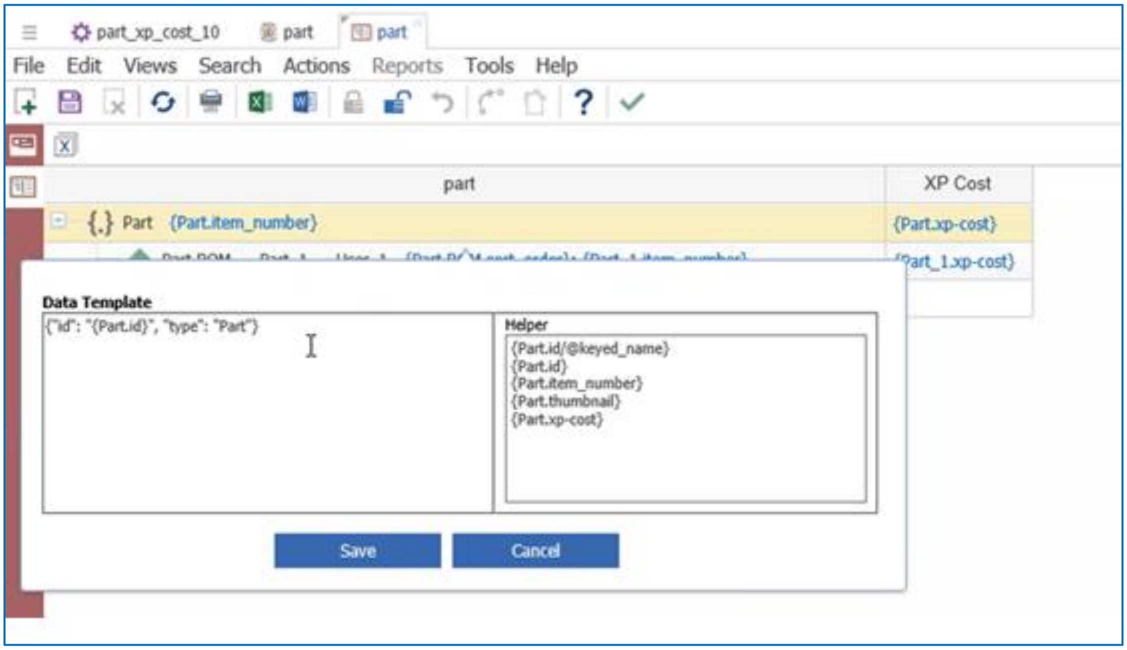
Chapter 4. Configuration Examples

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.

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:



part	XP Cost
Part {Part.item_number}	{Part.xp-cost}
	{Part_1.xp-cost}

Data Template

```
{"id": "{Part.id}", "type": "Part"}
```

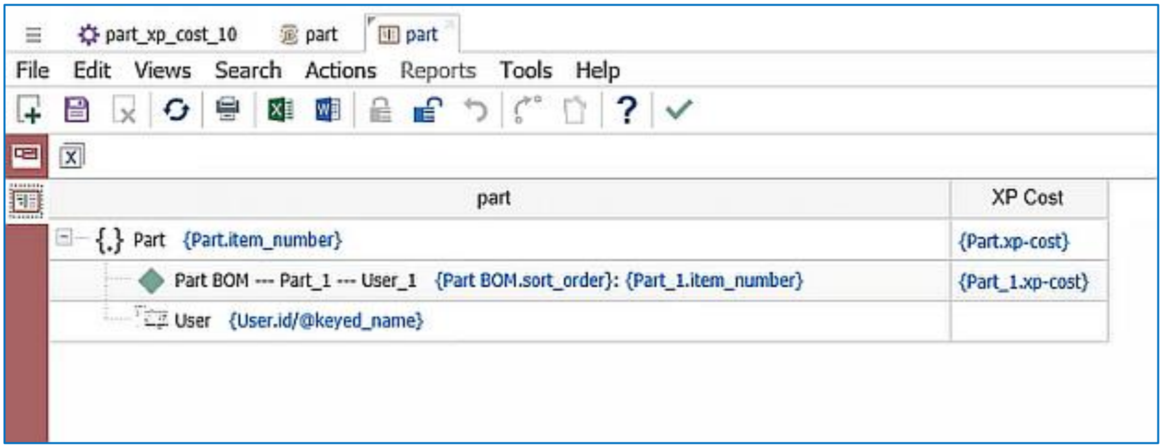
Helper

```
{Part.id/@keyed_name}
{Part.id}
{Part.item_number}
{Part.thumbnail}
{Part.xp-cost}
```

Figure 21.

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:



part	XP Cost
{.} Part {Part.item_number}	{Part.xp-cost}
Part BOM --- Part_1 --- User_1 {Part BOM.sort_order}: {Part_1.item_number}	{Part_1.xp-cost}
User {User.id/@keyed_name}	

Figure 22.

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:

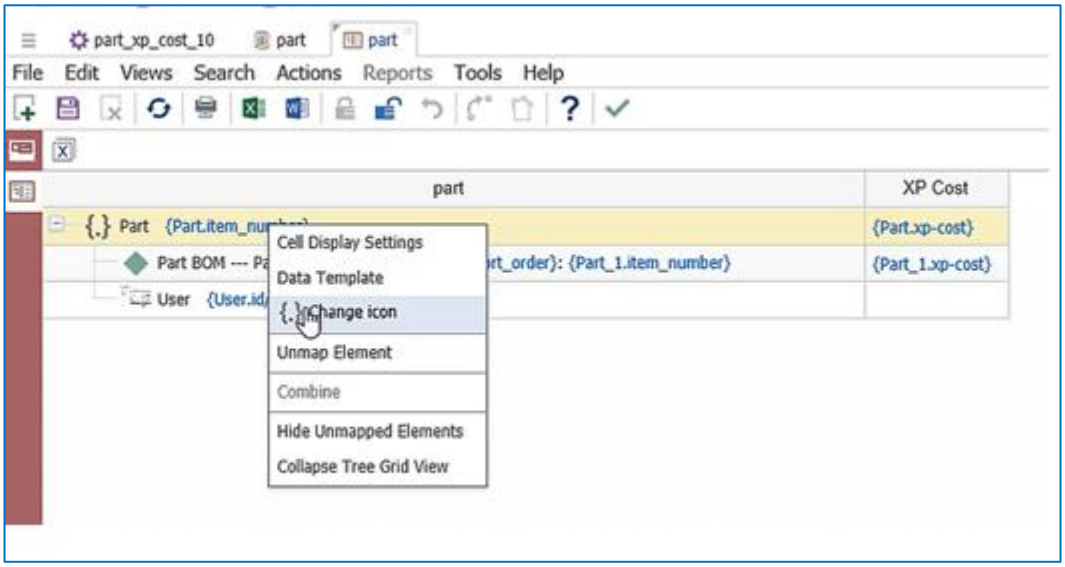


Figure 23.

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

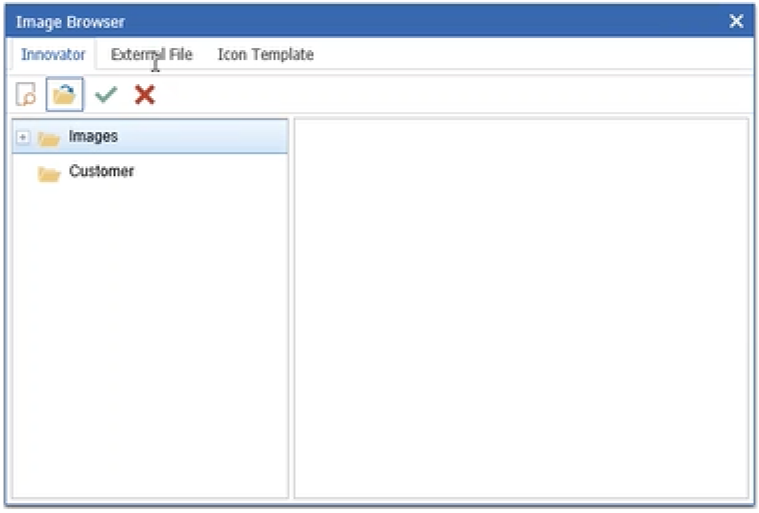
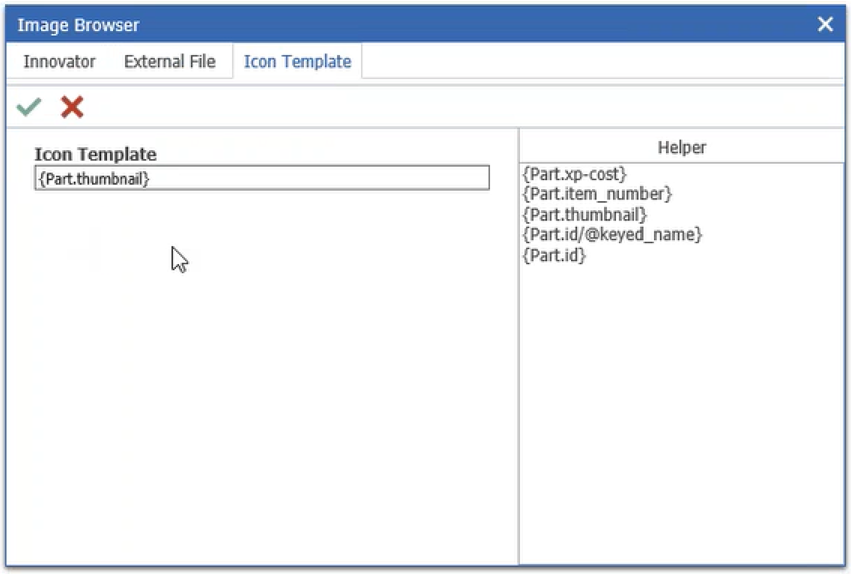


Figure 24.

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



Icon Template: {Part.thumbnail}

{Part.xp-cost}
{Part.item_number}
{Part.thumbnail}
{Part.id/@keyed_name}
{Part.id}

Figure 25.

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.

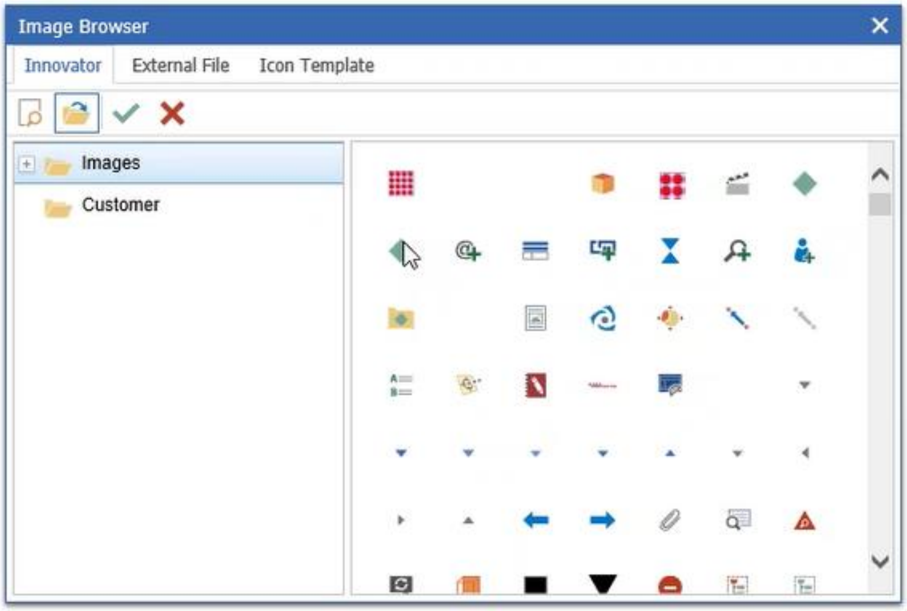


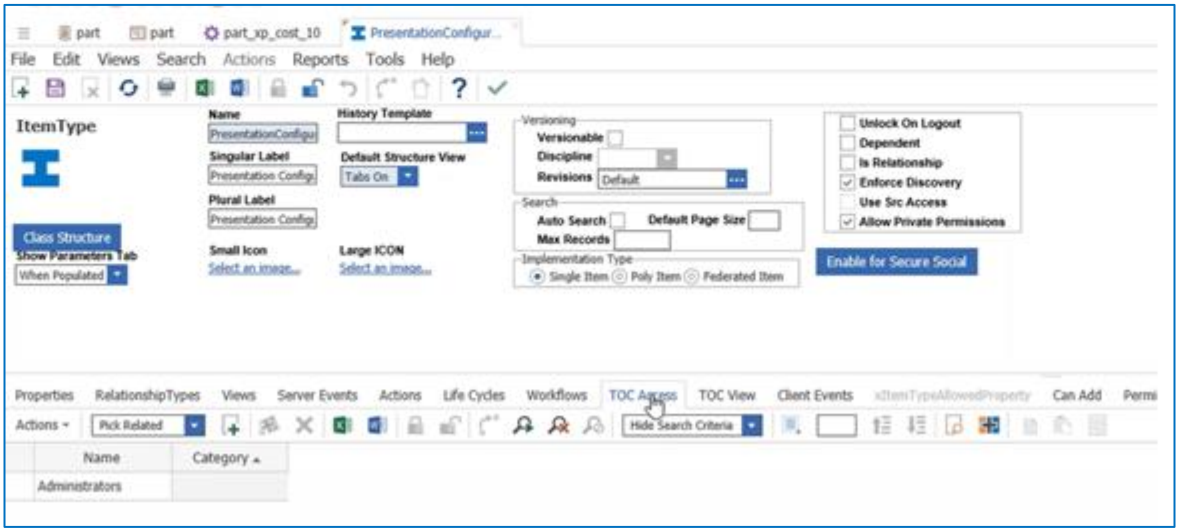
Image Browser
Innovator
External File
Icon Template
Images
Customer

Figure 26.

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:



Name: PresentationConfigu
History Template:
Singular Label: Presentation Config
Default Structure View: Tabs On
Plural Label: Presentation Config
Show Parameters Tab: When Populated
Small Icon: Select an image...
Large ICON: Select an image...

Versioning
Versionable □
Discipline
Revisions: Default

Search
Auto Search □
Default Page Size
Max Records

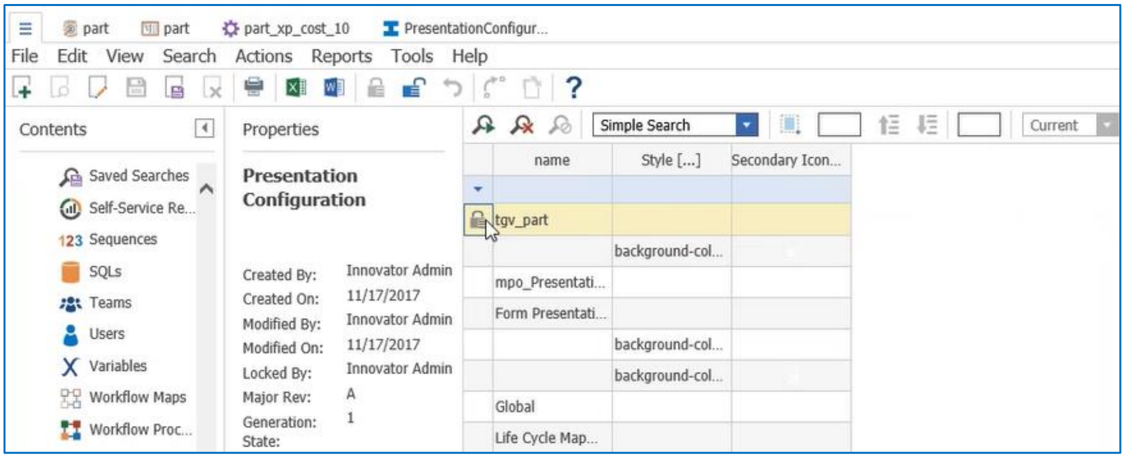
Implementation Type
● Single Item ○ Poly Item ○ Federated Item

□ Unlock On Logout
□ Dependent
□ Is Relationship
☑ Enforce Discovery
Use Src Access
☑ Allow Private Permissions

Name	Category
Administrators	

Figure 27.

2. Expand Templates in the TOC and select Presentation Configuration. The Presentation Configuration grid appears.



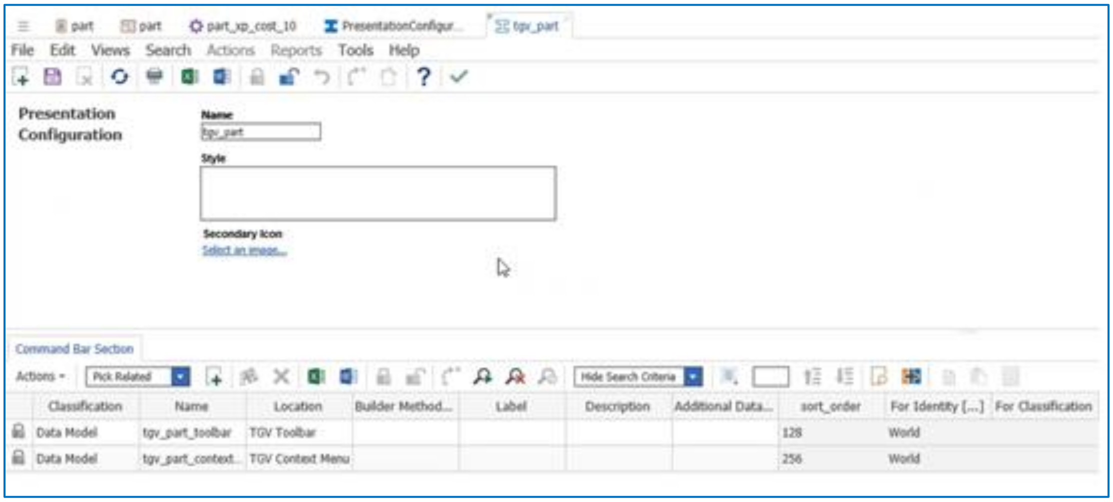
Presentation Configuration

Created By: Innovator Admin
Created On: 11/17/2017
Modified By: Innovator Admin
Modified On: 11/17/2017
Locked By: Innovator Admin
Major Rev: A
Generation: 1
State:

name	Style [...]	Secondary Icon...
tgv_part		
	background-col...	
mpo_Presentati...		
Form Presentati...		
	background-col...	
	background-col...	
Global		
Life Cycle Map...		

Figure 28.

3. Select **tgv_part** from the grid.



Name: tgv_part

Style

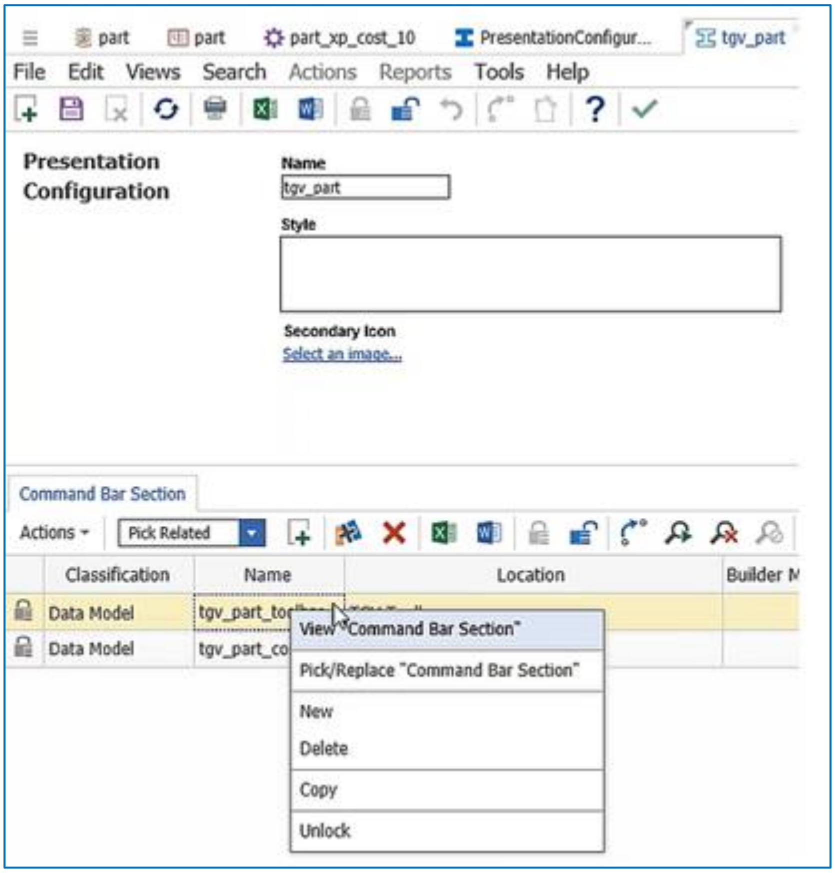
Secondary Icon: Select an image...

	Classification	Name	Location	Builder Method...	Label	Description	Additional Data...	sort_order	For Identity [...]	For Classification
	Data Model	tgv_part_toolbar	TGV Toolbar					128	World	
	Data Model	tgv_part_context...	TGV Context Menu					256	World	

Figure 29.

The tgv_part presentation configuration item contains TGV Toolbar and TGV Context Menu.

4. Right click **tgv_part_toolbar** and select **View "Command Bar Section"** from the context menu:



Name: tgv_part

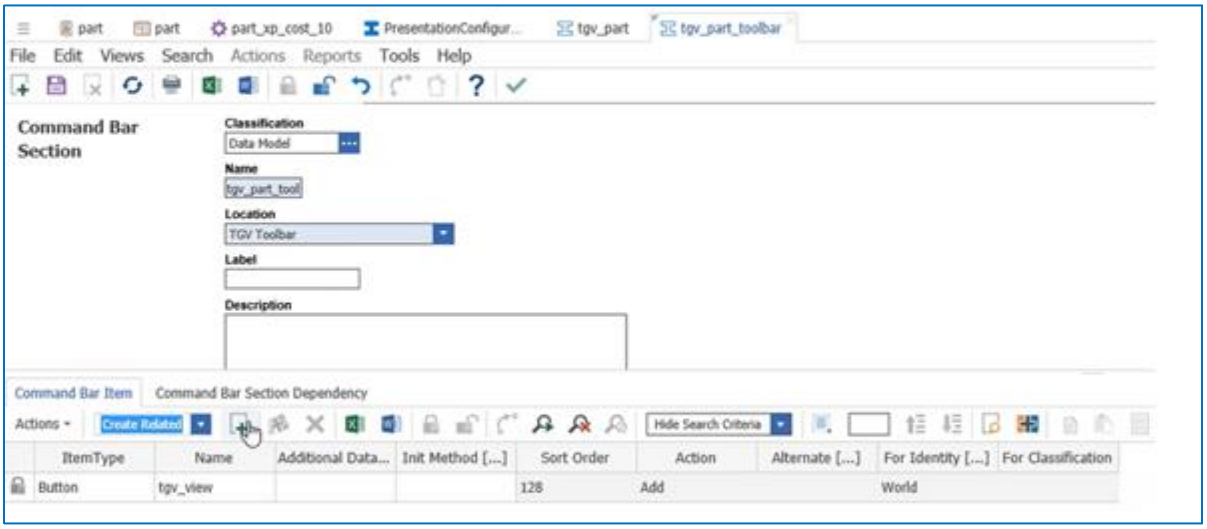
Style

Secondary Icon: Select an image...

	Classification	Name	Location	Builder M
	Data Model	tgv_part_to		
	Data Model	tgv_part_co		

Figure 30.

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



Command Bar Section

Classification: Data Model

Name: tgv_part_tool

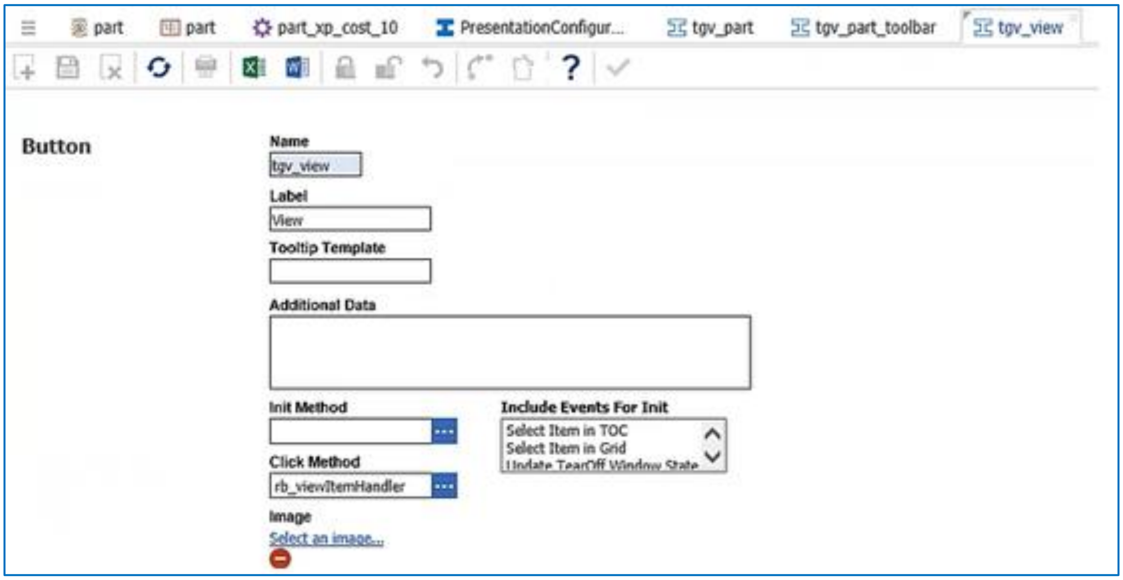
Location: TGV Toolbar

Label:

Description:

ItemType	Name	Additional Data...	Init Method [...]	Sort Order	Action	Alternate [...]	For Identity [...]	For Classification
Button	tgv_view			128	Add		World	

5. Right click Button in the grid and select “View Command Bar Item.” The tgv_view presentation item appears:



Button

Name: tgv_view

Label: View

Tooltip Template:

Additional Data:

Init Method:

Include Events For Init: Select Item in TOC, Select Item in Grid, Update TearOff Window State

Click Method: rb_viewItemHandler

Image:

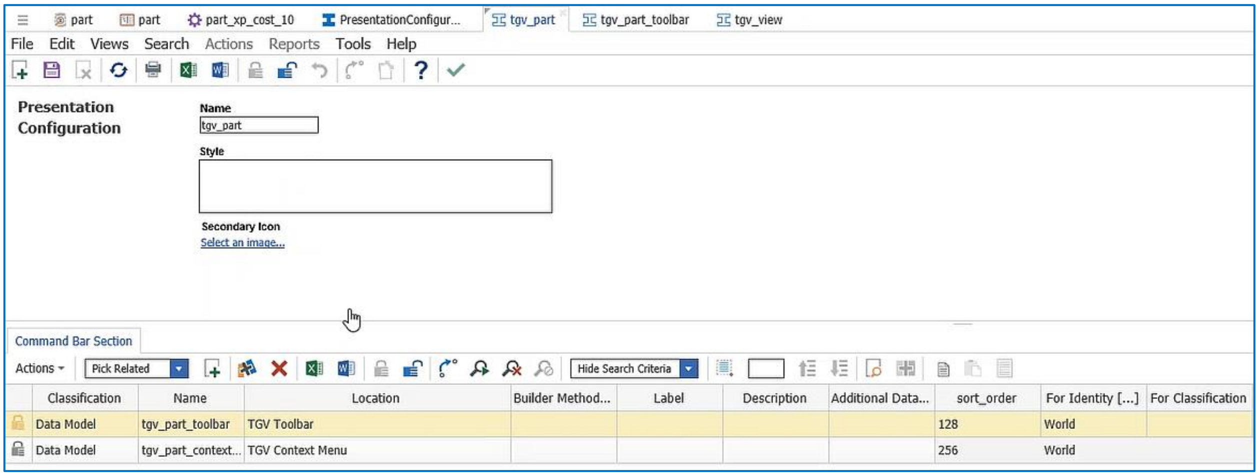
Figure 31.

Users can review the rb_viewItemHandler method code and develop similar code for their own uses.

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 and select **tgv_part_context** from the Command Bar section grid:



Presentation Configuration

Name: tgv_part

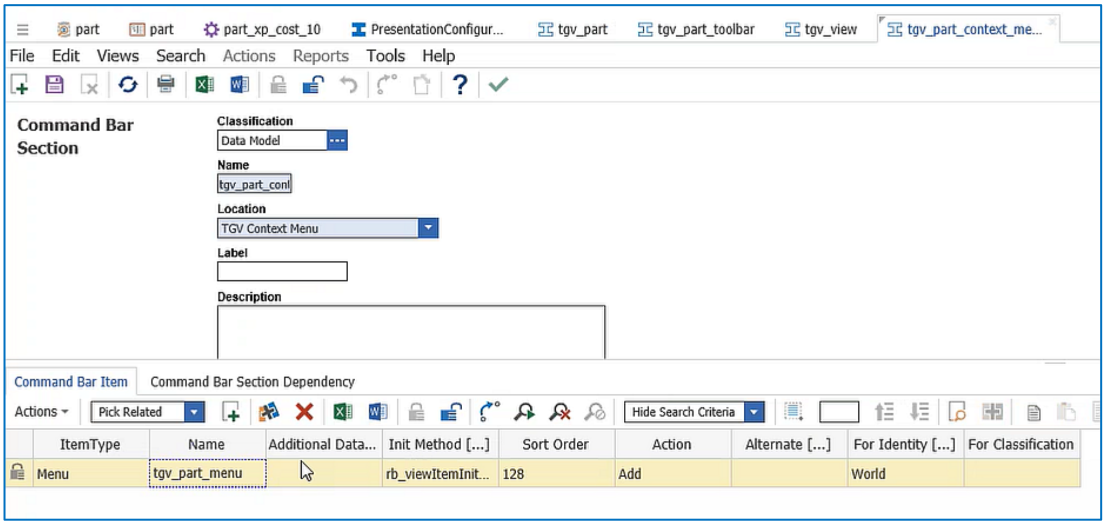
Style

Secondary Icon: Select an image...

Classification	Name	Location	Builder Method...	Label	Description	Additional Data...	sort_order	For Identity [...]	For Classification
Data Model	tgv_part_toolbar	TGV Toolbar					128	World	
Data Model	tgv_part_context...	TGV Context Menu					256	World	

Figure 32.

2. Select TGV Context Menu from the Command Bar grid.



Command Bar Section

Classification: Data Model

Name: tgv_part_cont

Location: TGV Context Menu

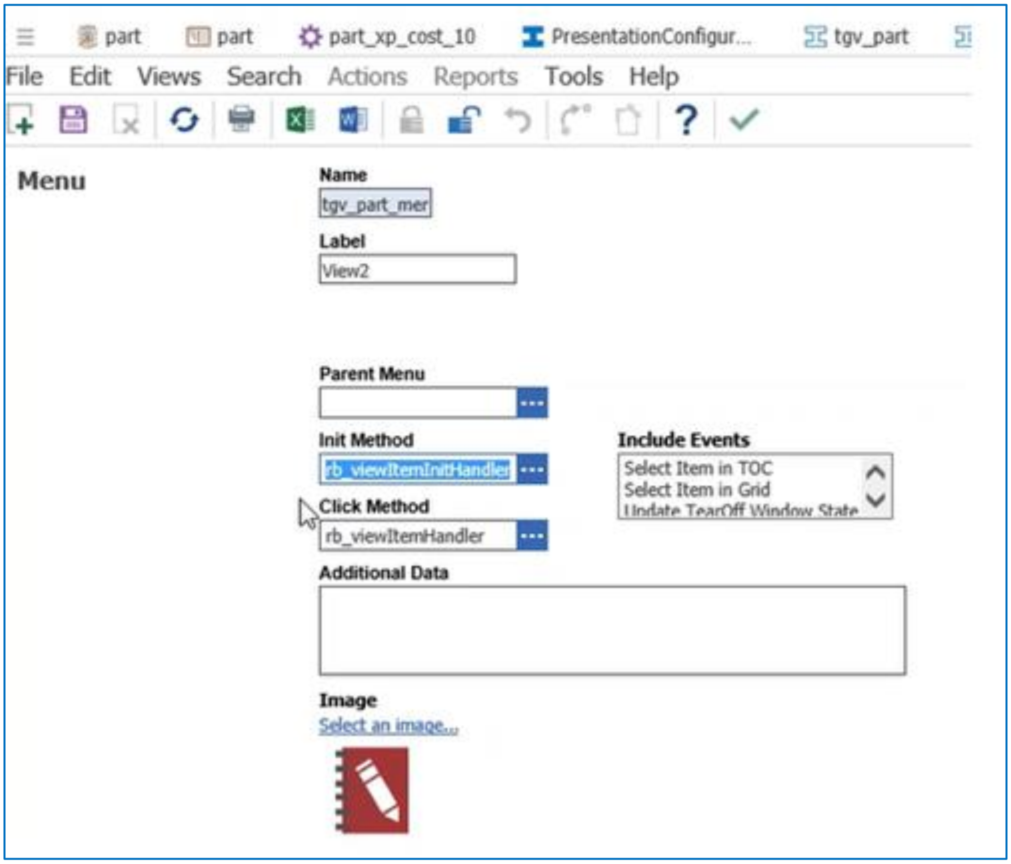
Label

Description

ItemType	Name	Additional Data...	Init Method [...]	Sort Order	Action	Alternate [...]	For Identity [...]	For Classification
Menu	tgv_part_menu		rb_viewItemInit...	128	Add		World	

Figure 33.

3. Right click tgv_part_menu in the Command Bar Item grid and select “View Command Bar Item.” The tgv_part_menu presentation item appears:



Name: tgv_part_mer

Label: View2

Parent Menu:

Init Method: rb_viewItemInitHandler

Click Method: rb_viewItemHandler

Include Events:
Select Item in TOC
Select Item in Grid
Update TearOff Window State

Additional Data:

Image: Select an image...

Figure 34.

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