

Creating Tree Grid Views

The Tree Grids are defined by the Tree Grid View Items that are found as **Contents --> Administration --> Configuration --> Tree Grid Views**.



Tree Grid Views

 Search

 Clear

Simple 



	Name	Query Definiti...	Max Query D...	Max Child It...	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	

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.



TGV x

TGV

Save
 Done
 Discard

Tree Grid View

Name

Query Definition
[Part Query](#)

Context Item Type
[Part](#)

Description

Max Visible Children On Expand

Linked Toolbar/Context Menu

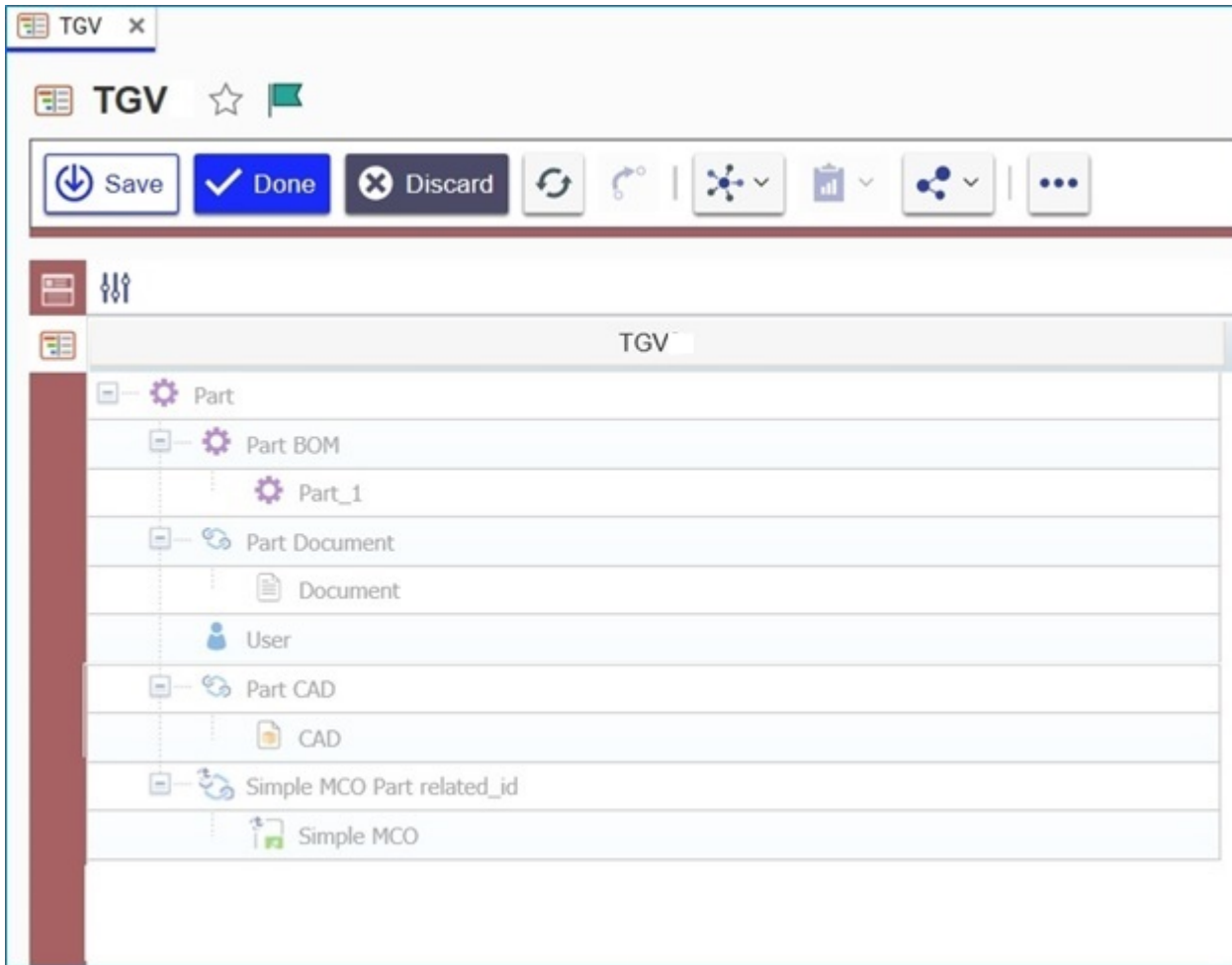
Max Grow Levels

Auto Grow On Refresh
☐

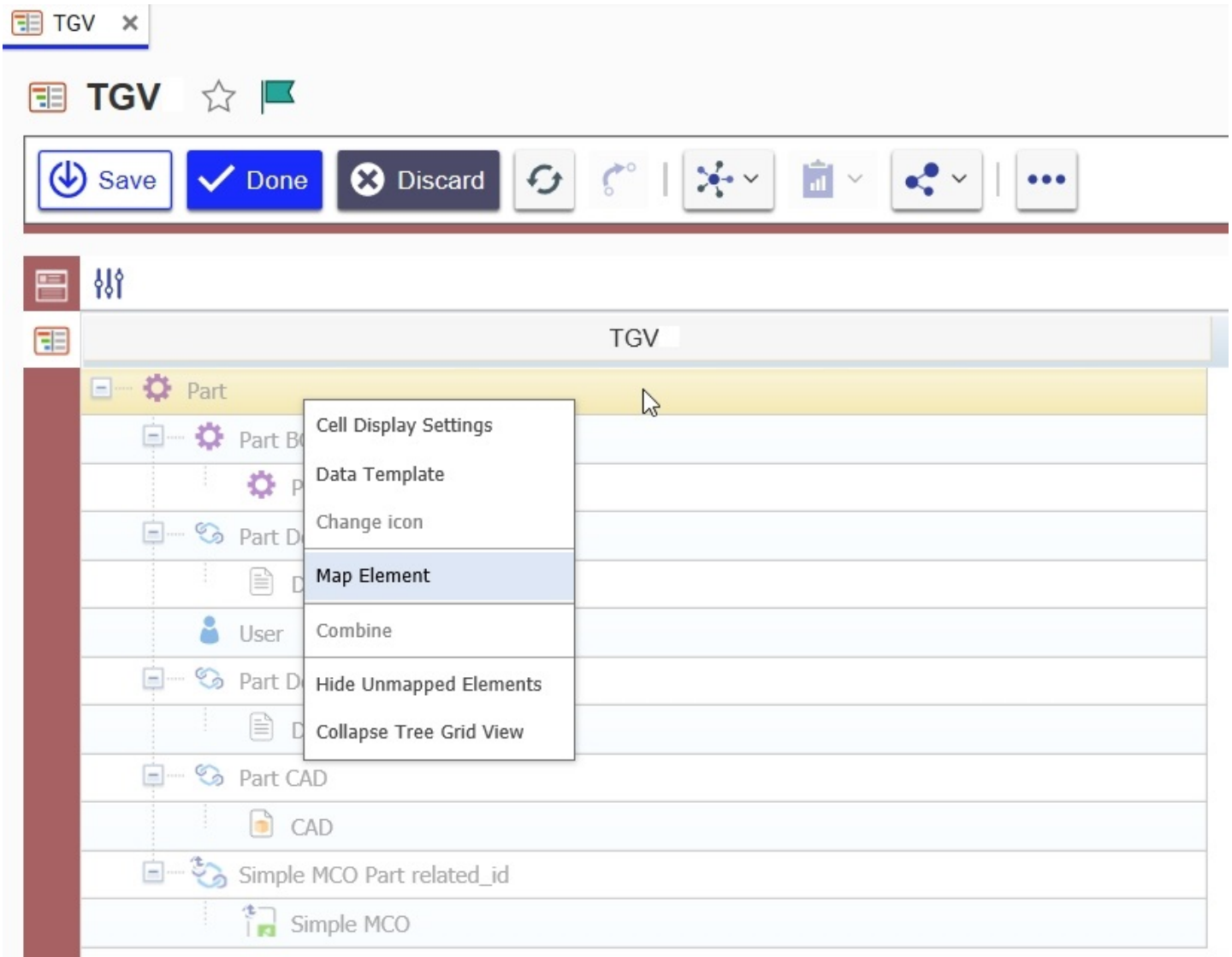
Important

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.



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



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

TGV x

TGV ☆

Edit ↺ ↻ ↺ ↻ ↺ ↻

TGV2

Part	
Part BOM	
Part_1	
Part Document	
Document	
User	
Part CAD	
CAD	
Simple MCO Part related_id	
Simple MCO	

- Add New Column
- Change Column Label
- Remove Column
- Show Unmapped Elements
- Collapse Tree Grid View

6. Right-click the new column, click **Change ColumnLabel**, and then call it **Name**.



TGV x

TGV

☆

🚩

Save

Done

Discard

↺

↻

⚙️

📊

🔗

⋮

⚙️

⚙️

TGV	Name
Part	
Part BOM	
Part_1	
Part Document	
Document	
User	
Part CAD	
CAD	
Simple MCO Part related_id	
Simple MCO	

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

©2026 Aras Corporation All Copyrights Reserved

8

TGV x

TGV ☆

Save

Done

Discard

Part

Part BOM

Part_1

Part Document

Document

User

Part CAD

CAD

Simple MCO Part related_id

Simple MCO

TGV2	Name	State	Created On

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 8 types of data.



The UI-supported data types



Data Type	Description
-----------	-------------

	Displays value
--	----------------

Text	plain text
-------------	------------

	Parses the decimal number to display
--	--------------------------------------

Decimal	delimiter as either " " or " "
----------------	--------------------------------

	Parses the data to display
--	----------------------------

Date	as a date. Supports short/long/date/time
-------------	--

	Displays color
--	----------------

Color	Displays a hyperlink
--------------	----------------------

Item	the specified Item
-------------	--------------------

	Displays List property value
--	------------------------------

	Displays Floating
--	-------------------

Float	Property value
--------------	----------------

	Displays Boolean property value
--	---------------------------------

Boolean	Displays an Integer
----------------	---------------------



Property
value

Important

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:
 1. **Cell View Type:** Text
 2. **Text Template:** {Part.name}

The screenshot shows the TGV2 interface. On the left is a tree view with the following structure:

- Part
 - Part BOM
 - Part_1
 - Part Document
 - Document
 - User
 - Part CAD
 - CAD
 - Simple MCO Part related_id
 - Simple MCO

On the right is a table with columns: Name, State, Created On. The first row under 'Part' has the value '{Part.name}' in the 'Name' column. A dialog box titled 'Cell View Type' is open, showing 'Text' as the selected view type and '{Part.name}' as the text template. A 'Helper' list on the right of the dialog shows the following properties: {Part.id/@keyed_name}, {Part.id}, {Part.item_number}, {Part.managed_by_id/@keyed_name}, {Part.managed_by_id}, {Part.name}, and {Part.state}. The dialog has 'Save' and 'Cancel' buttons.

Important

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

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



Part TGV x

Part TGV ☆

Edit ↺ ↻ ⚙️ 📄 🔗 ⋮

Part TGV		Name	State	Created On
Part		{Part.name}		
Part BOM				
Part		{Part.name}		
Part Document				
Document				
User				
Part Document_1				
Document_1				
Part CAD				
CAD		{CAD.name}		
Simple MCO Part related_id				
Simple MCO				

- Double-click the **Part Created ByName** 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 BOM				
Part		{Part.name}		
Part Document				
Document		{Document.name...}		
Part CAD				
CAD		{CAD.name}		
Part Created By				
Simple MCO Part related_id				
Simple MCO				

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}

Save Cancel



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

TGV x

TGV ☆

Save Done Discard

TGV		Name	State	Created On
Part		{Part.name}	{Part.state}	
Part BOM				
Part		{Part.name}	{Part.state}	
Part Document				
Document		{Document.name}	{Document.state}	
Part Created By		{Part Created B...}		
Part Document_1				
Document_1				
Part CAD				
CAD		{CAD.name}		
Simple MCO Part related_id				
Simple MCO				

5. Double-click on the **Simple MCOCreated On** cell and set the following properties:

1. Cell View Type = Date
2. Text Template = {Simple MCO.created_on}

TGV ☆

Save Done Discard

Part TGV		Name	State	Created On
Part {Part.item_number}		{Part.name}	{Part.state}	
Part BOM				
Part {Part.item_number}		{Part.name}	{Part.state}	
Part CAD				
CAD		{CAD.name}	{CAD.item_num...}	
Part Document				
Document		{Document.name}	{Document.state}	
Part Created By		{Part Created B...}		
Simple MCO Part related_id				
Simple MCO			{Simple MCO.st...}	{Simple MCO.cr...}

6. Double-click the top **PartPart TGV** cell and set the following properties:

1. Cell View Type: Item
2. Innovator ItemType Name: Part
3. Id Template: {CAD.id}
4. Text Template: {CAD.item_number}

	Name	State	Created On
Part {Part.item_number}	{Part.name}	{Part.state}	
Part BOM			
Part {Part.item_number}	{Part.name}	{Part.state}	
Part CAD			
CAD	{CAD.name}	{CAD.item_num...}	
Part Document			
Document	{Document.nam...}	{Document.state}	
Part Created By	{Part Created B...}		
Simple MCO Part related_id			
Simple MCO		{Simple MCO.st...}	{Simple MCO.cr...}

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

	Name	State	Created On
Part {Part.item_number}	{Part.name}	{Part.state}	
Part BOM			
Part {Part.item_number}	{Part.name}	{Part.state}	
Part CAD			
CAD	{CAD.name}	{CAD.item_num...}	
Part Document			
Document	{Document.nam...}	{Document.state}	
Part Created By	{Part Created B...}		
Simple MCO Part related_id			
Simple MCO		{Simple MCO.st...}	{Simple MCO.cr...}

Important

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 ByPart 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 BOM				
Part_1				
Part CAD				
CAD				
Part Document				
Document		{Document.nam...}	{Document.state}	
Part Created By	{Part Created By.login_name}	{Part Created B...		
Simple MCO Part related_id				
Simple MCO			{Simple MCO.st...}	{Simple MCO.cr...}

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

Part TGV		Name	State	Created On
Part	{Part.item_number}	{Part.name}	{Part.state}	
Part BOM				
Part_1				
Part CAD				
CAD				
Part Document				
Document		{Document.nam...}	{Document.state}	
Part Created By	{Part Created By.login_name}	{Part Created B...		
Simple MCO Part related_id				
Simple MCO			{Simple MCO.st...}	{Simple MCO.cr...}

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- MANUAL ACTION REQUIRED

