

STUDENT TRAINING GUIDE

ACE 2019

INNOVATE
TRANSCEND
REALIZE

Query Builder and Tree Grid View



Copyright © 2019 by Aras Corporation. This material may be distributed only subject to the terms and conditions set forth in the Open Publication License, V1.0 or later (the latest version is presently available at <http://www.opencontent.org/openpub/>).

Distribution of substantively modified versions of this document is prohibited without the explicit permission of the copyright holder.

Distribution of the work or derivative of the work in any standard (paper) book form for a commercial purpose is prohibited unless prior permission is obtained from the copyright holder.

Aras Innovator, Aras, and the Aras Corp "A" logo are registered trademarks of Aras Corporation in the United States and other countries.

All other trademarks referenced herein are the property of their respective owners.

Microsoft, Office, SQL Server, IIS and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Notice of Liability

The information contained in this document is distributed on an "As Is" basis, without warranty of any kind, express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose or a warranty of non-infringement. Aras shall have no liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this document or by the software or hardware products described herein.

STUDENT TRAINING GUIDE

ACE 2019

INNOVATE
TRANSCEND
REALIZE

Query Builder and Tree Grid View



Copyright © 2019 by Aras Corporation. This material may be distributed only subject to the terms and conditions set forth in the Open Publication License, V1.0 or later (the latest version is presently available at <http://www.opencontent.org/openpub/>).

Distribution of substantively modified versions of this document is prohibited without the explicit permission of the copyright holder.

Distribution of the work or derivative of the work in any standard (paper) book form for a commercial purpose is prohibited unless prior permission is obtained from the copyright holder.

Aras Innovator, Aras, and the Aras Corp "A" logo are registered trademarks of Aras Corporation in the United States and other countries.

All other trademarks referenced herein are the property of their respective owners.

Microsoft, Office, SQL Server, IIS and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Notice of Liability

The information contained in this document is distributed on an "As Is" basis, without warranty of any kind, express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose or a warranty of non-infringement. Aras shall have no liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this document or by the software or hardware products described herein.



Query Builder and Tree Grid View

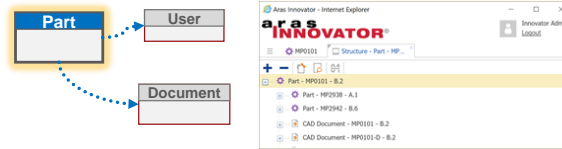
Overview: In this session, you will learn how to present item data to a user in an easy to navigate grid control. Using a graphical interface, you define and save complex reusable query definitions that retrieve property data from the Aras Innovator database. Query results are then mapped to a configurable Tree Grid View that allows a user to visualize information in a hierarchical display. Although a knowledge of AML is not required to create query definitions, a basic understanding of the data constructs is helpful.

- Objectives:**
- ✓ Item Referencing Tools
 - ✓ Describing the Tree View Grid
 - ✓ Describing the Query Builder
 - ✓ Creating a new Query Definition
 - ✓ Creating a Tree Grid View
 - ✓ Assign Tree Grid View Usage
 - ✓ Enhancing Tree Grid View Display
 - ✓ Enabling Query Parameters

Item Referencing Tools

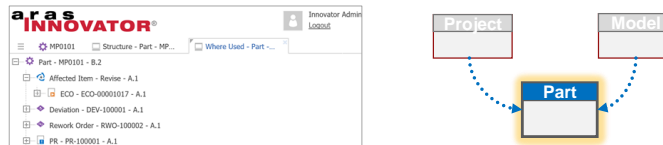
Structure Browser

- Show Items *referenced from* context item



Where Used

- Shows all Items that reference a context Item



Reviewing Existing Data Tools

Aras Innovator contains two pre-built data tools that allow a user to view data using a “top-down” display (Structure Browser) or a “bottom-up” display (Where Used) that shows all items that reference the current item in context.

Neither of these tools allow for any customization and the data displayed is often too limited for some users.



Try It – Structure Browser and Where Used

- Open a Browser window and enter the server URL given to you by the Instructor. The URL will be in the format: http://<servername>/<innovator_web_alias>, e.g. <http://localhost/Innovator11>.
- Login with the credentials Username: admin, Password: innovator and Database: Solutions11Demo.
- In the Table of Contents, navigate to Design > Parts and open the Parts Search Grid.
- Clear previous Search Criteria.
- Locate the Part Number MP2963 with Name Stepper Motor Assembly, and open it for viewing.
- Open the BOM relationship tab. Notice the related “child” parts.
- Open the CAD Documents relationship tab. Notice the related “child” CAD document.
- Click Views > Structure Browser to show Items referenced from the Stepper Motor Assembly (i.e. “child” Items).
- Return to the Part MP2963 and click Views > Where Used to show Items that reference the Stepper Motor Assembly (i.e. Parent Items: an ECO and two revisions of Body part number MP2942).
- Close the Part MP2963 Stepper Motor Assembly, as well as the Structure Browser and Where Used tabs.

User Stories

- *"I want a customized BOM, where I choose the Properties and Filters."*
- *"I want to view all Revisions of a given item in one tab, so I can retrieve any generation."*
- *"I want to view the parts in a given location, so I can manage inventory."*
- *"I want to view all the Projects and Programs where one part is used so I can make proper business decisions."*
- *"I want to view Part Relationships in a recursive tree view so I can see the entire multi-level item structure."*
- *"I want to view the first level part relationships so I can view parts and documents in a single tab."*



4 | © 2018 Aras

User Stories

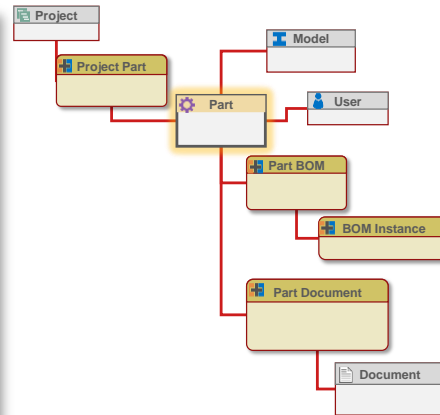
User requests are driving the development of data visualization in Aras Innovator. The Query Builder tool provides a way for users to create their own customized queries to retrieve data exactly how they need it to make business decisions. The Tree Grid View displays the customized queries in an expandable grid-style display.

Here we see some example user stories that inspired the Query Builder and Tree Grid tools.

Describing Tree Grid View

- Shows Items that reference and that are referenced by a context Item

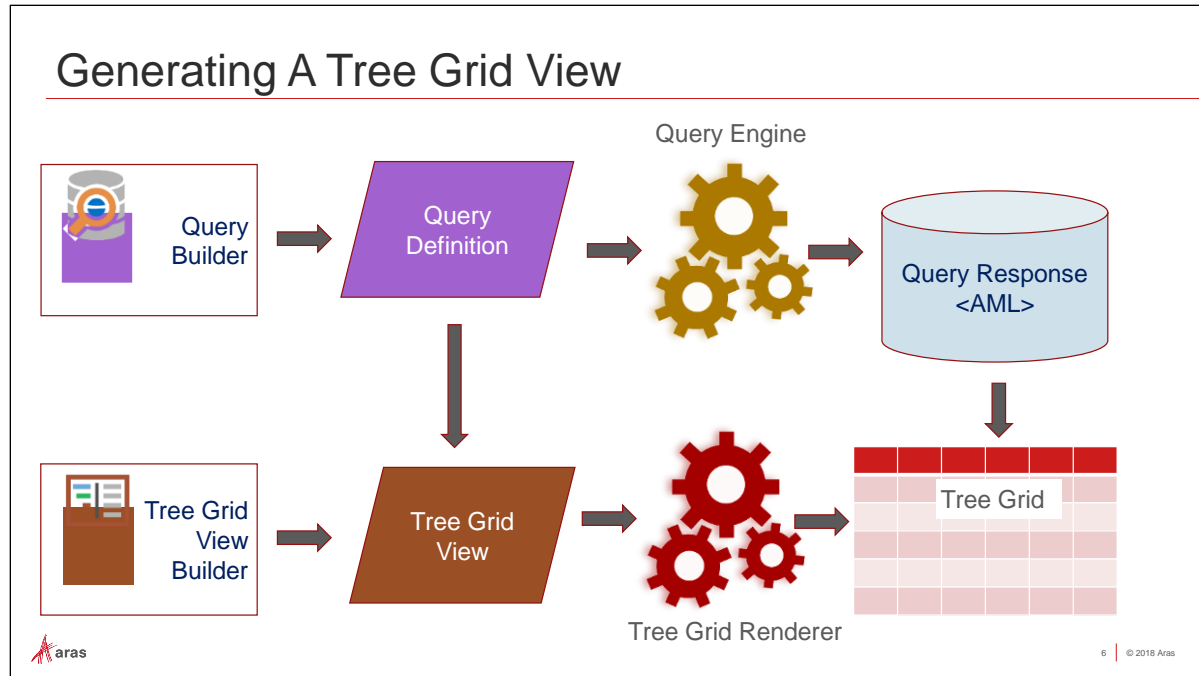
| | Name | Gen | Qty | Type |
|---------|-----------------------|-----|-----|-----------|
| P-123 | MakerBot Replicator | 1 | | Assembly |
| P-001 | Motor | 1 | 2 | Component |
| P-0234 | Gantry | 1 | 1 | Assembly |
| P-212 | Electronics | 2 | 2 | Assembly |
| P-2345 | Wiring | 4 | 1 | Assembly |
| Manuals | | | | |
| Doc-15 | Installation Guide | 1 | | Manual |
| Doc-23 | Catalog Description | 3 | | Manual |
| Doc-51 | Assembly Instructions | 3 | | Manual |



Describing Tree Grid View

The Tree Grid View control is a configurable user interface component that shows data using a hierarchical view. Users can expand and collapse data leaves to show many kinds of items that are typically related to or from a starting context item.

A Query Builder Definition is first created to retrieve the desired items and related data is then mapped to a Tree View Grid to allow end users to easily navigate through the returned results.



Generating a Tree Grid View

The diagram above outlines the basic steps involved in displaying data in a Tree Grid View.

First, an administrator/developer defines a Query Definition using the Query Builder editing tool that allows for the selection of items, relationships and property data from the Aras Innovator database. The Query Definition is executed through a Query Engine, which returns the resulting AML data from the Innovator server.

Separately, an administrator/developer defines a Tree Grid View definition using the Tree Grid View builder. This describes how to display the data in a series of hierarchical rows and columns. Property data retrieved from an associated Query Definition is mapped to columns in the Tree Grid View. The Tree Grid Renderer displays the data from the Query Definition in the format defined by the Tree Grid View.

When a user opens a starting context item, the Query Engine retrieves the item data, which is rendered by the Tree Grid Renderer, and the resulting grid is displayed to the end user.

Describing Query Builder

- Builds re-useable complex data Queries
- Uses a graphical interface
- Intended to work with Tree Grid View
- Features
 - Can specify which properties are returned
 - Query on Related Items
 - Query on Item Properties
 - Query on Parent Items
 - Perform recursive queries



6 | © 2018 Aras

Describing the Query Builder Tool

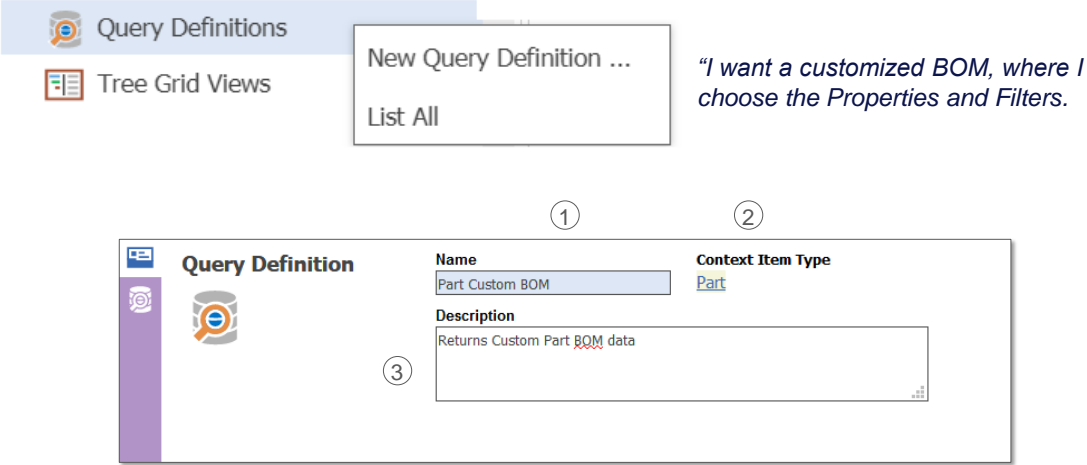
The Query Builder is used to create reusable Query Definitions that are then associated with one or more Tree Grid Views.

The Query Builder uses a graphical interface, which allows an admin/developer to choose which items/relationships/properties will be retrieved from the database.

The Query Builder features the ability to:

- Select property data to be returned
- Build queries that return related items based on a relationship
- Build queries that return item data based on Item datatype properties
- Perform “where used” queries that return referencing items
- Perform recursive relationship queries on nested data

Creating a New Query Definition



The screenshot shows the Aras Query Definitions interface. At the top, there's a 'Query Definitions' menu with a 'New Query Definition ...' option. A callout box next to it says: "I want a customized BOM, where I choose the Properties and Filters." Below this, the 'Query Definition' form is shown with three numbered callouts: 1 points to the 'Name' field (containing 'Part Custom BOM'), 2 points to the 'Context Item Type' field (containing 'Part'), and 3 points to the 'Description' field (containing 'Returns Custom Part BOM data').

Creating New Query Definitions

Query Definitions are created by an admin/developer from the Administration > Configuration > Query Definitions menu on the TOC. Choose New Query Definition... to build a new definition item.

The following fields are presented:

- ① **Name** - Name of Query Definition.
- ② **Context Item Type** - Item Type of the starting context item.
- ③ **Description** - Remarks about the definition.



Try it - Create a New Query Definition

1. From the Table of Contents, select Administration > Configuration > Query Definitions.
2. Click the Create a New Item button (+) to create a New Query Definition.
3. Provide a Name and select an existing Context Item Type. In this example, we will use the Part Item Type to retrieve a custom Bill of Materials for a starting Part.

| Property | Value |
|-------------------|------------------------------|
| Name | Part Custom BOM |
| Context Item Type | Part |
| Description | Returns Custom Part BOM data |

4. Save changes to the definition item, leaving it open in Edit mode.

Selecting Properties

- Item Type Properties
- Properties on Relationship Items
- Properties on Child Items
- Extended Properties (x-Properties)

9 | © 2018 Aras

Selecting Properties

Property to be retrieved are selected for each item type in the editor window. You can add and remove properties at any time during the definition process. In this example, we will define some Part properties to be returned.



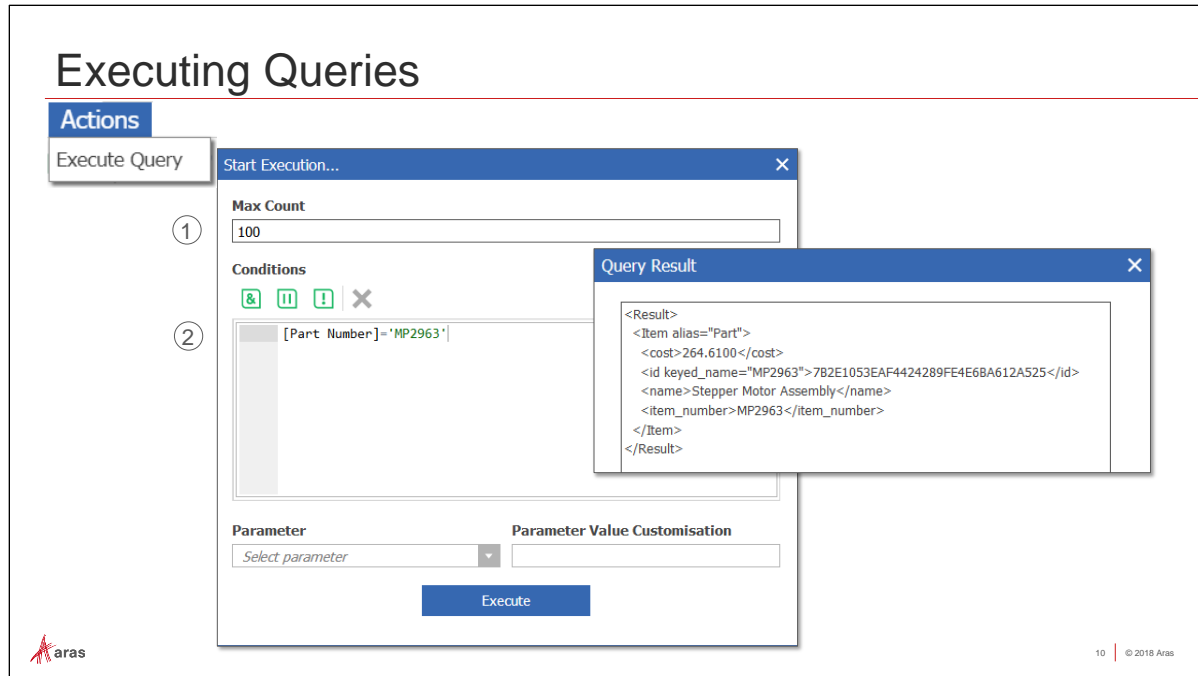
Try It - Select Properties on the Query

1. Click the Show Editor sidebar button to display the graphical representation of the item type.
2. Select the first row, which currently shows the Part ItemType.
3. Click the Properties button (or choose Edit Query Element > Selected Properties... from the context menu) to display the Properties dialog.
4. Add the following properties to the query definition. To remove a property, press the delete row (X) icon.

| Properties |
|-------------|
| Part Number |
| Name |
| Cost |
| id |

Hint: Type in the first letter of a property to locate it quickly

5. Press the Save button to capture the selected properties. You may have to scroll down to see the Save and Cancel buttons.
6. Hover over the Properties button to see your selected properties.



Executing Queries

You can test the query definition at any time by using the Execute Query Action. The AML results of the executed definition will appear in a Query Result window.

You can also specify Conditions and Parameters (discussed in more detail later in this session) to filter the results.

- ① **Max Count** – Maximum records returned. A default Max Count of 100 limits the number of results returned.
- ② **Conditions** – Filters using criteria to limit results.

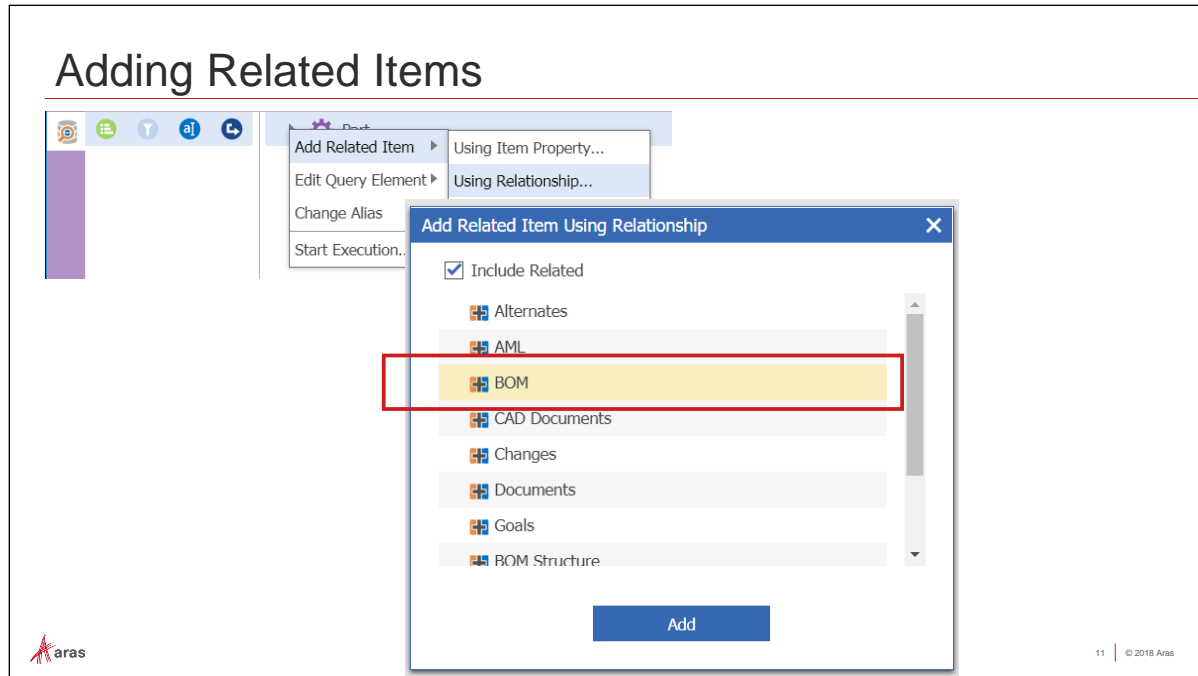


Try it - Execute a Query

1. Make sure to save the query definition and then choose Actions > Execute Query from the tear off menu.
2. To supply a test condition, enter the name of a property in the Conditions editor. The editor auto-fill will prompt for properties defined on the item. In this example, we will set the condition to find the [Part Number] = 'MP2963'.
3. Press Execute to run the query. Results display in AML format, inside the Query Result window.
4. Close the Query Result window before continuing.

Note

The Conditions editor supports basic SQL evaluation including =, >, <>, LIKE, IS NULL, etc. You can also create more complex statements using Boolean operators AND, OR and NOT.



Adding Related Item

To add a relationship data FROM the starting context item, use the Add Related Item > Using Relationship... option. A dialog displays which allows you to select from an existing relationship where the context item type is defined as the source of the relationship.



Try it - Add Related Items

1. Select the row to build a relationship FROM - in this example, select the row containing the Part ItemType.
2. Right mouse click and choose Add Related Item > Using Relationship... from the context menu.
3. Select the BOM relationship to create a custom Bill of Materials. This is the Part BOM Relationship.
4. Click Add to include this relationship in the query definition.
5. Save the Part Custom BOM Query Definition.
6. Once a relationship item is saved into the query definition, we have access to properties on it. Click on the Part BOM row and add the Quantity property to the query definition, by clicking the Properties button and adding Quantity.

Note

You can remove a related item at any time by selecting a row in the editor and choosing Remove Item from the context menu.

Reusing Query Element Definitions

The screenshot displays the Aras Query Builder interface. On the left, a tree view shows a hierarchy: 'Part' (gear icon) -> '[Part.id = [Part BOM].source_id] Part BOM' (join icon) -> '[[Part BOM].related_id = Part_1.id] Part_1' (gear icon). A context menu is open over the 'Part_1' row, with options: 'Add Related Item', 'Edit Query Element', 'Change Alias', and 'Reuse Query Element Definition'. The 'Reuse Query Element Definition' dialog is open on the right, showing the same hierarchy and an 'Add' button. On the left, a 'Reuse Property Definition' dialog is open, showing a list of properties: 'Part Number', 'Name', 'Cost', 'Length', and 'id'. An arrow points from the 'Length' property to the 'Part_1' row in the tree view. Another arrow points from the 'Extended Property xp-Length' label to the 'Length' property in the list.

Reusing Query Element Definitions


A powerful feature of the Query Builder is the ability to perform recursion over a collection of data that is “nested” in the database. For example, a typical Parts Bill of Materials contains a list of Parts which may in fact be assemblies that represent a further collections of Parts, etc.

The Reuse Query Element Definition feature allows you to automatically perform recursion over a collection of data by reusing the same query definition repeatedly until no more data is found.

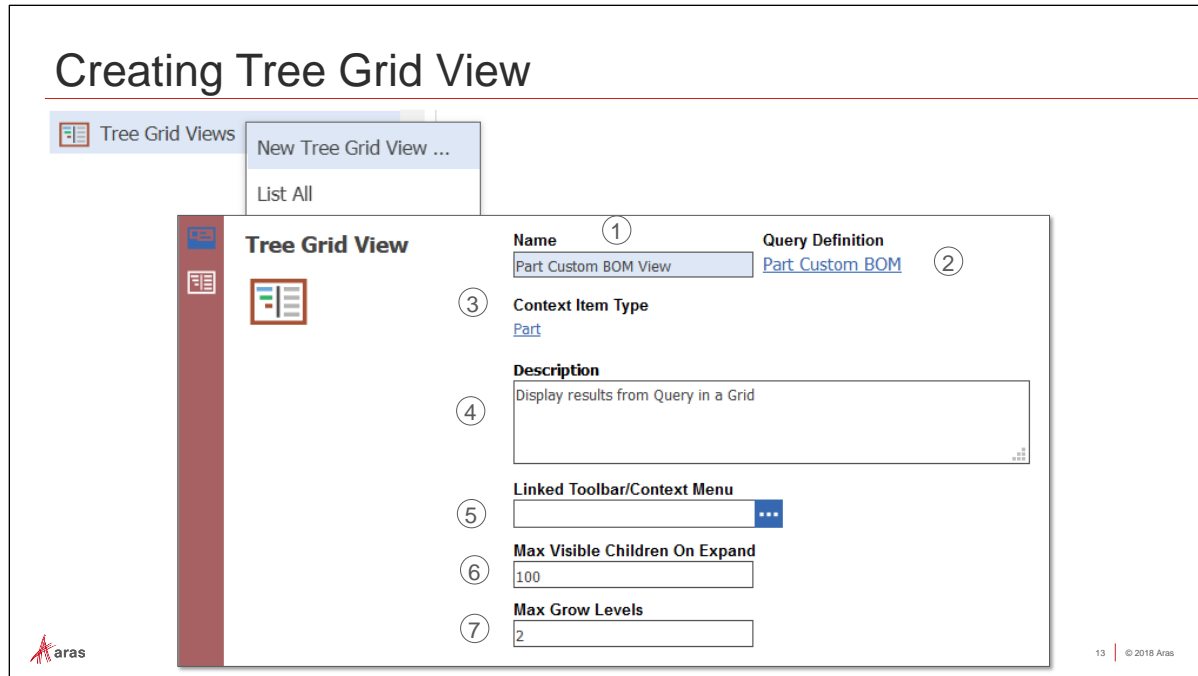


Try it - Reuse a Query Element Definition

1. Select the row representing the related item in the definition editor, in this example select the child Part row.
2. Right-mouse-click and choose Reuse Query Element Definition from the context menu.
3. The Reuse Query Element Definition dialog appears. Click the Part row (first row), then click the Add button to reuse the property selection we have already configured for Parts.

A reuse symbol  will appear to the left of the selected row indicating reuse is enabled.

4. We can also display Extended Properties in a Query result. On the child Part row, add the property Length to the child properties, by clicking the Properties button, adding Length. Click Save.
5. Execute the Query Definition to view the results with the relationship data included. Once again, set the condition [Part Number] = 'MP2963' to retrieve the same part, this time with its related parts and properties.
6. Close the Query Result dialog.
7. Save, Unlock and Close the Part Custom BOM Query Definition.



Creating a Tree View Grid

A Tree Grid View displays data that has been retrieved from a Query Definition. The control can then be associated with a relationship tab or a custom client control (using JavaScript code).

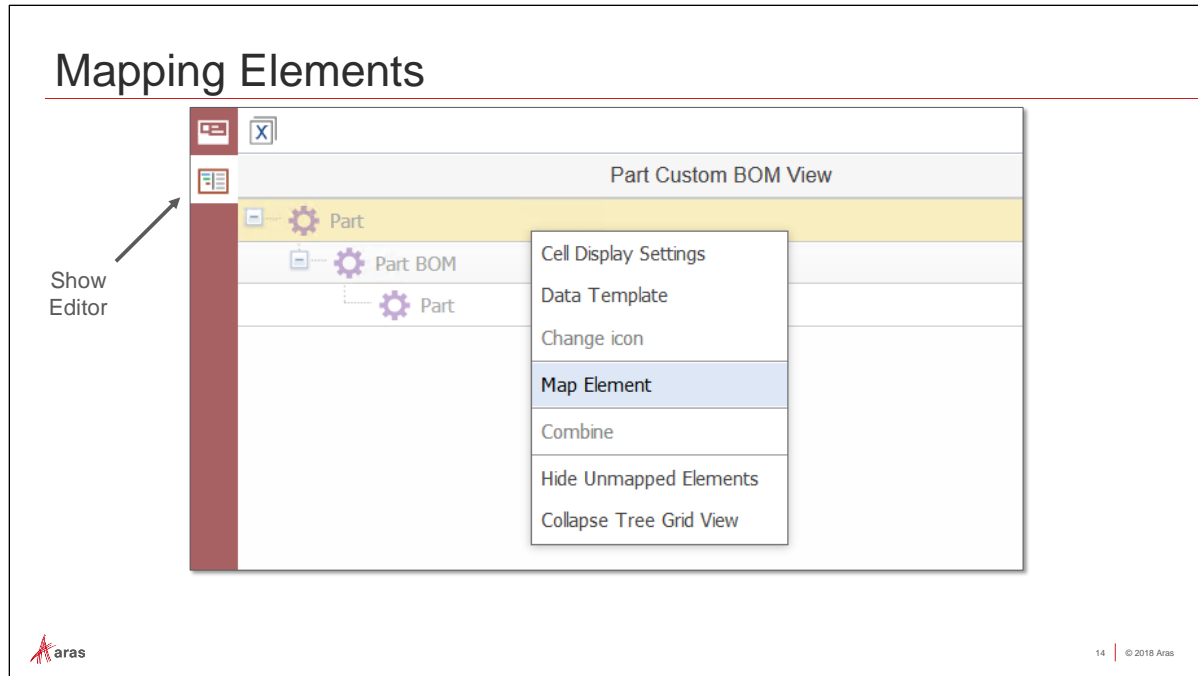
The Tree Grid View form contains the following fields:

- ① **Name** - Name of Tree Grid View.
- ② **Query Definition** - A definition that has been constructed using the Query Builder.
- ③ **Context Item Type** - The item type of the starting context item. This is provided by the selected Query Definition above.
- ④ **Description** – Descriptive Comments about this Tree Grid View.
- ⑤ **Linked Toolbar/Context Menu** - An optional Toolbar menu that has been created using the Configurable User Interface (CUI).
- ⑥ **Max Visible Children On Expand** – Optionally set a limit on the number of child parts displayed
- ⑦ **Max Grow Levels** – Optionally set a limit on the number of levels “deep” to be displayed



Try it - Create a Tree View

1. Navigate to Administration > Configuration > Tree Grid Views from the TOC and choose New Tree Grid View from the context menu.
2. Provide a name and Query Definition. In this example, we will create a Part Custom BOM View using the Query Definition defined in the previous section, which we named Part Custom BOM.
3. Add a Description, and leave the default settings for the remaining properties.
4. Save the Tree Grid View, remaining in edit mode.



Mapping Elements

To expose property values in a Tree Grid View requires mapping an element of the Query Definition to the control. All elements are unmapped by default.

In the example above the Query Definition retrieves Part items that are related to other Part items using the Part BOM relationship.



Try it - Map Elements

1. Click the Show Editor sidebar button to display the Tree Grid View Editor. The editor displays the graphical representation of the Query Definition. Each row in the editor represents a portion of the underlying definition.
2. Select the top row for the Part Item, then choose Map Element from the context menu (right-mouse click).
3. Select the row for Part BOM, then choose Map Element from the context menu (right-mouse click).
4. Select the bottom row for the child Part Item, then choose Map Element from the context menu (right-mouse click).
5. Notice now that each row text is no longer shaded, indicating that each element is mapped.
6. Save the Tree Grid View configuration, remaining in edit mode.

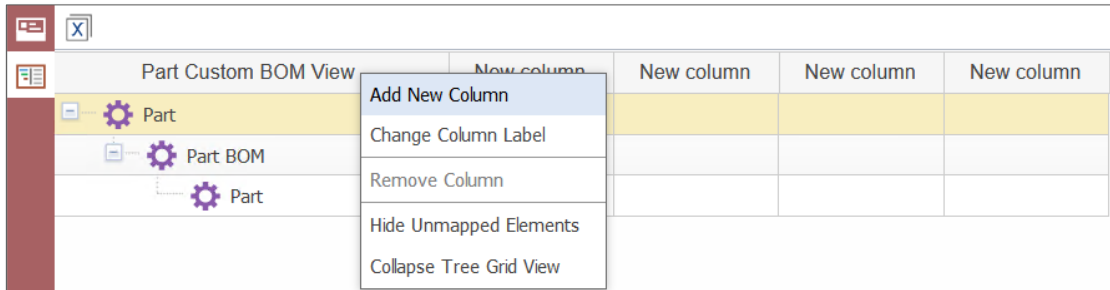
Notes

To remove element mapping select Unmap Element from the context menu.

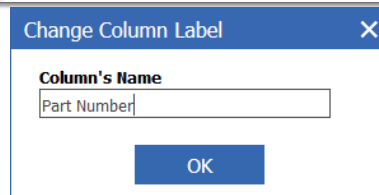
To suppress elements that will not be mapped in the editor choose Hide Unmapped Elements. For larger Query Definitions this can make property mapping easier by eliminating rows that will not be used.

Construct the Grid

- Construct the Grid with a Column for each Property



- Double-Click Column Heading to Rename



Construct the Grid

Build the Grid in the format you wish to display the Query Results. Add a column for each property you wish to display by clicking Add New Column on the context menu (right mouse click). Once the empty grid is created, you can rename the Column Headings and map Property data to each cell.



Try it – Construct the Grid

- In our example, we want to display the Properties: Part Number, Name, Quantity, Cost, and Length. So, we will need a total of five columns in our Grid. Right mouse click and choose Add New Column four times to configure a total of five columns.
- Resize the width of each column using the vertical column separator bar.
- Double-click each column heading to change the Label text, setting each to the property names: Part Number, Name, Quantity, Cost and Length.

Mapping Properties

- Right mouse click – Cell Display Settings
- Double-click cell contents

Mapping Properties

Once a query element (row) has been mapped, and the Grid View created, we will select the properties of that element from the Query Definition to display in the Tree Grid View. The Cell Display Settings editor lets you provide instructions above what displays based on the Cell View Type selected.

The Cell Display Settings editor contains the following fields:

- ① **Cell View Type** - Data type for cell display. Choices include Text, Decimal, Date, Color and Item.
- ② **Text Template** - Text and property data to be displayed in the current cell. Properties are always enclosed in curly brace characters {} and use the syntax *ItemType.propertyname*. You can provide more than one property expression in a cell as well as entering free form text.
- ③ **Helper** - The Helper window shows you the properties that have been configured in the Query Definition for the currently mapped element. Double click on a property to add it to the Text Template window.



Try it - Map a Property

1. The first row of the Tree Grid View displays the column headings. The other rows will display data.
2. Double the cell where the Part Number value will display for Part, displaying the Cell Display Settings.
3. Use the Helper pane to choose the {Part.item_number} property. Double-click for it to show in the Text Template to display in the grid.
4. Since we have set the two Part rows to repeat, you will see the {Part.item_number} placeholder displayed in both rows in the grid.
5. Press the Save button to capture the configuration.

Map Properties into each Grid Cell

The screenshot shows the 'Part Custom BOM View' interface. A context menu is open over the 'Part' node, listing options: 'Add New Column' (1), 'Change Column Label' (2), 'Remove Column', 'Hide Unmapped Elements', and 'Collapse Tree Grid View' (3). Below the menu, a grid view displays the following data:

| Part Number | Name | Quantity | Cost | Length |
|-------------------------|-------------|---------------------|-------------|------------------|
| Part {Part.item_number} | {Part.name} | | {Part.cost} | {Part.xp-length} |
| Part BOM | | {Part BOM.quantity} | | |
| Part {Part.item_number} | {Part.name} | | {Part.cost} | {Part.xp-length} |

Arrows point from the text 'Relationship Property' to the 'Part BOM' row and 'Source / Related Properties' to the 'Part {Part.item_number}' row.

17 | © 2018 Aras

Map Properties into Each Grid Cell

Additional columns show other property values in the Tree Grid View. Similar to a spreadsheet, you decide what mapped properties will be displayed where and then label each column header with appropriate text.

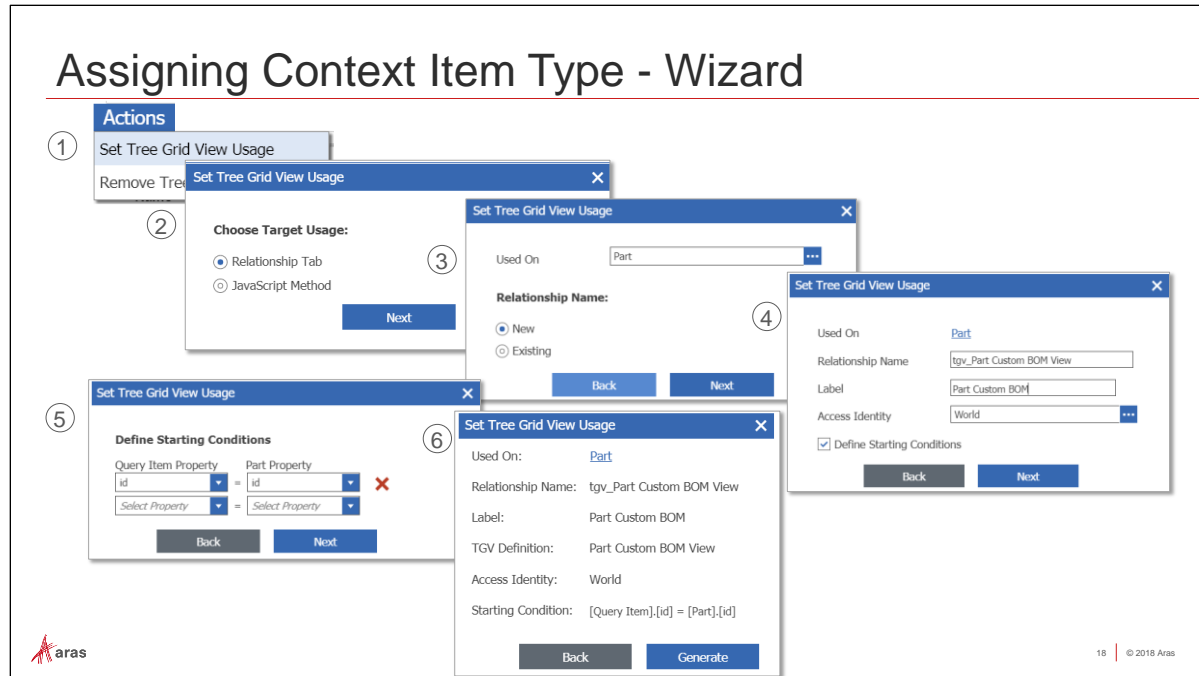
The following options are available from the column context menu.

- ① **Add New Column** - Adds (or inserts) a new column to the right of the currently selected cell.
- ② **Change Column Label** - Allows editing of the column header text.
- ③ **Remove Column** - Removes the currently selected column from the view.



Try it – Map Remaining Properties

1. Map the Part Name property into the Name column (row 2) by double-clicking inside the cell, and double-clicking in the Helper pane, so {Part.name} appears in the Text Template.
2. Map the Part BOM element (row 3) and choose {Part BOM.quantity} to be displayed in the Quantity column. Note that the quantity property is defined on the relationship item.
3. Map the remaining cells for Cost and Length of Parts (row 2).
4. Save changes, leaving the Tree Grid View in edit mode.



Assigning Context Item Type

Once a Tree Grid View has been configured, it is then assigned to an Item Type (or a custom JavaScript Method). The Set Tree Grid View Usage wizard steps you through the process of making the assignment to generate the configuration.



Try it - Assign Context Item Type

1. Select Actions > Set Tree Grid View Usage from the Actions menu of the Tree Grid View.
2. Select the Target Usage. In this example, we will create a new Relationship Tab. Press Next.
3. Supply the targeted Item Type and if this is a new or existing (a replacement). In this example, the Part Item Type is the target and a new relationship will be created. Press Next.
4. A default name for the relationship is provided (and can be modified). Provide a label for the new relationship tab as well who will have access to this view. Type in Part Custom BOM for the Label and use the World identity. Leave Define Starting Conditions checked. Click Next.
5. Define Starting Conditions indicates the starting property values to the underlying Query Definition. In this example, the id of the Part item will be matched to the id of the Query (id = id). Click Next.
6. The summary screen shows the selections you have made. To make changes press the Back button, otherwise press the Generate button to create the new relationship (and tab) on the targeted Item Type.
7. Click Done to confirm that the Configuration Generated Successfully.

Note

To remove a Tree Grid View from the targeted Item Type, use the Remove Tree Grid View Usage action instead. A list of current relationships will be presented that use this view. Select the relationship(s) to be deleted and press the Remove button.

Reviewing Results

The screenshot displays the 'Part Custom BOM' tab in the Aras software. The top section shows part details for MP2963, including revision A, state 'Released', and a cost of 264.6100. Below this, a BOM table lists the components of the 'Stepper Motor Assembly'.

| Part Number | Name | Quantity | Cost | Length |
|-------------|---|----------|----------|--------|
| MP2963 | Stepper Motor Assembly | | 264.6100 | |
| MP1872 | Stepper Motor NEMA17 | 1 | 230.4000 | |
| MP2664 | Spacer Black 22 mm length .14in ID 516in OD | 4 | 0.5000 | |

The interface also includes a sidebar with part metadata (Created By: Mike Miller, Created On: 10/16/2017, etc.) and a top navigation bar with tabs like BOM, BOM Structure, Alternates, AML, Documents, CAD Documents, Goals, Changes, and Part Submission Warrants.

Reviewing Results

To display a Tree Grid View, open an item based on the targeted Item Type and select the relationship tab that corresponds to the Tree Grid View.

We will examine Part Number MP2963 to see the Tree Grid View in a Tab.

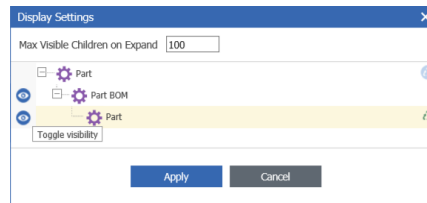


Try It – Review Results in Tab

1. Navigate to Design > Parts in the TOC.
2. Search for Part Number MP2963 Stepper Motor Assembly, and open it in View mode.
3. Locate the new tab labeled Part Custom BOM in the Relationship Tab area.
4. Notice the grid that you designed contains the data as configured.

Tree Grid Toolbar

- Display Settings
- Modify Parameters
- Grow / Grow Depth
- Trim
- Refresh



Part Custom BOM

| Part Number | Name | Quantity | Cost | Length |
|-------------|------------------------|----------|----------|--------|
| MP2963 | Stepper Motor Assembly | | 264.6100 | |
| | | 1 | | |
| | | 4 | | |
| | | 1 | | |
| | | 1 | | |
| | | 4 | | |
| | | 4 | | |



20 | © 2018 Aras

Tree Grid Toolbar

The Tree Grid Toolbar allows the user to change the default view settings, and control the expansion and collapse of the data tree.



Try It – Review Results in Tab

1. Click the Display Setting button on the Tree Grid View Toolbar to toggle visibility of the rows. Click the Apply button to save your settings.
2. Return the Display settings to the default to show all rows.
3. Change the Grow Depth value to be 10, overriding the default value of 2.
4. Click the Grow button to expand all child levels.
5. Change the Grow Depth value to be 1, then click the Grow button to limit the displayed rows.
6. Change the Grow Depth value back to the default value of 2.
7. Click the Grow button to display two levels again.
8. Click on the row containing the Part Number MP2963, and then click the Trim button to collapse all data.
9. Notice how you can rearrange the columns, and resize column width just like other grids in Innovator.
10. Close the Part Number MP2963 Stepper Motor Assembly.

Enhancing Tree Grid View – Combine Rows

The screenshot shows a Tree Grid View with columns: Part Number, Quantity, Name, and Cost. It has three rows: 'Part' (with property {Part.item_number}), 'Part BOM' (with property {Part BOM.quantity}), and another 'Part' (with property {Part.item...}). A context menu is open over the 'Part BOM' row, showing options: Cell Display Settings, Data Template, Change Icon, Unmap Element, and **Combine**. An arrow points from the text 'Combine Selected Rows' to the 'Combine' option.

▪ Must Remap Properties

| Part Number | Name | Quantity | Cost | Length |
|--------------------------------------|-------------|---------------------|-------------|------------------|
| Part {Part.item_number} | {Part.name} | | {Part.cost} | |
| Part BOM --- Part {Part.item_number} | {Part.name} | {Part BOM.quantity} | {Part.cost} | {Part.xp-length} |

▪ Result Combined Rows

| Part Number | Name | Quantity | Cost | Length |
|-------------|---|----------|----------|--------|
| MP2963 | Stepper Motor Assembly | | 264.6100 | |
| MP1872 | Stepper Motor NEMA17 | 1 | 230.4000 | |
| MP2664 | Spacer Black 22 mm length .14in ID 516in OD | 4 | 0.5000 | |
| MP2405 | 83 Tooth GT2 Timing Belt | 1 | 13.7300 | |
| MP1868 | Pulley 5mm shaft | 1 | 17.9600 | |
| MP0370-007 | M3x31 Socket Cap Bolt | 4 | 0.1100 | 31 |

Enhancing Tree Grid View

Tree Grid Views contain configurable features to enhance the display and usability of the view. The first feature allows Query Definition rows to be combined so that property data from more than one item can be displayed on a single row.



Try it - Combine Rows

1. Return to the Tree Grid View we created for Parts by navigating to Administration > Configuration > Tree Grid Views and opening the Part Custom BOM View for edit.
2. In the Editor, select the rows Part BOM element (row 3) and the related Part element (row 4) by using CTRL – click.
3. Right mouse click and select Combine to merge these two rows into a single display row.
4. A warning message appears that all property mapping will be removed. When you accept this message, you must then remap all of the columns using the Cell Display editor.
5. Remap the child Part properties into their appropriate columns. Note that the combined row shows properties for both Part BOM and the related Part in the Cell Display Helper window.
6. Make sure to save your changes to the Tree Grid View.
7. Return to the Part Number MP2963 Stepper Motor Assembly to see your changes reflected in the Part Custom BOM tab.
8. Close the Part Number MP2963 Stepper Motor Assembly.

Enhancing Tree Grid View – Item Links

■ Creating Item Links

Part BOM --- Part {Par...} {Part BOM.quantity} {Part.name} {Part.cost}

Cell View Type: Item
 ItemType Name: Part
 Item Property: {Part.id}

Text Template: {Part.item_number}

Helper:

- {Part.BOM.quantity}
- {Part.cost}
- {Part.id/@keyed_name}
- {Part.id}
- {Part.item_number}
- {Part.name}

| Part Number | Name | Quantity | Cost | Length |
|-------------|---|----------|----------|--------|
| MP2963 | Stepper Motor Assembly | | 264.6100 | |
| MP1872 | Stepper Motor NEMA17 | 1 | 230.4000 | |
| MP2664 | Spacer Black 22 mm length .14in ID 516in OD | 4 | 0.5000 | |
| MP2405 | 83 Tooth GT2 Timing Belt | 1 | 13.7300 | |
| MP1868 | Pulley 5mm shaft | 1 | 17.9600 | |
| MP0370-007 | M3x31 Socket Cap Bolt | 4 | 0.1100 | 31 |
| MP2690 | Washer 0.355 46mm Thk 3.175mm ID 7.95mm OD | 4 | 0.0200 | |

■ Hyperlink

aras 22 © 2018 Aras

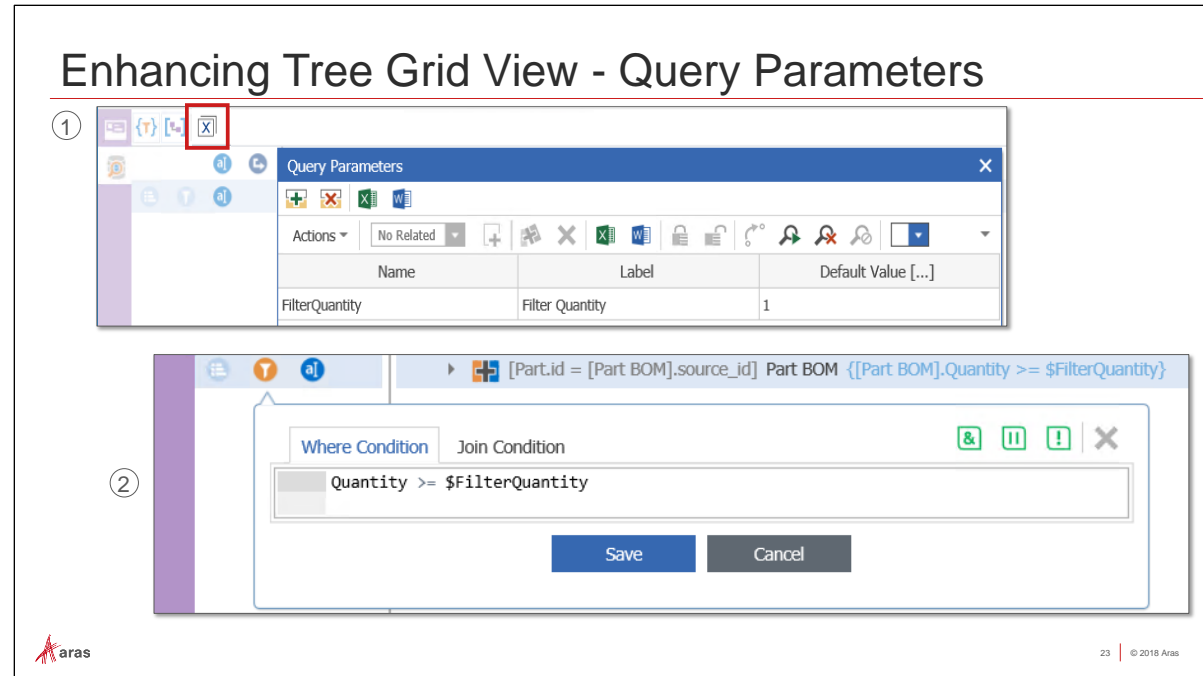
Enhancing Tree Grid View

Another feature to enhance the Tree Grid View changes a property value to a hyperlink, which the user can click to directly open another item window.



Try it - Create Item Links

1. Return to the Part Custom BOM View Tree Grid View configuration, and open the Edit tool again.
2. Select the cell containing the Part Item Number for the child part (3rd row). This will be set as a hyperlink to the child part.
3. Double click to open the Cell Display editor.
4. In the Cell View Type dropdown menu, change to the value to be Item.
5. Set the ItemType Name to be Part.
6. Using the Helper panel, double click on {Part.id} property to set the Item Property to the destination of the hyperlink.
7. Note that the Text Template will still display the value of the Part item_number property {Part.item_number}.
8. Click Save.
9. Save changes to the Tree Grid View.
10. Return to Parts and open the Part Number MP2963 Stepper Motor Assembly to see your changes reflected in the Part Custom BOM tab. You should now be able to click a hyperlink to open the child Part item.
11. Close the Part Number MP2963 Stepper Motor Assembly. Save, Unlock and Close the Part Custom BOM View.



Enhancing Tree Grid View - Query Parameters

Query Definitions can also be parameterized with user-supplied criteria that filters the results of a query at runtime. Parameters will substitute as variables in a Condition statement.



Try it - Define a Query Parameter

1. Navigate to TOC > Configuration > Query Definitions, and open the Query Definition Part Custom BOM for Edit.
2. Click the Show Editor button on the sidebar.
3. Click the Edit Parameters button (X) on the Editor toolbar.
4. In the Query Parameters dialog, click the New (+) button to create a new Parameter with the following values:

| Property | Value |
|---------------|-----------------|
| Name | FilterQuantity |
| Label | Filter Quantity |
| Default Value | 1 |

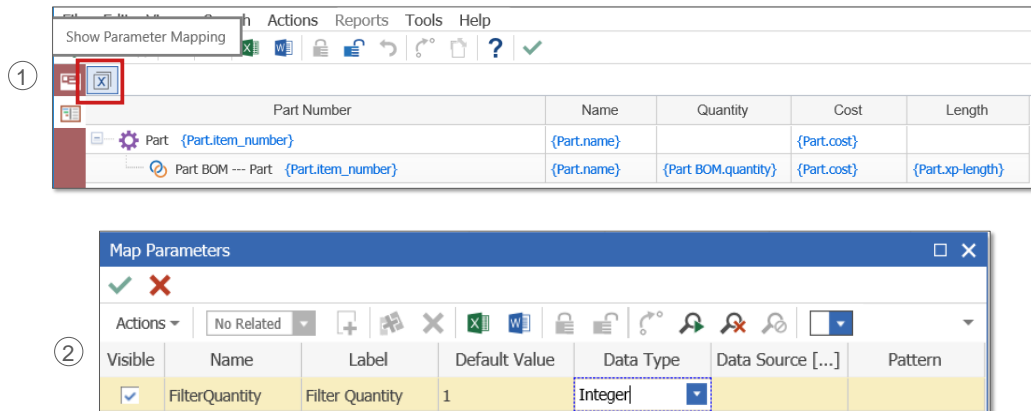
5. Click X to close the Query Parameters dialog.
6. Select the Part BOM Element Query Element Row, and click the Where Condition button (filter).
7. Type in the following condition where the value of the Quantity property on the Part BOM must be greater than or equal to the value of the FilterQuantity parameter:
`Quantity >= $FilterQuantity`
8. Click Save on the Where Condition dialog.
9. Save the Part Custom BOM Query Definition.

Note

To use parameters in a Condition statement, prefix the name with a \$ (dollar sign).

Enhancing Tree Grid View - Enabling Parameters

- Make Parameter Visible on the Tree Grid View



Enhancing Tree Grid View – Enabling Parameters

To allow users to supply parameter values when a Tree Grid View is displayed you must configure the Parameter to be visible.



Try it - Enable Tree Grid View Parameters

- Navigate to Administration > Configuration > Tree Grid Views and open the Part Custom BOM View Tree Grid View for edit.
- Open the Tree Grid View editor and click the Show Parameter Mapping button (X).
- Check the Visible checkbox for the Filter Quantity Parameter and select Integer for the Datatype.
- Click the green checkmark to Save, Unlock and Close the Map Parameters dialog.
- Save, Unlock and Close the Part Custom BOM Tree Grid View.

Enhancing Tree Grid View - Using Parameters

The screenshot illustrates the process of filtering a Bill of Materials (BOM) using parameters. It shows the 'Part' details for MP2963 (Stepper Motor Assembly) and the 'Part Custom BOM' tab. A 'Parameters' dialog box is open, allowing the user to set a 'Filter Quantity' to 4. The resulting BOM table below shows only the components with a quantity of 4.

Parameters Dialog:

| Property | Value |
|-----------------|-------|
| Filter Quantity | 4 |

Filtered BOM Table:

| Part Number | Name | Quantity | Cost | Length |
|-------------|---|----------|----------|--------|
| MP2963 | Stepper Motor Assembly | | 264.6100 | |
| MP2664 | Spacer Black 22 mm length .14in ID 516in OD | 4 | 0.5000 | |
| MP0370-007 | M3x31 Socket Cap Bolt | 4 | 0.1100 | 31 |
| MP2690 | Washer 0.355 46mm Thk 3.175mm ID 7.95mm OD | 4 | 0.0200 | |

Annotations:

- 1. Click the 'Part Custom BOM' button in the toolbar.
- 2. Click the 'Parameters' button (X) in the toolbar.
- 3. Click the 'Grow' button (G) in the toolbar to refresh the filtered data.

Users Can Provide Parameter Value

Grow to See Filtered Results

Enhancing Tree Grid View – Using Parameters

Once a Tree Grid View has been configured with Parameters, a user can click the Modify Parameter button in the relationship grid to supply new filter values. The grid view will update when the user confirms the new value and refreshes or grows the Grid View.



Try it – Use Tree Grid View Parameters

1. Return to Design > Parts open the MP2963 Stepper Motor Assembly part.
2. Click on the Part Custom BOM tab.
3. Click the Modify Parameters button (X) on the Tree Grid View toolbar.
4. Enter a new Integer value for the Filter Quantity, like 4 shown in this example.
5. Save, Unlock and Close the Parameters dialog.
6. Click the Grow button on the toolbar to refresh the filtered data and display the results.



Summary

In this session, you learned how to create a Query Definition that populates a Tree Grid View control.

You should now be able to:

- ✓ Run basic Referencing Tools
- ✓ Describe the Tree View Grid
- ✓ Describe the Query Builder
- ✓ Create a new Query Definition
- ✓ Create a Tree Grid View
- ✓ Assign Tree Grid View Usage
- ✓ Enhance Tree Grid View Display
- ✓ Enable Query Parameters



On Your Own Activity

Goal:

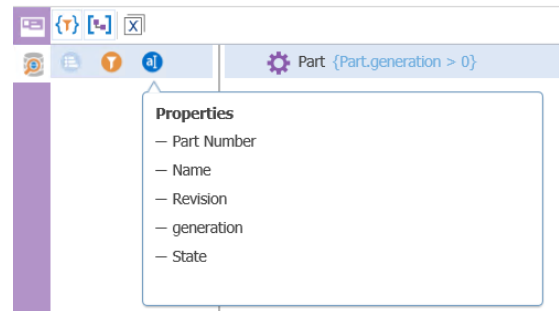
To create new Query Definitions and Tree Grid Views to fulfill User Stories.

User Story:

“I want to view all Revisions of a given item in one tab, so I can retrieve any generation.”

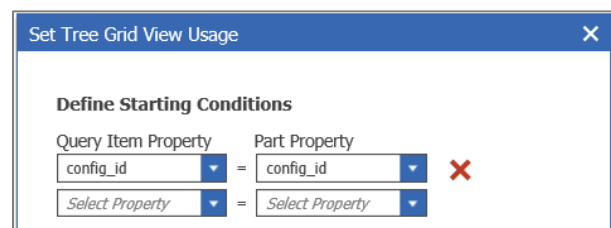
Steps:

1. Create a new Query Definition named QRY All Generations of Part, setting Part for the Context Item Type.
2. Select the Properties: Part Number, Name, Revision, Generation, State and id.
3. Set the Filter condition to: `generation > 0`
4. Set the Sort Order to generation ascending.
5. To test, execute the query filtering on one part that has multiple generations, for example: `[Part Number]='MP0101'`
6. Create a new Tree Grid View named TGV All Generations of Part, using the QRY All Generations of Part as the Query Definition.
7. Create the grid and map properties to display Part Number, Name, Revision, Generation and State.



| Part Number | Name | Revision | Generation | State |
|-------------------------|-------------|------------------|-------------------|--------------|
| Part {Part.item_number} | {Part.name} | {Part.major_rev} | {Part.generation} | {Part.state} |

8. Set the Tree Grid View Usage to display in a Relationship Tab, Used On the Part ItemType, with the label All Generations.
9. The Starting Condition for this Tree Grid View should return all generations. In Innovator, the config_id property will be consistent across generations. So, set the Query Item Property and Part Property conditions to match.
10. To test, open one Part that has multiple generations, for example, MP0101.
11. Optionally, set the Part Number to be a hyperlink for users to open each generation of the Part.



| BOM | BOM Structure | Alternates | AML | Documents | CAD Documents | Goals | Changes | Part Submission Warrants | Part Custom BOM | All Generations |
|-------------|---------------------|------------|------------|-------------|---------------|-------|---------|--------------------------|-----------------|-----------------|
| Part Number | Name | Revision | Generation | State | | | | | | |
| MP0101 | Makerbot Replicator | A | 1 | In Change | | | | | | |
| MP0101 | Makerbot Replicator | B | 2 | Preliminary | | | | | | |
| MP0101 | Makerbot Replicator | B | 3 | Preliminary | | | | | | |