

Configuration Examples

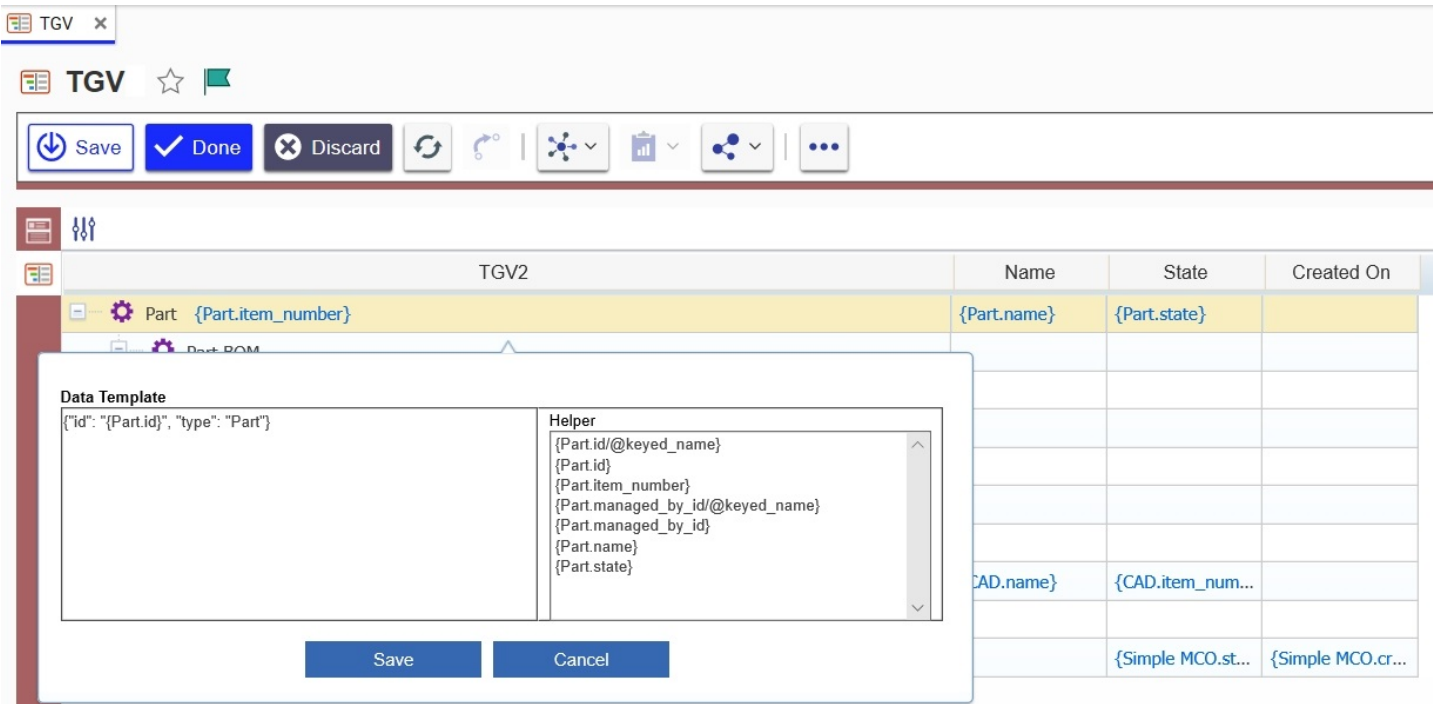
This section contains examples of how to configure buttons, context menus, toolbars, and Tree Grid rows to use in Tree Grid View. The configuration items in this section are examples of advanced functionality.

Important

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:



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 B		XP Cost
Part	{Part.item_number}	{Part.xp-cost}
Part BOM	{Part BOM.sort_order}	
Part_1	{Part_1.item_number}	{Part_1.xp-cost}
User	{User.keyed_name}	

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

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

TGV x

TGV ☆

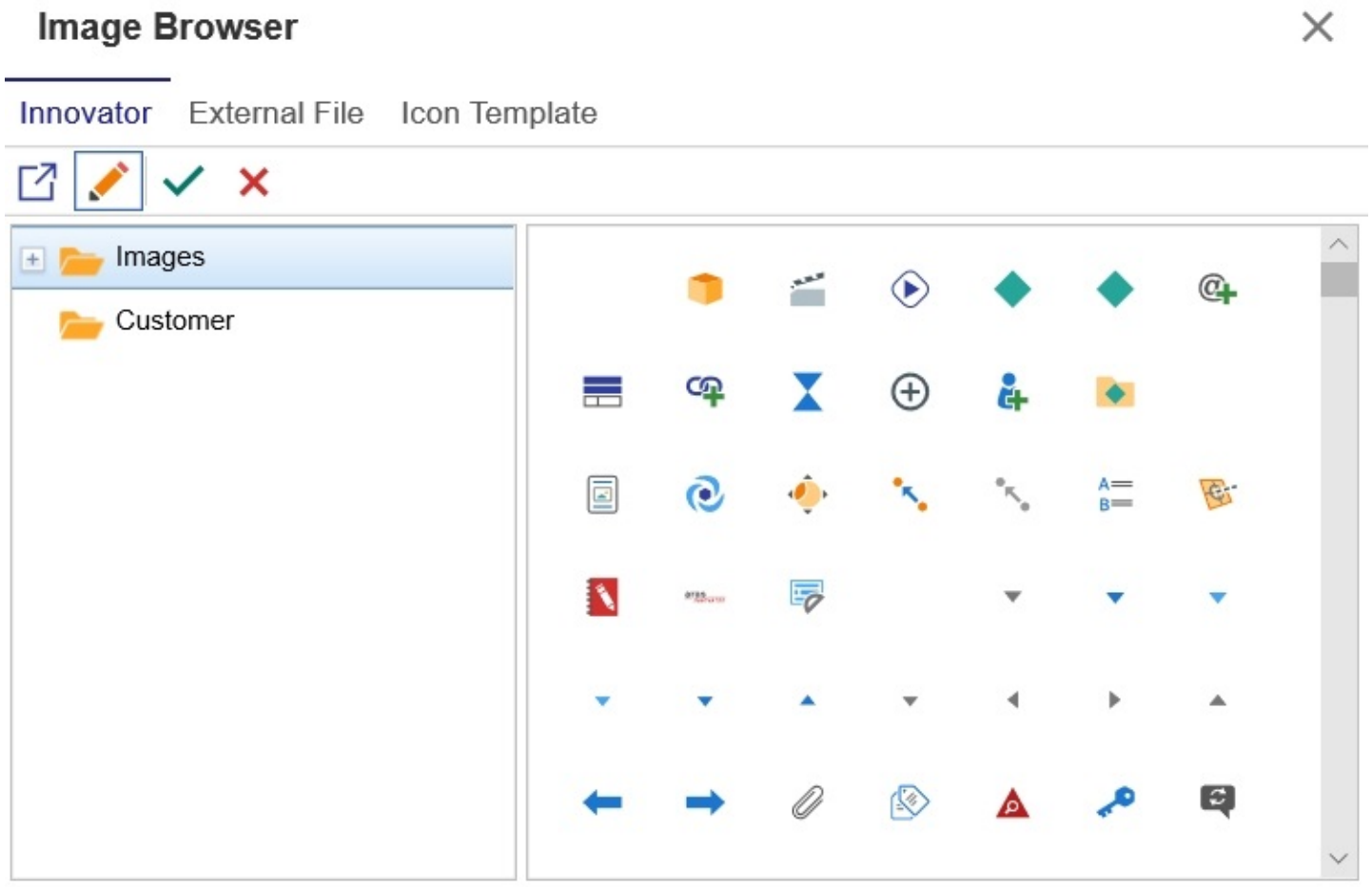
Save Done Discard

TGV2		Name	State	Created On
Part	{Part.item_number}	{Part.name}	{Part.state}	
Part BOM				
Part_1				
Part Document				
Document				
User				
Part CAD				
CAD		{CAD.name}	{CAD.item_num...}	
Simple MCO Part related_id				
Simple MCO			{Simple MCO.st...}	{Simple MCO.cr...}

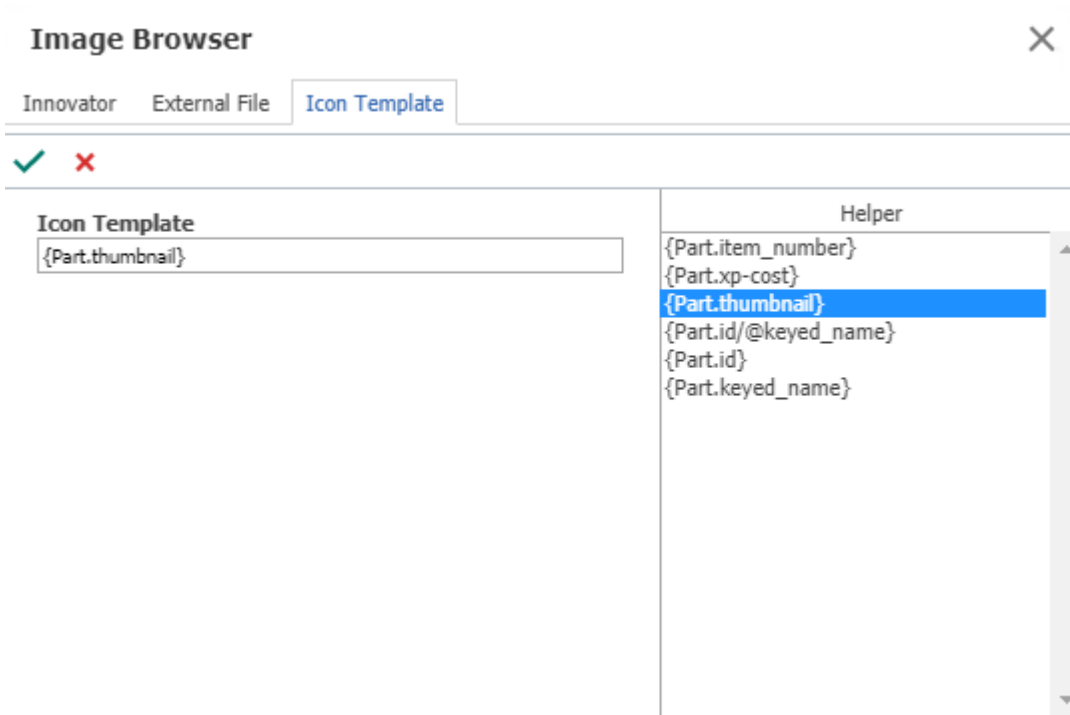
Cell Display Settings
Data Template
Change icon
Unmap Element
Combine
Show Unmapped Elements
Collapse Tree Grid View

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



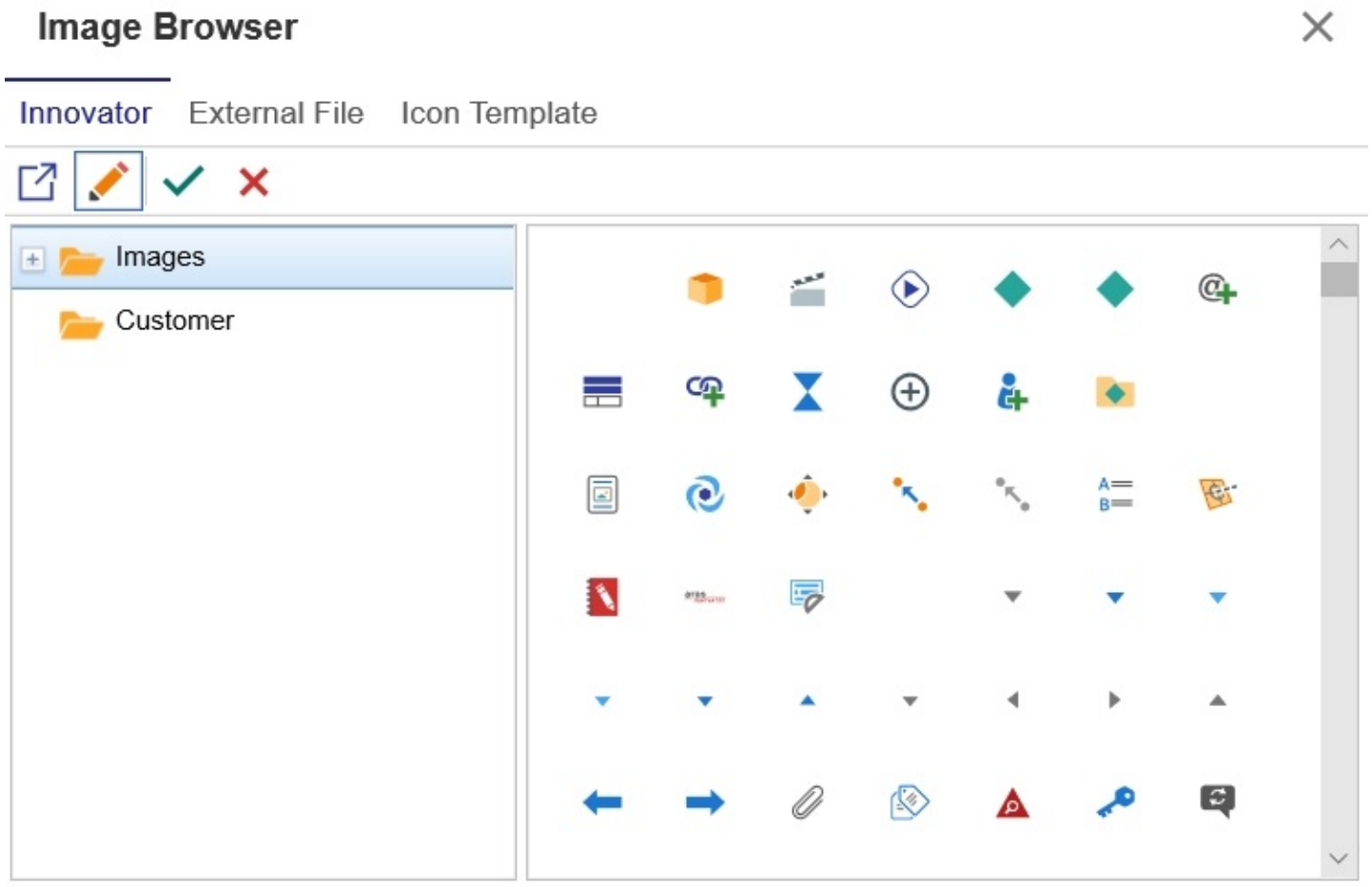


3. Click **Icon Template** and then click **{Part.thumbnail}** in the **Helper** area to change to the **{.}** icon.



4. Return to the **Part** Grid and right-click the **Part BOM** row.
5. Click **Change icon** in the context menu. The **Image Browser** dialog appears.
6. Click **Innovator** and then click an icon.





Configuring a Toolbar Button

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

1. Click the **PresentationConfiguration** ItemType in the grid, click the **TOC Access** tab, and add **Administrators**.

PresentationCo... x

PresentationConfiguration ☆

Save Done Discard ↺ ↻ 🔗 📊 🔗 ⋮

^ ItemType

Name

Singular Label

Show Parameters Tab

Class Structure

History Template

Plural Label

Default Structure View

Small Icon

[Select an image...](#)

Versioning

Versionable ☐

Discipline

Revisions

Large Icon

[Select an image...](#)

Search

Auto Search ☐

Default Page Size

Max Records

☐ Unlock On Logout

☐ Dependent

☐ Is Relationship

☒ Enforce Discovery

☐ Use Src Access

☒ Allow Private Permissions

Properties RelationshipTypes Views Server Events Actions Life Cycles Workflows **TOC Access** TOC View

Identities ☆

⚙️ ⚙️ ✖️ 🔍 🔍 Hidden 🔍 🖨️ 🔍 🔗

Name	Category ↑
Administrators	

2. Open the TOC and click **Presentation Configuration**.

← Presentation Configurations ☆

⚙️ Create New Presentation Configuration

🔍 Search Presentation Configurations

3. Click **CreateNew Presentation Configuration**. The **Presentation Configuration** form appears.

4. Enter **tgvs_part** in the **Name** field and select a color such as orange.

tgvs_part x

tgvs_part ☆

Edit ↺ ↻ ⚙️ 📊 🔗 ⋮

^ Presentation Configuration

Color
[Orange]

Name
tgvs_part

^ Command Bar Section cui_PresentConfigWinSection

CommandBarSection ☆

⚙️ ⚙️ ⚙️ | 🔍 ⚙️ Hidden ⚙️ ⚙️ ⚙️

	Classification	Name	Builder Meth...	Label	Description	Additional Da...	Location [...]	sort_order	For Identity [...]	For Classifica...
--	----------------	------	-----------------	-------	-------------	------------------	----------------	------------	--------------------	-------------------

5. Click the New Command Bar null in the Command Bar Section relationship tab to add a new related Item.

6. Enter **tgvs*** in the **Name** cell of the **Search** grid and add the **tgvs_part_toolbar** and **tgvs_part_ContextMenu** Items:

^ Command Bar Section cui_PresentConfigWinSection

CommandBarSection ☆

⚙️ ⚙️ ⚙️ | 🔍 ⚙️ Hidden ⚙️ ⚙️ ⚙️

	Classification	Name	Builder Meth...	Label	Description	Additional Da...	Location [...]	sort_order	For Identity [...]	For Classifica...
⚙️	Data Model	tgvs_part_toolb...		tgvs_part_toolb...					Administrators	
⚙️	Data Model	tgvs_part_Cont...		tgvs_part_Cont...					Administrators	

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



tgvp_part

Presentation Configuration

Color

Name

Command Bar Section | cui_PresentConfigWinSection

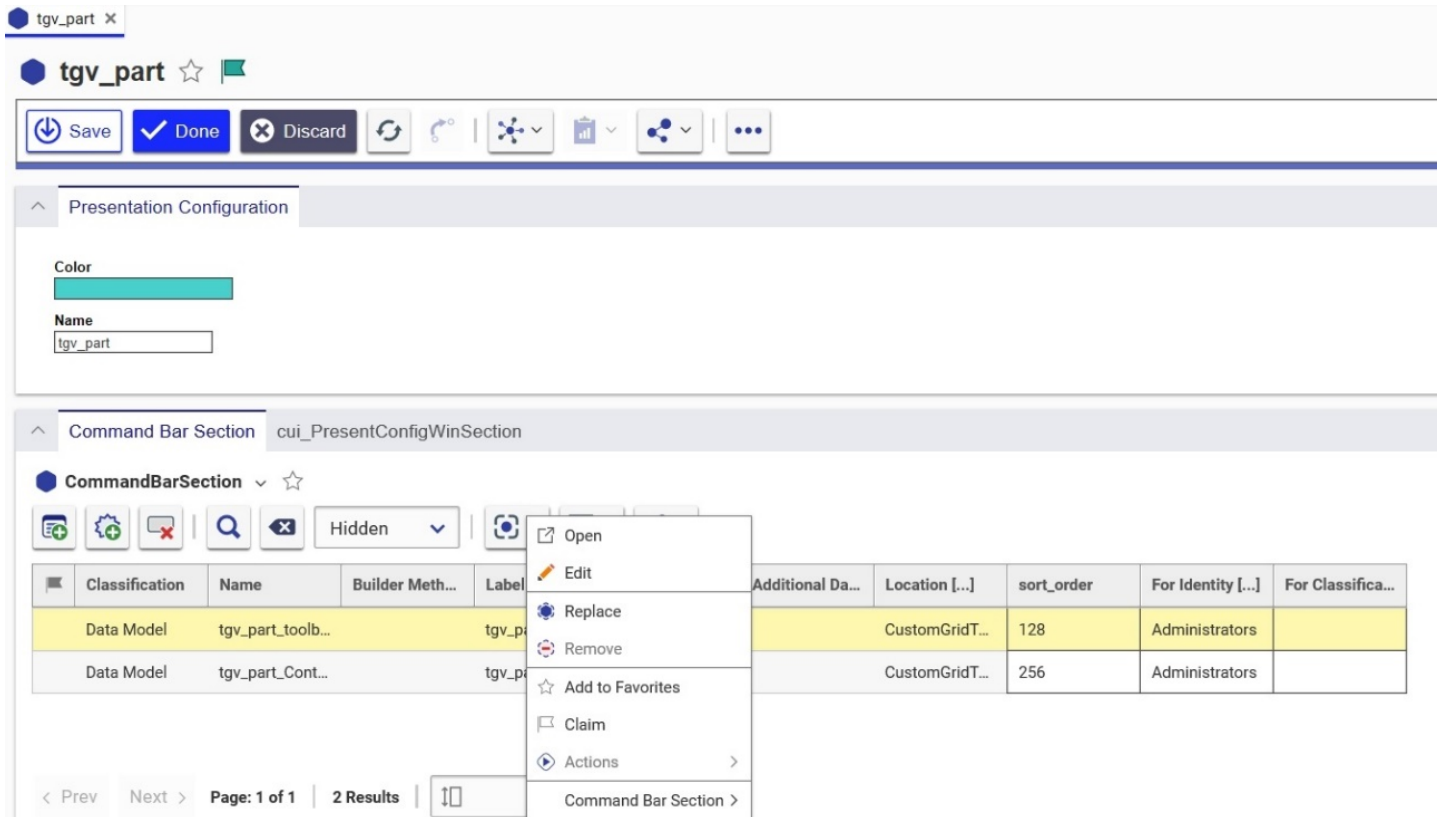
CommandBarSection

|

 Hidden ▾
 |
 ▾
 ▾
 ▾

Classification	Name	Builder Meth...	Label	Description	Additional Da...	Location [...]	sort_order	For Identity [...]	For Classifica...
Data Model	tgvp_part_toolb...		tgvp_part_toolb...			CustomGridT...	128	Administrators	
Data Model	tgvp_part_Cont...		tgvp_part_Cont...			CustomGridT...	256	Administrators	

8. Add an identity to the **For Identity** column. The specified Identity determines which users have access to the buttons and menus.
9. Click the null button to save the `tg_v_part` Presentation Configuration Item.
10. Right-click **tg_v_part_toolbar** and select **Open** from the context menu:



The `tgview_part_toolbar` presentation configuration item appears. It contains the `tgview_view` button.

Select Item Type



Select the Type of Item that is to be created:



- ☐ Button
- ☐ Separator
- ☐ Menu
- ☐ Menu Button
- ☐ Menu Checkbox
- ☐ DropDown
- ☐ Checkbox
- ☐ Shortcut
- ☐ Edit
- ☐ Menu Separator

OK

Cancel

12. Select **Button** and click **OK** to add it to the command bar.
13. Right-click the Item to access the context menu and click **Open**. The **tgview** Presentation Item appears.



The screenshot shows a web-based configuration interface for a toolbar. At the top, there is a tab labeled 'tgv_view' with a close button. Below the tab is a header bar with the text 'tgv_view' and a star icon. A toolbar is displayed with several icons: a blue 'Edit' button, a refresh icon, a circular arrow icon, a network icon, a bar chart icon, a share icon, and a more options icon. Below the toolbar, there is a section titled 'Button' with a dropdown arrow. This section contains several input fields: 'Name' (containing 'tgv_view'), 'Label' (containing 'View'), 'ARIA Label' (empty), 'Tooltip Template' (empty), and 'Additional Data' (empty). At the bottom, there are two sections: 'Init Method' and 'Include Events'. The 'Include Events' section has a dropdown menu with 'Select Item in TOC' selected.

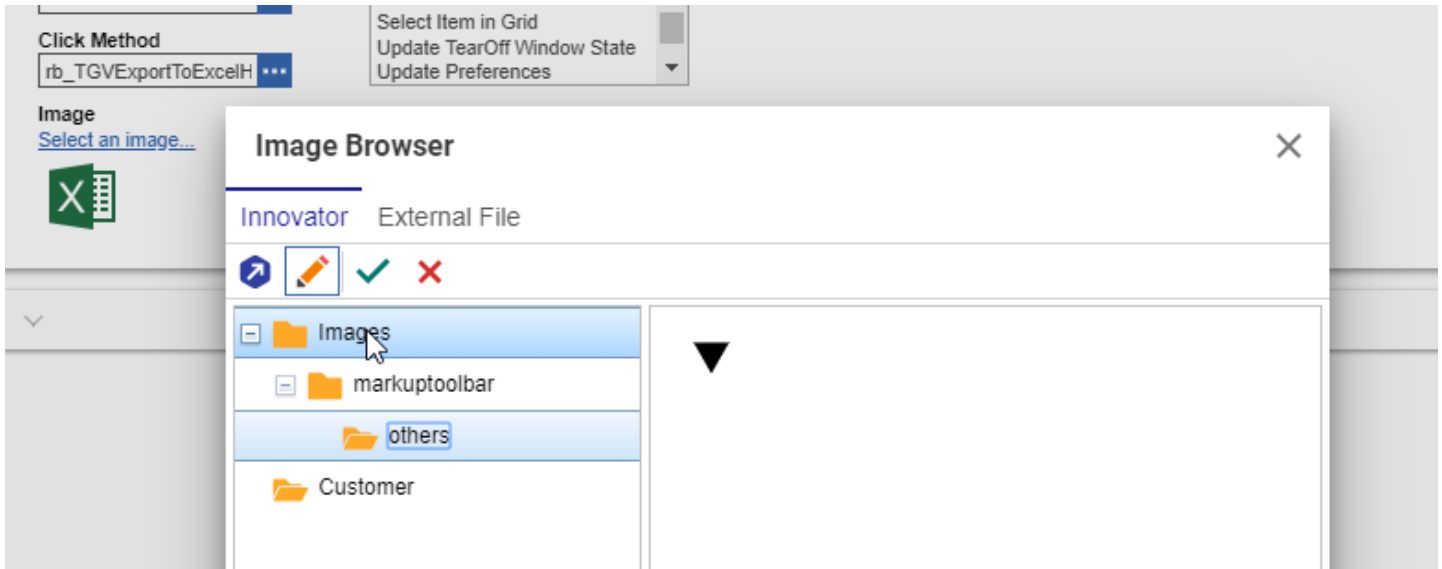
Additional logic can be added to the context menu or toolbar items when they are initialized by adding an **Init Method**. Users can review the **rb_viewItemHandler** Method code and develop similar code for their own uses.

Adding an Image to a Button

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

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





2. Select the appropriate image and click the null button.
3. Click null to save the Presentation Configuration.

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. Go to **Contents --> Administration --> Configuration --> Tree Grid Views** and click the **Search** button. A list of Tree Grid views appears in the search grid.
2. Click **Part TGV**.

Part TGV x

Part TGV ☆

Edit

Tree Grid View

Name

Part TGV

Query Definition

[Part Query](#)

Context Item Type

[Part](#)

Description

Max Visible Children On Expand

100

Max Grow Levels

2

Auto Grow On Refresh

☐

Linked Toolbar/Context Menu

Important

Make sure that the Item is locked before continuing.

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



Select Items

Presentation Configurations

name	Color [...]
my process ...	
ReplicationT...	
SPDocument...	
Forum_TOC...	
Workflow Ma...	
gn_ViewCard...	
Variable_TO...	
dac_Present...	
PackageDefi...	
cmf_Style_T...	
tgw_part	
SystemEvent...	
Permission ...	

< Prev
Next >
Page: 1 of 1
120 Results

OK Cancel

4. Select tgw_part and click null .

Part TGV 1 x

File Edit Views Actions Reports Tools Help

Name

Part TGV 1

Query Definition

Part Query

Context Item Type

Part

Description

Max Visible Children On Expand

100

Linked Toolbar/Context Menu

tgw_part

Max Grow Levels

2

5. Save the Tree Grid View.



6. Load an Item configured for TGV to see the new toolbar button. For example, Figure below shows a **Part** Item.

The screenshot displays a software interface with a top toolbar containing buttons for Edit, Refresh, Undo, Redo, and a dropdown menu. Below the toolbar is a 'Part' configuration form with the following fields:

- Part Number:** car
- Revision:** A
- State:** Preliminary
- Assigned Creator:**
- Name:** car
- Designated User:**
- Type:** (dropdown menu)
- Unit:** EA
- Make / Buy:** Make
- Cost:**
- Effective Date:** (calendar icon)
- Long Description:** (text area)
- ☐ Changes Pending

Below the form is a navigation bar with tabs: BOM, BOM Structure, Alternates, AML, Documents, CAD Documents, Goals, Changes, Part Submission Warrants, and **tgv_Part TGV**. The 'tgv_Part TGV' tab is active, showing a table with the following data:

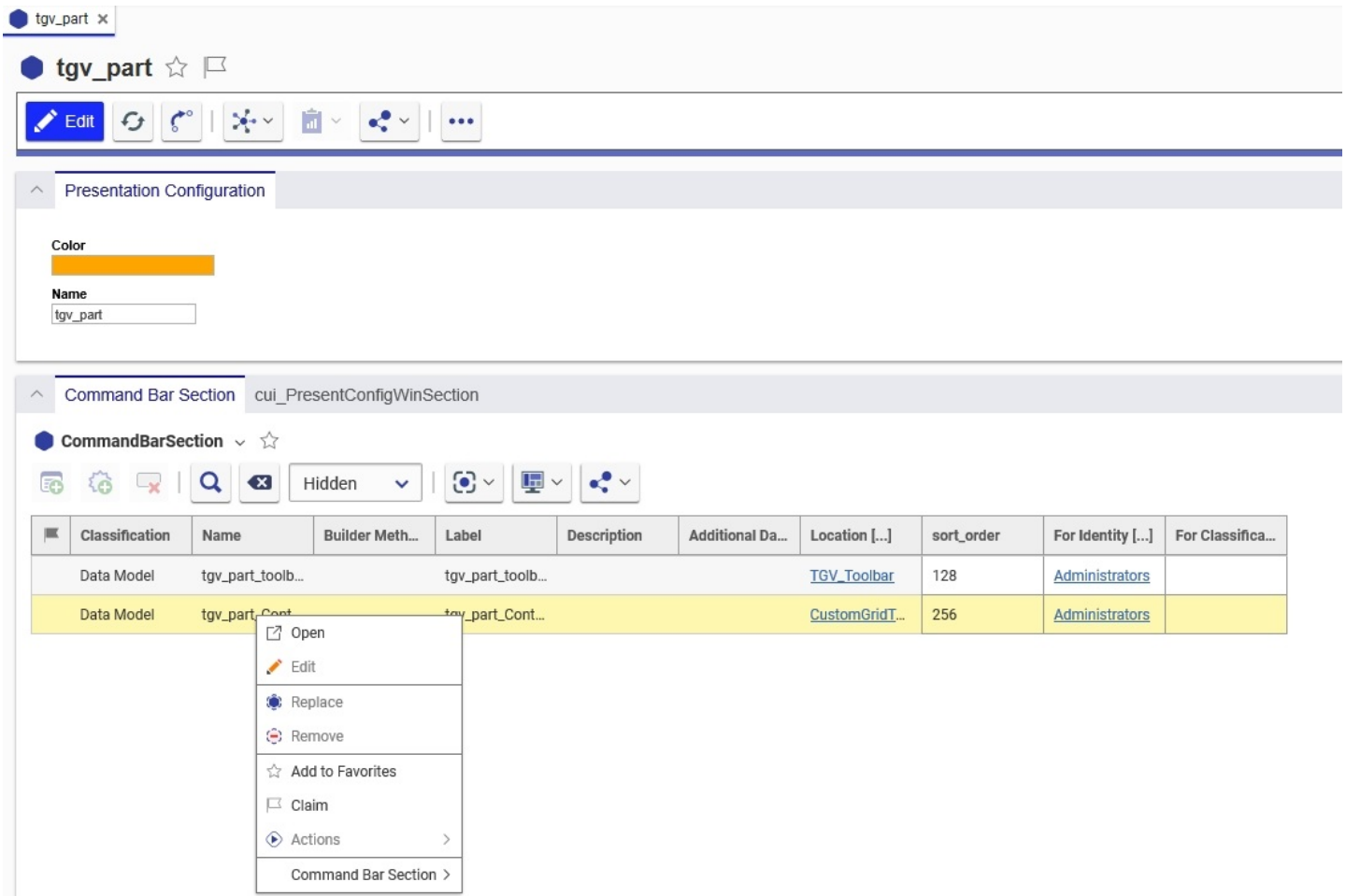
	Part TGV	Name	State	Created On
PC_001	Computer System	Preliminary		
Doc_01	User Guide	Preliminary		
MCO-100002		New	3/23/2017	
admin	Innovator Admin			
PC_001-A - Qty: 1	Computer Tower	Preliminary		

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** in the context menu.





The **tgvp_part_context_menu** Item appears.

tgvs_part_ContextMenu x

tgvs_part_ContextMenu ☆

Edit ↻ ↺ | 🔗 | 📊 | 🔗 | ⋮

^ Command Bar Section

Name
tgvs_part_ContextMenu

Location
CustomGridToolbar

Classification
Data Model

Label
tgvs_part_ContextMenu

Description

^ Command Bar Item Command Bar Section Dependency

CommandBarItem ☆

📄 ⚙️ ✖️ | 🔍 🔍 Hidden 🔍 | 🔍 🔍 🔍

Item Type	Name	Additional Da...	Init Method [...]	Sort Order	Action	Alternate [...]	For Identity [...]	For Classifica...

2. Click null to edit the **tgvs_part_context_menu** Item.
3. Click **Create New Presentation Configuration**. The **Presentation Configuration** Item appears.
4. Enter **tgvs_part** in the **Name** field and select the orange color.

tgvs_part

Edit ↻ ↺ | 🔗 | 📊 | 🔗 | ⋮

^ Presentation Configuration

Color
[Orange]

Name
tgvs_part

- Click the **New Command Bar** null button in the **Command Bar Section** to add a new related Item.
- Enter **tg^v*** in the **Name** cell of the search grid and add both **tg^v_part_toolbar** and **tg^v_part_ContextMenu** Items.

Select Items

CommandBarSection ▾

🔍 🗨 Simple ▾

Classification	Name	Builder Method [...]	Label	Description	Additional D...	Location [...]
Data Model	tg ^v *					
Data Model	tg ^v _part_ContextMenu		tg ^v _part_Co...			CustomGridT...
Data Model	tg ^v _part_toolbar		tg ^v _part_tool...			TGV Toolbar

- Make sure to add the Items to the Relationships grid.

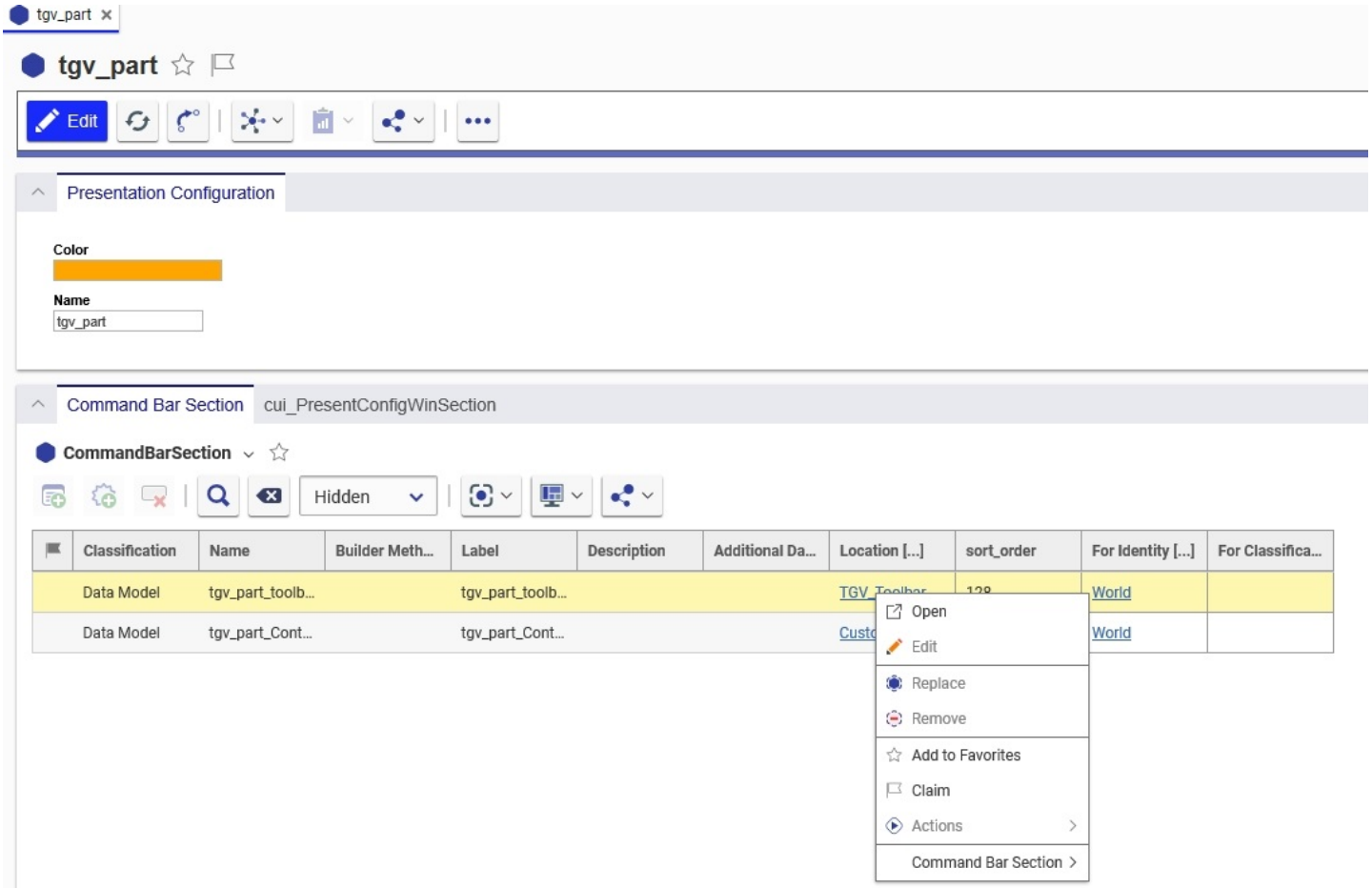
^ Command Bar Section cui_PresentConfigWinSection

CommandBarSection ▾ ☆

🔍 🗨 Hidden ▾ 🔄 🖨 🌐 ▾

Classification	Name	Builder Meth...	Label	Description	Additional Da...	Location [...]	sort_order	For Identity [...]	For Classifica...
Data Model	tg ^v _part_toolb...		tg ^v _part_toolb...			TGV Toolbar	128	World	
Data Model	tg ^v _part_Cont...		tg ^v _part_Cont...			CustomGridT...	256	World	

- Add an identity in the **For Identity** column. This Identity determines which users will have access to the buttons/menus.
- Click null to save the **tg^v_part** Presentation Configuration Item.
- Right-click **tg^v_part_toolbar** and select **Open** from the context menu.



The **tgview_part_toolbar** Presentation Configuration Item appears. It contains the **tgview_view** button.

[illegible]

Location

Classification

Label

tgv_part_toolbar

Description

[illegible]

CommandBarItem ☆










	ItemType	Name	Additional Da...	Init Method [...]	Sort Order	Action	Alternate [...]	For Identity [...]	For Classifica...
---	----------	------	------------------	-------------------	------------	--------	-----------------	--------------------	-------------------

- Click the **New Command Bar** null button in the **Command Bar Item** tab. The **Select Item Type** dialog appears.

Select Item Type



Select the Type of Item that is to be created:



- ☐ Button
- ☐ Separator
- ☐ Menu
- ☐ Menu Button
- ☐ Menu Checkbox
- ☐ DropDown
- ☐ Checkbox
- ☐ Shortcut
- ☐ Edit
- ☐ Menu Separator

OK

Cancel

12. Select **Menu** and click **OK**. It appears in the **Command Bar Item** tab.

13. Right-click the **Menu** ItemType and click **Open** in the context menu.

14. Right-click **tg_v_part_menu** in the **Command Bar Item** grid and select **View Command Bar Item**. The **tg_v_part_menu** Presentation Item appears.



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

Expanding and 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** null button 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.

Expanding Tree Grid Rows Using the Grow Icon

Use the following procedure:

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



PC-001 x

PC-001 ☆

Edit ↻ ↺ ⚙ ⚙ ⚙ ⋮

Part

Part Number

PC-001

Revision

A

State

Preliminary

Assigned Creator

Innovator Admin

Name

Computer System

Designated User

Innovator Admin

Type

Assembly

Unit

EA

Make / Buy

Make

Cost

Effective Date

Long Description

An assembly of a single computer system.

☐ Changes Pending

My BOM BOM BOM Structure Alternates AML Documents CAD Documents Goals Changes tgv_Part TGV

Part TGV

2

	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 - Obj_1	Computer Tower	Preliminary	
CAD_001	Mechanical Draw...	Preliminary	
DOC_002	Computer Specifi...	Preliminary	
MCO-100003		New	3/23/2017
admin	Innovator Admin		

Important

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

Important

Grow can be executed on multiple rows. [See Figure 86](#)

- 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 entering a different value in the **Grow depth** field or by editing the related Tree Grid View definition.

Part TGV

Part TGV

Edit

Tree Grid View

Name

Part TGV

Query Definition

[Part Query](#)

Context Item Type

[Part](#)

Description

Max Visible Children On Expand

100

Linked Toolbar/Context Menu

[tgv_part](#)

Max Grow Levels

2

Auto Grow On Refresh

☐

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:



PC-001

PC-001

Edit

Part

Part Number

PC-001

Revision

A

State

Preliminary

Assigned Creator

[Innovator Admin](#)

Name

Computer System

Designated User

[Innovator Admin](#)

Type

Assembly

...

Unit

EA

▼

Make / Buy

Make

▼

Cost

Effective Date

Long Description

☐ Changes Pending

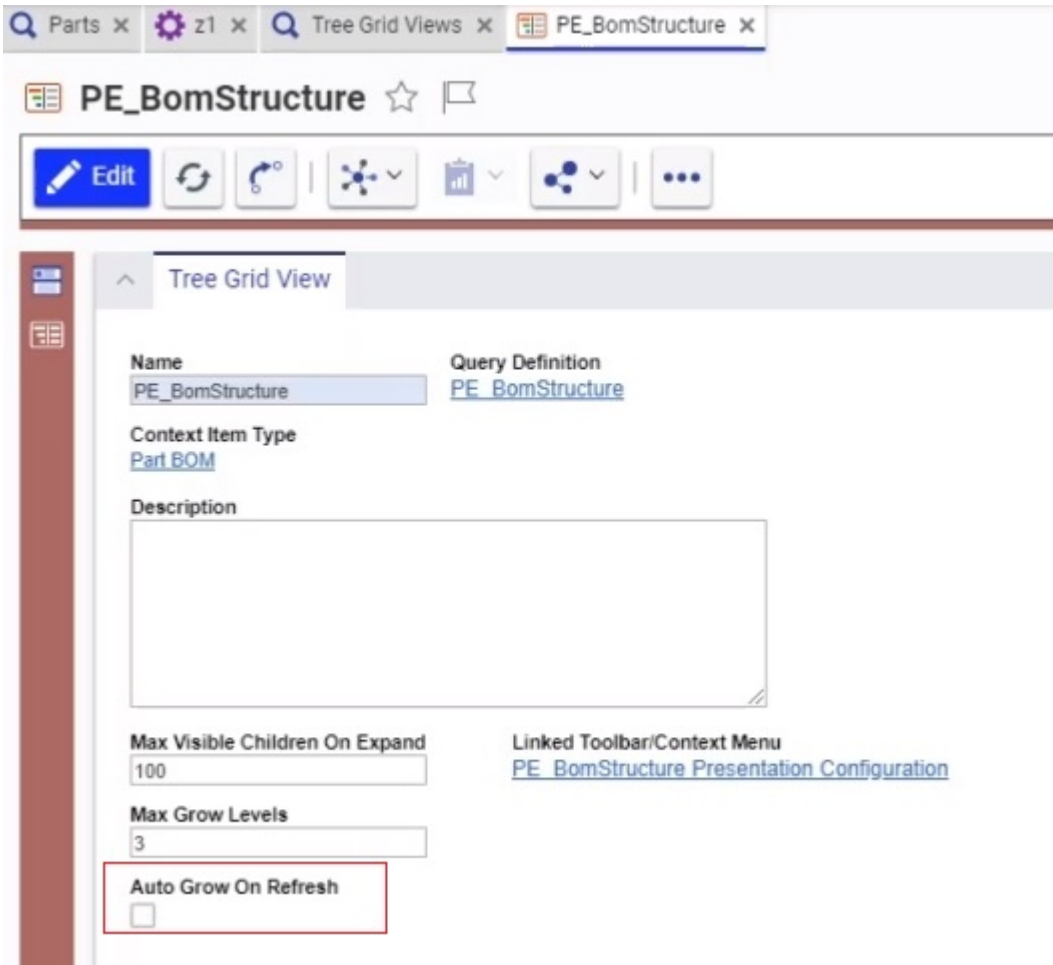
BOM
BOM Structure
Alternates
AML
Documents
CAD Documents
Goals
Changes
tgw_Part TGV

2

Using Auto Grow on Refresh

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





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 null .

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: null , null , null , null .
- Click the null button and then click Create New Revision in the drop-down menu.

Clicking the **Display Settings** null or **Parameter Mapping** null buttons also initiates a refresh.

BOM				
BOM Structure				
Alternates AML Documents CAD Document				
3				
Part Number	R...	State	Seque...	Quantity
z1.1	A	Preliminary	1	1
z1.1.1	A	Preliminary	1	1
z1.1.1.1	A	Preliminary	1	1
z1.1.1.1.1	A	Preliminary	1	1
z1.1.1.1.2	A	Preliminary	2	1
z1.1.1.2	A	Preliminary	2	1
z1.1.2	A	Preliminary	2	1
z1.2	A	Preliminary	2	1

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

Collapsing Tree Grid Rows

Use the following procedure:

1. Select a row in the Tree grid view.

Tree Column		
ID	Name	
989631704C054FD3B0F6ECD35601C529	RootPart	
5879CA57E2474E24A8CD5A8169207DCC	RP_1	
91DE5AE843574422852C35122CF57883	RP_1	
30C13C1E5F5948CFB1638E9A5743749F	RP_2	
4AC9A00C05AF4C1F96CA859821B040E6	RP_3	
F31FA926FE174A4CB9835D4A85B0979E	RP_2	
3AC05CC1B74E4AC7A3C6A44D2D83A0A4	RP_2	

2. Click the **Trim** null button to collapse the rows.



RB_Part_Related RB_CAD_Related TestRB_ComplexJoin TestRB_SimpleStructure TestRB_startPageHtmlWithIframe TestRB_iframeInHtmlFieldOfForm		
10		
Tree Column	ID	Name
RootPart	989631704C054FD3B0F6ECD35601C529	RootPart
RP_1	5879CA57E2474E24A8CD5A8169207DCC	
RP_2	F31FA926FE174A4CB9835D4A85B0979E	
RP_3	D78B97DE3A25480584DF3BD9E5153DBF	
RP_4	88F87FDD594543CD8653CE4F6FF129D2	
TP_1	14D2987C1DF8438BBA036F905E3C8A9F	
TP_2	10F3A751854A4A94994345DEB69DF581	
TP_3	A1E5A9D56049421D937582ED698BA457	

Important

You can also click **Trim** in the context menu by right-clicking the row that you want to collapse. Multiselectd rows can also be trimmed. [MANUAL ACTION REQUIRED: unresolved cross-reference]

Important

Multiselectd rows can also be trimmed. [See Figure 88](#)

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. Go to **Contents --> Design --> Parts**. In this example, we use **RootPart**.



RootPart x

RootPart ☆

Edit

Part

Part Number

RootPart

Revision

A

State

Preliminary

Assigned Creator

Name

RootPart

Designated User

Type

Assembly

Unit

EA

Make / Buy

Make

Cost

Effective Date

Long Description

☐ Changes Pending

RB_Part_Related
RB_CAD_Related
TestRB_ComplexJoin
TestRB_SimpleStructure
TestRB_startPageHtmlWithIframe
TestRB_iframeInHtmlFieldOfForm
tgv_TestR

Tree Column

ID

Name

RootPart	989631704C054FD3B0F6ECD35601C529	RootPart
RP_1	5879CA57E2474E24A8CD5A8169207DCC	
RP_1	91DE5AE843574422B52C35122CF57883	RP_1
RP_2	30C13C1E5F5948CFB1638E9A5743749F	

- Click the **Display Settings** button. The **Display Settings** dialog appears with the Tree Grid view configuration.

Display Settings

×

Max Visible Children on Expand

Top RB Part

☐

RB Parts

☐

Top RB Part

Apply

Cancel

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** null button 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.



Display Settings

×

Max Visible Children on Expand

			Top RB Part	
			RB Parts	
			Top RB Part	

Apply

Cancel

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



RootPart x

RootPart

Edit | Refresh | Undo | Redo | Save | Print | More

Part

Part Number: RootPart Revision: A State: Preliminary Assigned Creator:

Name: RootPart Designated User:

Type: Assembly Unit: EA Make / Buy: Make Cost: Effective Date:

Long Description:

☐ Changes Pending

My BOM BOM BOM Structure Alternates AML Documents CAD Documents Goals Changes tgv_Part TGV

Part TGV Name State Created On

RootPart	RootPart	Preliminary	
----------	----------	-------------	--

6. Click **Grow**. The grid does not show the related Items; it only shows the **Part BOM** Relationships.

RootPart x

RootPart ☆

Edit

Part

Part Number

RootPart

Revision

A

State

Preliminary

Assigned Creator

Name

RootPart

Designated User

Type

Assembly

Unit

EA

Make / Buy

Make

Cost

Effective Date

Long Description

☐ Changes Pending

RB_Part_Related
RB_CAD_Related
TestRB_ComplexJoin
TestRB_SimpleStructure
TestRB_startPageHtmlWithIframe
TestRB_iframeInHtmlFieldOfForm
tgv_TestR

Tree Column

ID

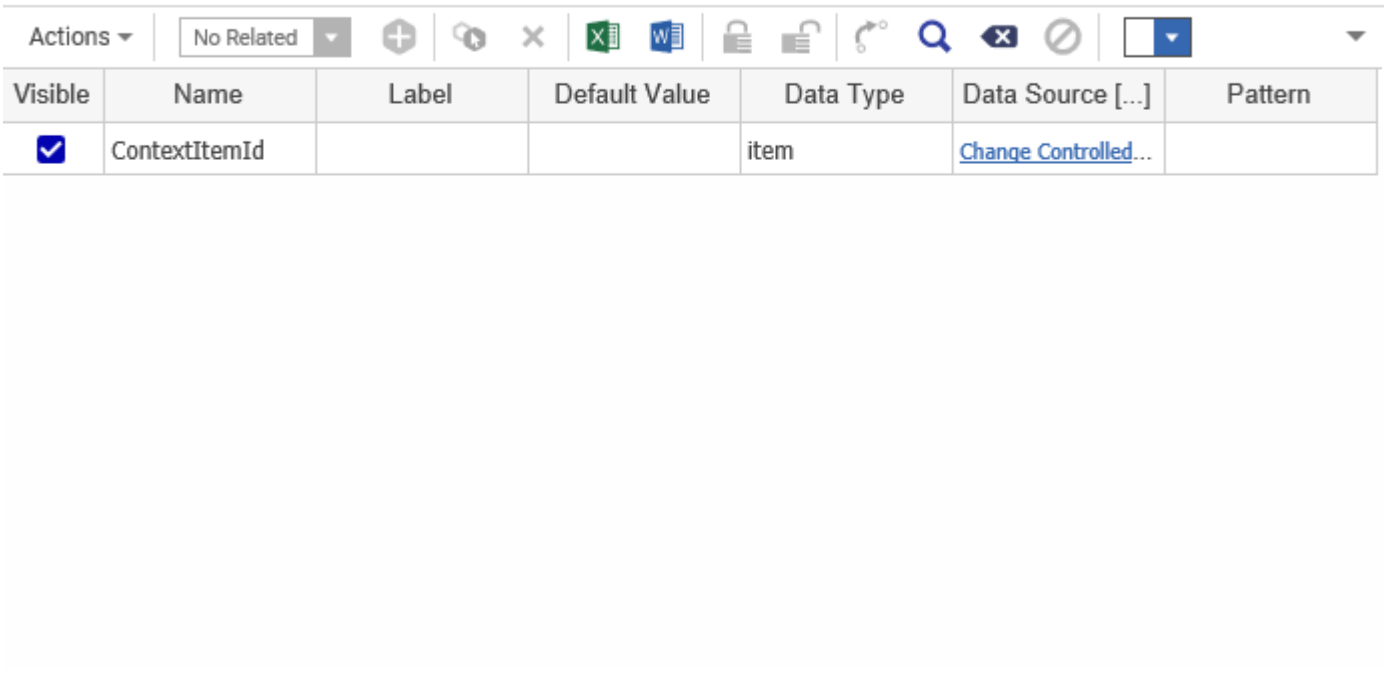
Name

RootPart	989631704C054FD3B0F6ECD35601C529	RootPart
RP_1	5879CA57E2474E24A8CD5A8169207DCC	
RP_1	91DE5AE843574422B52C35122CF57883	RP_1
RP_2	30C13C1E5F5948CFB1638E9A5743749F	

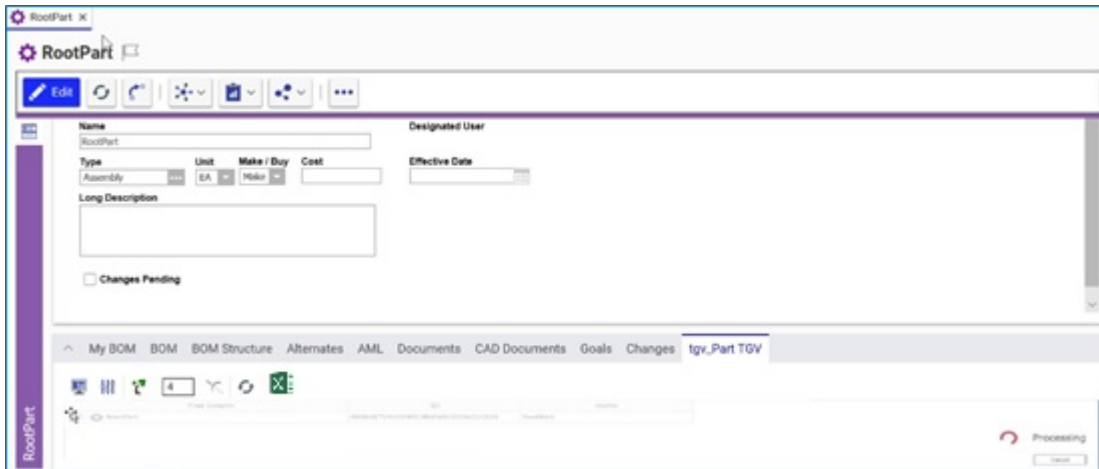
Show Parameter Mapping

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

□ ×



Cancelling a Long Running Query



More About Cell View Types

Support for the following cell view data types has been added in Aras Innovator 38:

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

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:

PE_CAD_Revers... x

PE_CAD_ReverselItemsCAD ☆

Edit

Tree Grid View

Name

PE_CAD_ReverselItemsCAD

Query Definition

PE_CAD_ReverselItemsCAD

Context Item Type

PE_ReverselItemsFedCAD

Description

Max Visible Children On Expand

100

Linked Toolbar/Context Menu

PE_ReverselItems

Max Grow Levels

2

Auto Grow On Refresh

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



PE_CAD_Revers... x

PE_CAD_ReversItemsCAD ☆

Edit ↺ ↻ ⚙️ 📄 🔍 ⋮

	Document Number	Revisi...	Name	Type	State	Claimed By [...]	Native File [...]	Viewable File [...]	Authoring Tool	Changes	Standard	Template	Gen
	{.} Root	{Root.ite...	{Root....	{Root.name}	{Root.classification}	{Root.state}	{Root.locked_by...	{Root.native_file/@...	{Root.viewable_file/...	{Root.authoring_tool}	{Root.ha...	{Root.is...	{Root.is... {Roo...

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

PE_CAD_Revers... x

PE_CAD_ReversItemsCAD ☆

Edit ↺ ↻ ⚙️ 📄 🔍 ⋮

	Document Number	Revisi...	Name	Type	State	Claimed By [...]	Native File [...]	Viewable File [...]	Authoring Tool	Changes	Standard	Template	Gen
	{.} Root	{Root.ite...	{Root....	{Root.name}	{Root.classification}	{Root.state}	{Root.locked_by...	{Root.native_file/@...	{Root.viewable_file/...	{Root.authoring_tool}	{Root.ha...	{Root.is...	{Root.is... {Roo...

Cell View Type: List

List Name: Authoring Tools

List Value Template: {Root.authoring_tool}

Helper: {Root.authoring_tool}, {Root.classification}, {Root.generation}, {Root.has_change_pending}, {Root.id@keyed_name}, {Root.id}, {Root.is_standard}, {Root.is_template}, {Root.item_number}

Save Cancel

3. Select **List** in the **Cell View Type** drop-down list and enter the name of the list in the **List Name** field. When applying the **List** data type, the **List** Item label will appear in the Tree Grid View.
4. Click **Save**. Figure below shows the result when displaying a List Property.



cad1.1

^ CAD Document

Document Number:
 Revision:
 State:
 Assigned Creator:

Name:
 Designated User:

Type:
 Authoring Tool:
 Version:
 From Template:

Description:
☐ Changes Pending
 ☐ Standard
 ☐ Template
 ☐ Part
 ☐ Native File
 ☐ Viewable File

^ Structure Parents Files Changes

Document Number	Re...	Name	Type	State	Locked By [...]	Native File [...]	Viewable File [...]	Authoring Tool	Chan...	Stan...	Tem...	Gen
cad1	A			Preliminary	Innovator Ad...			Microsoft Word	☐	☐	☐	1

Using the Boolean Cell View Type

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

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

PE_CAD_Revers... X

PE_CAD_ReverseltemsCAD ☆

Document Number	Revisi...	Name	Type	State	Claimed By [...]	Native File [...]	Viewable File [...]	Authoring Tool	Changes	Standard	Template	Gen		
{}	Root	{Root.it...	{Root...	{Root.name}	{Root.classification}	{Root.state}	{Root.locked_by...}	{Root.native_file/@...}	{Root.viewable_file/...}	{Root.authoring_tool}	{Root.ha...}	{Root.is...}	{Root.is...}	{Roo...

2. Right-click an entry in the **Changes** column and select **Boolean** in the **Cell View Type** drop-down list:



locked By [...]	Cost	Changes	generation
locked_by_id/@keyed_name	{Root.cost}	{Root.has_change_pending}	{Root.generation}

Cell View Type

Boolean

Text Template

{Root.has_change_pending}

Helper

{Root.classification}
{Root.cost}
{Root.generation}
{Root.has_change_pending}
{Root.id/@keyed_name}
{Root.id}
{Root.item_number}
{Root.link_status}
{Root.locked_by_id/@keyed_name}

Save

Cancel

3. Click **Save**.

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 TGV Definition refers to the **Cost** property in the associated QD:

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



	State	Locked By [...]	Cost	Changes
ation}	{Root.state}	{Root.locked_by_id/@ke...	{Root.cost}	{Root.has_change_pending}

Cell View Type
Decimal

Precision
19

Scale
4

Text Template
{Root.cost}

Helper
{Root.classification}
{Root.cost}
{Root.generation}
{Root.has_change_pending}
{Root.id/@keyed_name}
{Root.id}
{Root.item_number}
{Root.link_status}
{Root.locked_by_id/@keyed_name}

Save
Cancel

Using the Float Cell View Type

The following example also uses the **PE_CAD_ReverselItemsPart** TGV. This example contains a QD that refers to the **Cost** floating point property.

- 1. Click the **Show Editor** button to view the parameters.

Part Number	Revisi...	Name	Type	State	Claimed By [...]	Cost	Changes	gen...
{,} Root	{Root...	{Root.name}	{Root.classification}	{Root.state}	{Root.locked_by...	{Root.cost}	{Root.ha...	{Roo...

- 2. Right-click an entry in the **Cost** column and select **Float** in the **Cell View Type** drop-down list.



	State	Locked By [...]	Cost	Changes
ition}	{Root.state}	{Root.locked_by_id/@ke...	{Root.cost}	{Root.has_change_pending}

Cell View Type

Float

Text Template

{Root.cost}

Helper

{Root.classification}
{Root.cost}
{Root.generation}
{Root.has_change_pending}
{Root.id/@keyed_name}
{Root.id}
{Root.item_number}
{Root.link_status}
{Root.locked_by_id/@keyed_name}

Save

Cancel

Using Date or Time Cell View Types

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

TGV Date Time Formats

Version

Modified Date and Time

Document	{Document.name}	{Document.gen...	{Document.modified_on}
----------	-----------------	------------------	------------------------

Cell View Type

Date

Date Format

Short Date

Short Date

Short DateTime

Long Date

Long Date Time

Text Template

{Document.modified_on}

Helper

(Document.created_by_id/@keyed_name)
(Document.created_by_id)
(Document.generation)
(Document.modified_on)
(Document.name)

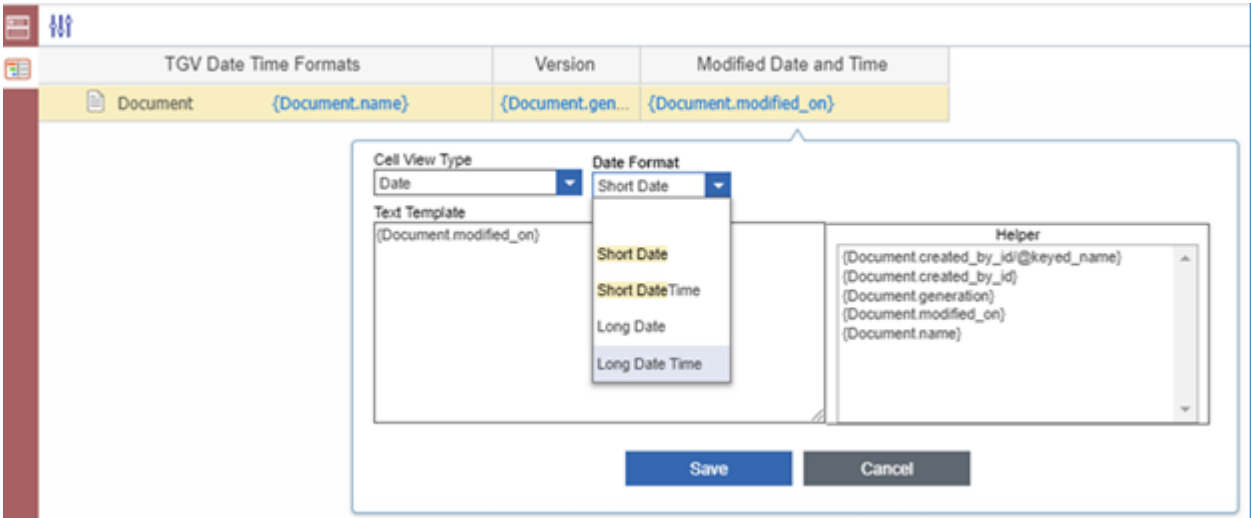
Save

Cancel

In this example, the **created_on** date type property is displayed as the **Long Date Time** Cell View Type. The results appear as shown in Figure below.

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

Selecting **Short Date** does not provide enough information; see the figure above.



In this example, the **created_on** date type property is displayed as the **Long Date Time** Cell View Type. The results appear as shown in the figure below.



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

Using the Integer Cell View Type

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**, a value is sorted numerically (ascending or descending) unlike **Text**, whose resulting values are sorted numerically instead.



TGV Date Time Formats	Version	Creator	Creation Date
Document {Document.name}	{Document.gen...}	{Document.created_by_id}	{Document.crea...}

Cell View Type
Integer

Text Template
{Document.generation}

Helper

- {Document.created_by_id/@keyed_name}
- {Document.created_by_id}
- {Document.created_on}
- {Document.generation}
- {Document.name}

Save Cancel

A common usage for **Integer** is **Version (generation)**.



Tree Grid View Sample



TGV Date Time Formats	Version ↑
.....  Lorem Ipsum	1
.....  Lorem Ipsum	2
.....  Lorem Ipsum	3
.....  Lorem Ipsum	4
.....  Lorem Ipsum	5
.....  Lorem Ipsum	6
.....  Lorem Ipsum	7
.....  Lorem Ipsum	8

Tree Grid View Multi-Select Support

Tree Grid View (TGV) supports multi-row selection by default. This enhancement allows users to select multiple rows and apply Grow and Trim commands across all selected items at the same time.

Multi-Row Selection Behavior

Multi-row selection is enabled automatically for all Tree Grid Views. Users can:

- Use **Ctrl+Click** to add individual rows to the current selection
- Use **Shift+Click** to select a continuous range of rows
- Execute **Grow** and **Trim** commands on all selected rows

No configuration changes are required to enable this functionality.

Disabling Multi-Select



If a solution requires single-row selection behavior, multi-select can be disabled using one of the following methods:

1. URL Parameter

Append the following parameter to the Tree Grid View URL: ?multiSelect=false

2. Programmatic API

Use the Tree Grid View grid API to control the multi-select state:

- options.tgvModule.getMultiSelect()
- options.tgvModule.setMultiSelect(isMultiselectEnabled : Boolean)

Grow Command Behavior with Multi-Select

When one or more rows are selected:

- The system determines the topmost selected parent in each hierarchy chain
- If both a parent and its child are selected, only the parent row is expanded
- Each topmost selected row is expanded based on the configured Grow Depth value
- Expansion operations are executed in parallel for improved performance
- When no rows are selected:
- The Grow command behaves as Expand All and expands all rows in the Tree Grid View

This behavior extends the Grow functionality described in Section "Expanding and Collapsing Tree Grid Rows", while maintaining backward compatibility.

Before expanding:





Part Number	Re...	State	Sequence	Quantity	Claimed By
  Part002	A	Preliminary	1	1	Super User
  Part003	A	Preliminary	2	1	
  Part003_01	A	Preliminary	1	1	
  Part003_02	A	Preliminary	2	1	
  Part003_03	A	Preliminary	3	1	
  Part003_04	A	Preliminary	4	1	
  Part003_05	A	Preliminary	5	1	
  Part004	A	Preliminary	3	1	
  Part005	A	Preliminary	4	1	
  Part001	A	Preliminary	5	1	

After expanding:





Part Number	Re...	State	Sequence	Quantity	Claimed By
Part002	A	Preliminary	1	1	Super User
Part002_01	A	Preliminary	1	1	
Part002_01_01	A	Preliminary	1	1	
Part002_02	A	Preliminary	2	1	
Part002_03	A	Preliminary	3	1	
Part003	A	Preliminary	2	1	
Part003_01	A	Preliminary	1	1	
Part003_01_01	A	Preliminary	1	1	
Part003_01_01_01	A	Preliminary	1	1	
hh	A	Preliminary	1	1	
Part003_02	A	Preliminary	2	1	
Part003_03	A	Preliminary	3	1	
Part003_04	A	Preliminary	4	1	
Part003_05	A	Preliminary	5	1	
Part004	A	Preliminary	3	1	

Trim Level Command Behavior with Multi-Select

When multiple rows are selected:



- The Trim Level command invokes collapseLevel for each selected row
- collapseLevel collapses all expanded items at the same hierarchical level as the selected row
- The behavior for each row is identical to the single-row Trim Level behavior described in Section "Expanding and Collapsing Tree Grid Rows"

Before collapsing:





Part Number	Re...	State	Sequence	Quantity	Claimed By
Part002	A	Preliminary	1	1	Super User
Part002_01	A	Preliminary	1	1	
Part002_01_01	A	Preliminary	1	1	
Part002_02	A	Preliminary	2	1	
Part002_03	A	Preliminary	3	1	
Part003	A	Preliminary	2	1	
Part003_01	A	Preliminary	1	1	
Part003_01_01	A	Preliminary	1	1	
Part003_01_01_01	A	Preliminary	1	1	
hh	A	Preliminary	1	1	
Part003_02	A	Preliminary	2	1	
Part003_03	A	Preliminary	3	1	
003_03_1	A	Preliminary	1	1	
Part003_04	A	Preliminary	4	1	
Part003_05	A	Preliminary	5	1	

After collapsing:



    3   						
Part Number	Re...	State	Sequence	Quantity	Claimed By	
  Part002	A	Preliminary	1	1	Super User	
  Part003	A	Preliminary	2	1		
  Part004	A	Preliminary	3	1		
  Part005	A	Preliminary	4	1		
  Part001	A	Preliminary	5	1		

Updating Custom Logic for Multi-Select Compatibility

Custom client methods that operate on Tree Grid View selections must be updated to support multiple selected rows. **Accessing selected rows:**

- Retrieve the Tree Grid instance using `window.mainPage.getTreeGrid()`
- Access the selected row identifiers via `treeGrid.settings.selectedRows`
- Iterate over the `selectedRows` array to apply logic to each selected row

Disabling Command Bar Actions for Multi-Select

Certain actions may not be applicable when multiple rows are selected. Command bar buttons can be conditionally disabled based on the current selection count. **Update Client method to disable a command bar button when more than one row is selected:**

- Retrieve the Tree Grid instance using `window.mainPage.getTreeGrid()`
- Access the selected rows using `treeGrid.settings.selectedRows`
- Check whether `selectedRows.length` is greater than one
- Return `{ disabled: true }` from the button state handler